Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

ViserDex: **Vi**sual **S**im-to-**Re**al for Robust **Dex**terous In-hand Reorientation

Arjun Bhardwaj[1], Maximum Wilder-Smith[1], Mayank Mittal[1,2], Vaishakh Patil[1] and Marco Hutter[1]
[1]ETH Zurich, [2]NVIDIA
Email: {abhardwaj, mwilder, mittalma, patilv, mahutter}@ethz.ch

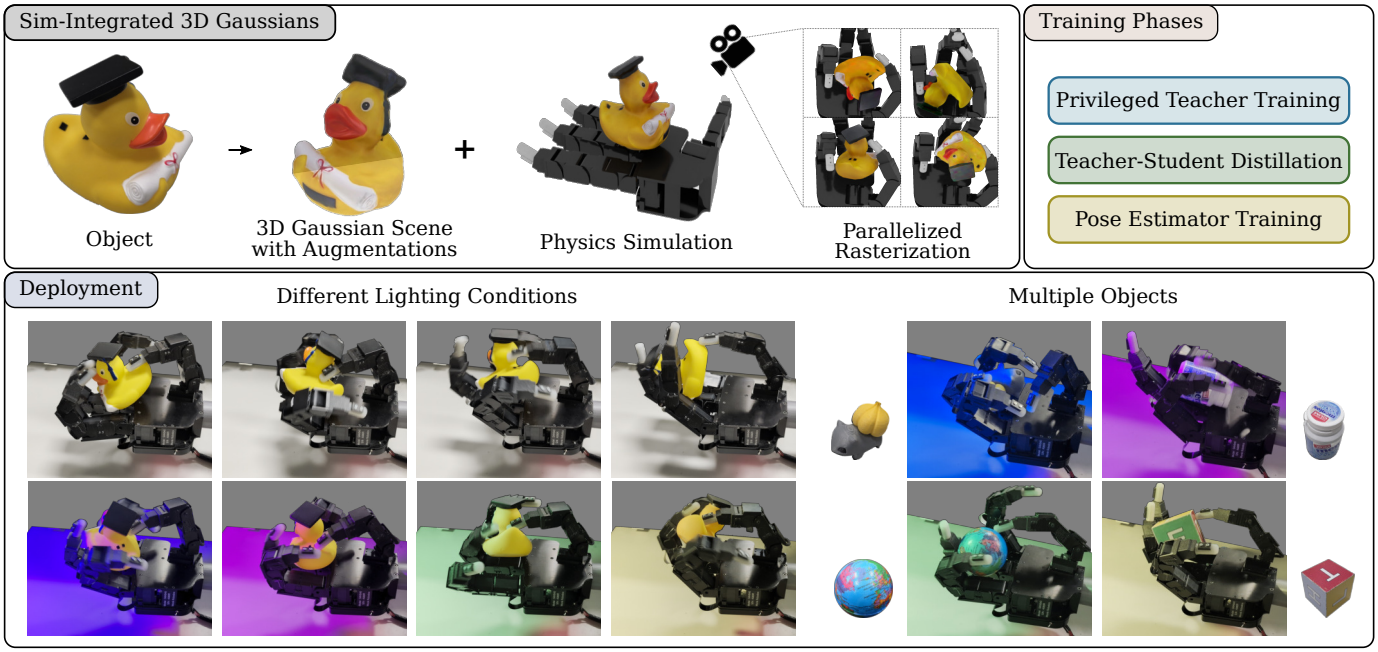

Fig. 1: We introduce a pipeline for training vision-based policies in simulation using 3D Gaussian Splatting. We successfully deploy these policies to the real world using a single monocular RGB camera, demonstrating robust in-hand reorientation of complex object geometries even under adversarial lighting conditions.

***Abstract*—In-hand object reorientation requires precise estimation of the object pose to handle complex task dynamics. While RGB sensing offers rich semantic cues for pose tracking, existing solutions rely on multi-camera setups or costly ray tracing. We present a sim-to-real framework for monocular RGB in-hand reorientation that integrates 3D Gaussian Splatting (3DGS) to bridge the visual sim-to-real gap. Our key insight is performing domain randomization in the Gaussian representation space: by applying physically consistent, pre-rendering augmentations to 3D Gaussians, we generate photorealistic, randomized visual data for object pose estimation. The manipulation policy is trained using curriculum-based reinforcement learning with teacher–student distillation, enabling efficient learning of complex behaviors. Importantly, both perception and control models can be trained independently on consumer-grade hardware, eliminating the need for large compute clusters. Experiments show that the pose estimator trained with 3DGS data outperforms those trained using conventional rendering data in challenging visual environments. We validate the system on a physical multi-fingered hand equipped with an RGB camera, demonstrating robust reorientation of five diverse objects even under challenging lighting conditions. Our results highlight Gaussian splatting as a practical path for RGB-only dexterous manipulation. For videos of the hardware deployments and additional supplementary materials, please refer to the project website: https://rffr.leggedrobotics.com/works/viserdex/.**

I. Introduction

Robotic dexterity requires not only grasping objects but also reorienting them within the hand into precise, functional poses. Deep Reinforcement Learning (DRL) has shown promise in acquiring such skills [2, 6]. However, existing methods often succeed only with visually simple objects, such as colored cubes, and struggle with realistic textures, complex shapes, and varied appearances. A major challenge is the perception–control gap: rapid in-hand motions create severe self-occlusions, making accurate object pose estimation from real sensors extremely difficult. Alternative sensing modalities, such as tactile arrays [17, 33], depth cameras [4], or multi-view rigs [6], offer partial solutions but introduce instrumentation overhead, calibration complexity, or limited scalability. Consequently, sim-to-real dexterous control has often avoided relying primarily on monocular RGB camera observations, resulting in a perception–robustness bottleneck that constrains current approaches.

A fundamental challenge in RGB-based manipulation lies in the simulation pipeline. Achieving the photorealism required for robust sim-to-real transfer via standard mesh-based ren-

dering is computationally intractable for high-throughput RL training. Existing methods [25, 26] attempt to address this gap using high-fidelity visual simulation; however, generating sufficient visual diversity over days of training demands massive compute clusters, even for simple objects [6]. Explicit scene representations, particularly 3D Gaussian Splatting (3DGS) [9], enable real-time, photorealistic rendering that outperforms traditional mesh rasterization. Its compact and flexible scene representation allows efficient manipulation of scenes, making it ideal for RL tasks that require large-scale visual diversity. Despite these advantages, standard 3DGS is limited to static scenes and entangles illumination with geometry [9]. This prevents independent manipulation of lighting and material properties, which is a prerequisite for Domain Randomization (DR) that facilitates robust sim-to-real transfer. In this work, we investigate strategies to overcome these limitations to generate diverse, dynamic visual data for robust object pose estimation during dexterous manipulation.

This paper proposes a monocular RGB-based training and deployment pipeline for robust in-hand reorientation of complex objects. The system is decomposed into two components: object pose estimation from RGB images as geometric keypoints, and an RL control policy that reorients the object to a desired goal pose. We integrate 3DGS directly into the simulation loop, overcoming the limitations of standard mesh-based rendering for high-throughput simulation-based training. Our key contribution is a suite of *pre-rasterization augmentations* for Gaussian scenes. These augmentations generate consistent, diverse visual data and relax the static scene assumption of vanilla 3DGS. Furthermore, we simplify the RL process by replacing the extensive DR schemes used in previous works [1, 6] with a performance-based curriculum and a student-teacher distillation framework, substantially improving training efficiency. Notably, our pipeline enables learning complex manipulation behaviors using only a single consumer-grade GPU. We validate this approach through zero-shot sim-to-real transfer on a 16-DoF Allegro Hand with a monocular RGB camera. Our approach demonstrates robust performance across five objects under both nominal and adversarial lighting conditions (shown in Fig. 1), achieving over 25 consecutive successful reorientations on average.

II. Related Works

A. Sim-to-Real RL for In-Hand Manipulation

In-hand manipulation tasks in sim-to-real RL can be categorized into two primitives: continuous in-hand rotation and goal-conditioned reorientation. Continuous rotation, where the objective is to spin an object around a canonical axis, has been demonstrated using proprioceptive and tactile feedback [19, 32, 33], as well as through repetitive open-loop finger gaits [3].

Goal-conditioned reorientation, in contrast, requires precise object state tracking and geometric reasoning to repose the object to the target configuration. Pitz et al. [17] propose a tactile-based object state estimator. While tactile sensing captures local geometry, it lacks a global reference frame, making their method susceptible to drift over long horizons and unable to resolve fine-grained features. Depth-based approaches [4] provide geometric structure but miss the semantic texture information needed to disambiguate the orientation of symmetric or visually complex objects.

RGB vision provides dense semantic feedback useful for robust in-hand reorientation. Prior works [2, 6] have applied this modality to simple objects, such as colored cubes, but typically rely on multi-camera setups to handle occlusions. These methods also use computationally expensive Automatic Domain Randomization (ADR) [1] to bridge the sim-to-real gap, requiring large-scale compute clusters. While recent efforts improve training efficiency through student-teacher distillation [25], generating photorealistic visual data remains a bottleneck. Developing a monocular RGB-based framework that is both computationally efficient and generalizes to complex, real-world objects remains an open challenge.

B. 3D Gaussian Splatting for Robotic Manipulation

3D Gaussian Splatting (3DGS) [9] was developed for fast novel-view synthesis, generating photorealistic images from unseen viewpoints. Its rapid, high-fidelity reconstruction has been adopted in robotics for SLAM [12, 8], teleoperation [30, 10], and planning [5, 13]. The ability to easily capture and reconstruct arbitrary objects makes 3DGS a promising tool for reducing sim-to-real gaps in deploying visual policies.

Methods such as SplatSim [20] leverage 3DGS to produce higher-fidelity observations than mesh-based renders, improving realism and reducing sim-to-real discrepancies in manipulation tasks. However, traditional 3DGS scenes are optimized for single objects or static scenes, which limits domain randomization on them. GSRL [28] addresses this by training a network to accelerate Gaussian generation across multiple scenes, and RL-GSBridge [31] and RoboGSim [11] combine physics-compatible meshes with high-quality Gaussian rendering for visual RL. RoboGSim further expands the training domain with a Scene Composer that randomizes objects, backgrounds, and viewpoints. Our work extends 3DGS by integrating it directly into a high-throughput simulation loop, combined with pre-rasterization augmentations. This approach addresses both visual realism and domain diversity challenges, paving the way for efficient sim-to-real transfer of dexterous manipulation policies.

III. Method

We consider the problem of continuous, goal-conditioned in-hand object reorientation using a multi-fingered robotic hand observed only through a monocular RGB camera. From a control perspective, this task constitutes a Partially Observable Markov Decision Process (POMDP) with severe perceptual challenges. During manipulation, the object undergoes rapid motion, frequent self-occlusions by the fingers, and significant motion blur, making direct state estimation from RGB images highly unreliable. At the same time, learning the underlying dexterous manipulation skills requires accurate reasoning about object geometry and contact dynamics, which is difficult to acquire from raw visual input alone.

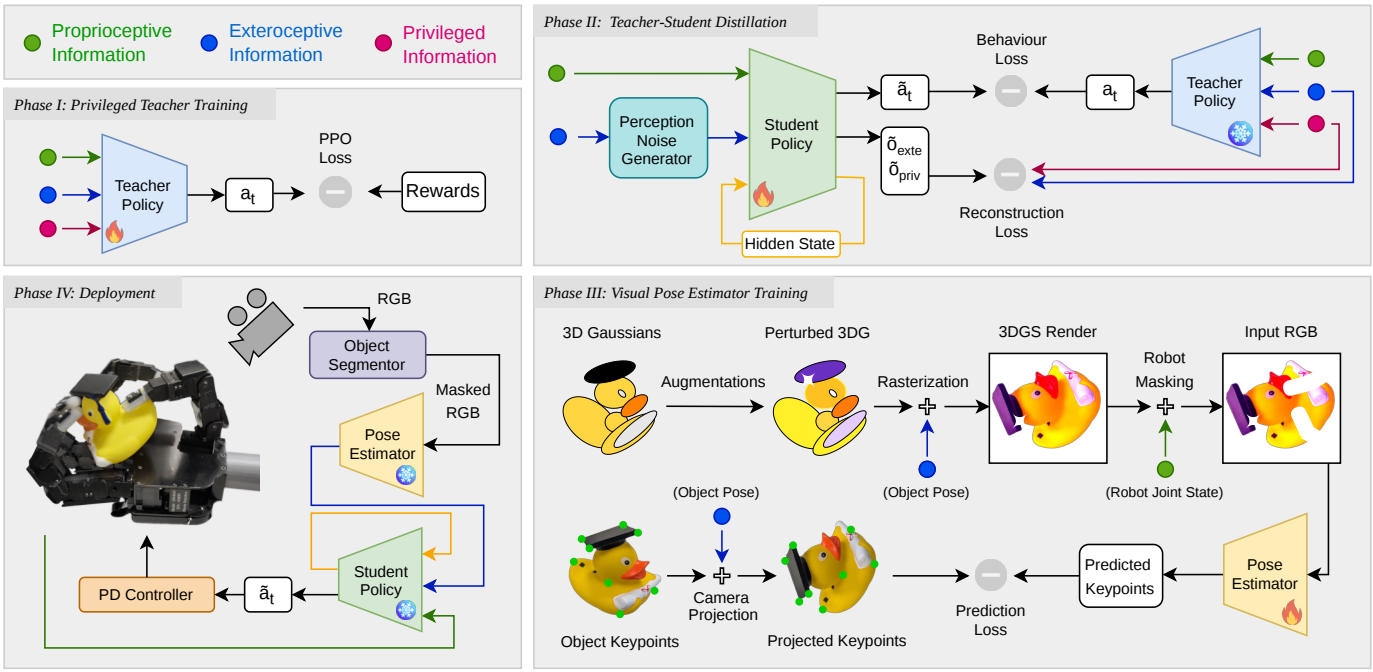

Fig. 2: Overview of our sim-to-real in-hand reorientation pipeline. We first train a teacher policy in simulation with full state access, then distill it into a recurrent student policy that operates from noisy observations. A monocular RGB pose estimator trained with 3D Gaussian Splatting data provides object poses to the student policy, enabling goal-conditioned dexterous manipulation on a real multi-fingered hand.

A naive end-to-end approach that learns control directly from images must simultaneously solve three difficult problems: learn dexterous motor skills, infer hidden object state from partial observations, and overcome the visual sim-to-real gap. Jointly optimizing these objectives with reinforcement learning is challenging and sample inefficient. Instead, we follow a principled decomposition that addresses each challenge in a separate phase:

1) **Teacher Training using RL:** We first train a teacher policy in simulation with full state access using reinforcement learning. This stage focuses solely on acquiring the geometric and contact-rich manipulation skills required for reorientation.
2) **Student Distillation:** We distill the teacher into a recurrent student policy that learns to infer the underlying system state from noisy, real-world observations.
3) **Visual Pose Estimator Training:** We train a monocular RGB pose estimator using synthetic images rendered from a 3D Gaussian Splatting representation of the object. By performing domain randomization directly in the Gaussian space prior to rendering, we generate photorealistic and diverse training data that significantly reduces the visual sim-to-real gap.

At deployment, we predict object pose estimates from RGB images, which are fed to the recurrent student policy to produce goal-conditioned actions on the real robot. This modular design enables efficient training on consumer-grade hardware. An overview of the pipeline is shown in Fig. 2.

A. Teacher Training using Reinforcement Learning

1) MDP Formulation: We formulate the reorientation task as a goal-conditioned MDP, and train a policy $\pi_\theta(a_t|o_t, g_t)$ to rotate the object to a target orientation $g_t \in \mathrm{SO}(3)$ using PPO [24]. The policy commands the robot's joint position targets $a_t \in \mathbb{R}^{16}$. The agent receives a dense reward for aligning the object with the goal, a sparse success bonus, and penalties that encourage smooth actions. Upon reaching a goal, a new target is sampled. Episodes terminate if the object is dropped or if no goals are achieved within a specified time window. Additional details are provided in the Appendix.

2) Observations: We divide the observation space $\mathcal{O}$ into three groups: *proprioceptive*, *exteroceptive*, and *privileged*. Proprioceptive observations $\mathcal{O}_{\text{prop}}$ consist of robot's joint positions, action history from the last four steps, the current object goal, and the remaining episode time. Exteroceptive-derived observations $\mathcal{O}_{\text{exte}}$ provide the object's current pose in the hand frame and its orientation relative to the goal. Privileged observations $\mathcal{O}_{\text{priv}}$ is available only to the teacher policy and contain ground-truth state information, including the object's velocity, robot's fingertip contact forces, and randomized physical properties (*e.g.*, object mass and scale).

3) Performance-based Curriculum: Prior work on in-hand reorientation [1, 6] uses ADR over many environment parameters, which is computationally expensive. Instead, we propose a lightweight, performance-driven curriculum [15] that increases the task complexity according to the agent's average consecutive success count. The curriculum has three complementary components. First, we gradually increase the regularization penalties, allowing the policy to prioritize task completion before refining smoothness and efficiency. Second, we incrementally increase the random action latency to prepare the agent for asynchronous delays on real hardware. Finally, we also progressively narrow the allowed time window between consecutive successes, encouraging more efficient

object reorientation. Each curriculum component scales with the moving average of consecutive successes over all the environments. Together, these mechanisms improve sample efficiency and stabilize training, shown later in Section IV-C.

B. Student Training using Distillation

The RL policy from the previous phase has access to privileged signals, which are unavailable during deployment. We therefore distill it into a student policy which receives noisy proprioception $o_{\text{prop}}^{\text{noisy}}$ and noisy exteroceptive $o_{\text{exte}}^{\text{noisy}}$ observations. To handle partial observability, we parameterize the student as a recurrent network with a belief encoder [14], allowing it to implicitly infer the system state.

1) Perception Noise Generator: Inspired by [6], we corrupt simulated object pose observations with four perturbations. We apply temporal downsampling to simulate low frame rates, stochastic jitter to model variable latency, systematic bias for calibration errors, and inject random poses for occasional tracking failures. This realistic noise improves student policy robustness and facilitates effective sim-to-real transfer.

2) Student Policy Architecture: The student policy uses a *belief encoder-decoder* network architecture, previously applied in perceptive locomotion [14]. At each timestep, the recurrent encoder updates a latent belief state $z = f_\phi(o_{\text{prop}}^{\text{noisy}}, o_{\text{exte}}^{\text{noisy}})$. During training, the belief decoder network reconstructs the teacher's observations $(\tilde{o}_{\text{exte}}, \tilde{o}_{\text{priv}}) = h_\psi(z, o_{\text{exte}}^{\text{noisy}})$. The encoder-decoder is trained with a reconstruction loss $\mathcal{L}_{\text{recon}}(\phi, \psi)$, which penalizes errors in reconstructing the privileged and exteroceptive information. This encourages the latent z to capture the structure necessary to combine noisy inputs and compensate for partial observability.

The control head outputs the actions $\tilde{a} = g_\rho(z, o_{\text{prop}}^{\text{noisy}}, o_{\text{exte}}^{\text{noisy}})$, supervised with a behavior cloning loss $\mathcal{L}_{\text{BC}}(\phi, \rho)$, that minimizes the L2 distance between the student and teacher actions. We train the student policy end-to-end using a composite loss function $\mathcal{L} = \mathcal{L}_{\text{BC}} + \lambda \mathcal{L}_{\text{recon}}$ and employ an online variant of DAgger [22] detailed in the Appendix.

C. Visual Object Representation and Augmentations

The student policy's exteroceptive input requires object pose estimates from RGB images. A key challenge in training a robust perception model for this task is generating diverse visual data that handles lighting variations and heavy occlusions. To avoid the computational overhead of ray-tracing in simulators, we integrate Gaussian Splatting rasterization into the simulation loop.

1) 3D Gaussian Object Representation: We represent object geometry and appearance using 3DGS [9]. The object is represented by a set of 3D Gaussians, each characterized by a position, covariance, opacity, and spherical harmonic (SH) coefficients [23]. To capture view-dependent effects, the color $c(\mathbf{d})$ along the viewing direction $\mathbf{d}$ is computed by spherical harmonics up to degree $L = 3$:

$$c(\mathbf{d}) = \text{Sigmoid}\left(\sum_{\ell=0}^{L}\sum_{m=-\ell}^{\ell} k_\ell^m Y_\ell^m(\mathbf{d})\right) \tag{1}$$

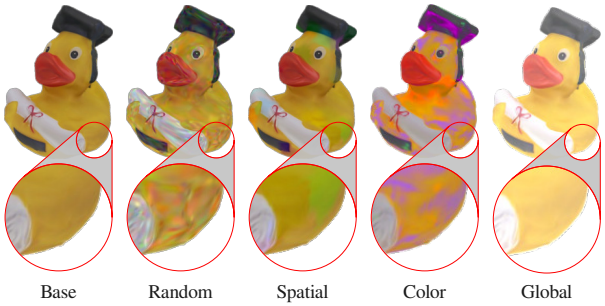

Fig. 3: Pre-rasterization augmentation examples. Visualizations of the proposed SH-based perturbations applied to clustered Gaussians, producing structured variations in color, reflectance, and spatial appearance without ray-tracing.

where k_ℓ^m are the learned coefficients. The 0^{th}-order coefficients (SH0) capture view-independent Lambertian base colors, while the higher-order coefficients (SHN) encode high-frequency specular effects.

2) Simulation-Integrated GS Rendering: During in-hand manipulation, the camera is fixed while the object moves due to robot interactions. To render the moving object with 3DGS, we apply the inverse of the object's transform to the camera and produce RGB and depth D_{splat}. This transformation keeps the scene as static, satisfying the assumptions of vanilla 3DGS, while producing images that reflect changing object poses. However, the object-centric rendering ignores occlusions from the robot's fingers. To restore physical consistency, we generate a depth map of the hand D_{phys} from the same viewpoint using a low-fidelity depth raycaster within the physics simulator. We mask out pixels in the RGB image where the hand is in the front ($D_{\text{phys}} < D_{\text{splat}}$). This aligns visual observations with the simulation state without full-scene ray-tracing.

3) Pre-Rasterization Augmentations: Strategies for generating diverse visual data typically fall into two categories. First, *post-process image augmentations* (*e.g.*, color jitter, brightness), which are computationally cheap but apply global 2D transformations that disregard 3D geometry. Second, *scene parameter randomizations* (domain randomization) [26] alter lighting and materials during rendering but require computationally expensive ray-tracing.

We propose a hybrid approach termed *pre-rasterization augmentation*, which leverages the explicit nature of the 3D Gaussian representation. We operate directly on the Gaussian attributes, specifically the Spherical Harmonic (SH) coefficients, before rasterization. This provides fine-grained control over scene appearance without the cost of full scene ray-tracing. Naïvely randomizing individual Gaussians, however, breaks photometric consistency, producing high-frequency noise rather than realistic lighting variations. To generate diverse yet plausible data, we exploit the fact that lighting and material changes are inherently structured, typically affecting spatially proximal regions or specific materials uniformly. We therefore cluster Gaussians based on geometric or photometric correlations and perturb each cluster as a group. These perturbations include additive and scaling noise on the base color (SH0) and specular components in higher-order SH

TABLE I: Pre-Rasterization Augmentation Parameters

Augmentation	Targets	Probability	Fraction	Range
RANDOM NOISE				
Additive	SH0, SHN	0.2	1.0	$[-0.1, 0.1]$
Scaling	SH0, SHN	0.2	1.0	$[0.8, 1.2]$
SPATIAL CLUSTER				
Additive	SH0, SHN	0.8	0.10	$[-0.1, 0.1]$
Scaling	SH0, SHN	0.8	0.20	$[0.9, 1.1]$
COLOR CLUSTER				
Additive	SH0	0.8	0.10	$[-0.2, 0.2]$
Additive	SHN	0.8	0.10	$[-0.1, 0.1]$
Scaling	SH0, SHN	0.8	0.10	$[0.6, 1.4]$
GLOBAL SHIFT				
Additive	SHN	0.2	1.0	$[-0.1, 0.1]$
Scaling	SH0, SHN	0.2	1.0	$[0.6, 1.4]$
Uniform Additive	SH0, SHN	0.8	1.0	$[-0.2, 0.2]$
Uniform Scaling	SH0	0.8	1.0	$[0.9, 1.4]$

coefficients (SHN). They are summarized as follows:

- **Random Noise Group:** We apply random perturbations to each Gaussian independently. This unstructured randomization effectively simulates high-frequency sensor noise, pixel-level artifacts, and minor mesh imperfections.
- **Spatial Cluster Group:** Real-world variations such as shadows, damage, and marks are often localized in a small region on the object. To mimic this effect, we cluster Gaussians by spatial location into 64 clusters using k-means. Perturbing these spatially contiguous clusters simulates local inconsistencies and patch-level noise.
- **Color Cluster Group:** Objects are often composed of distinct materials that interact differently with light. Assuming that these material properties correlate with diffuse color, we cluster Gaussians based on their 0^{th} order spherical harmonic (SH0) into 32 clusters. Perturbing these clusters simulates material-specific shifts, such as albedo modifications or reflectance changes, across photometrically similar regions.
- **Global Shift Group:** To simulate macro-level environmental changes, we treat the entire scene of Gaussians as a single cluster. We apply global shifts and scaling to the color attributes, where noise is sampled either independently per dimension or as a single value (Uniform) for the entire vector. These perturbations effectively replicate environmental variations such as ambient brightness, color temperature, camera exposure, and saturation.

Each Gaussian scene is preprocessed to identify the cluster indices for these strategies. During visual data generation, we sequentially apply stochastic variations to the SH coefficients, allowing different visual perturbations to compound. This produces a diverse set of scene appearances from a single static representation, as shown in Fig. 3. Table I summarizes the augmentation types and parameters. Each augmentation is applied with a specified probability, and only a fraction of the clusters are perturbed at a time. Despite their distinct semantic targets (*e.g.*, spatial vs. color), the procedural logic for applying any augmentation layer is consistent.

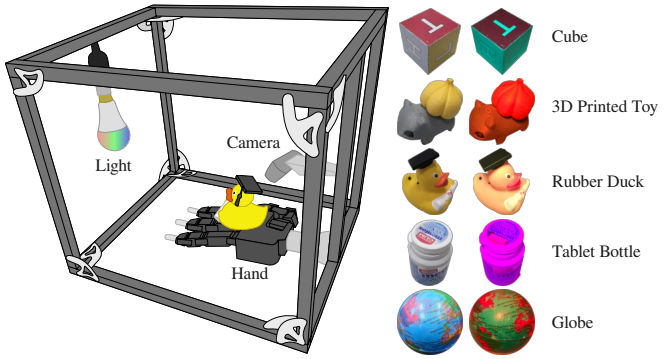

Fig. 4: Left: The experimental setup with an RGB camera, an Allegro Hand, and a multi-colored light source for adversarial lighting. Right: The object set displayed under normal lighting (first column) and adversarial lighting (second column).

D. Visual Object Pose Estimator Training

We train a keypoint-based pose estimator to recover the object pose from RGB images, thereby providing the exteroceptive input required by the student policy during real-world deployment. The training dataset is generated by rolling out the expert teacher policy within the simulation and rendering the RGB images through our 3DGS pipeline with pre-rasterization augmentation strategies. This process results in a large-scale, annotated dataset with ground-truth object poses. To improve robustness to real-world sensor imperfections, we apply random ISO noise and motion blur to the images during training.

The pose estimator uses a ResNet-34 [7] backbone, initialized with ImageNet-pretrained weights. The network is trained to regress a set of nine keypoints, corresponding to the eight object-specific points plus the geometric centroid. For each keypoint, the network predicts the normalized 2.5D coordinates (u, v, d), where (u, v) represent the normalized pixel coordinates in the image plane and d represents the metric depth. The predicted 2.5D keypoints can be resolved to a 6D object pose via the Rigid Procrustes algorithm [27].

E. Experimental Setup

We consider a 16-DOF Allegro Hand with a wrist-mounted Intel RealSense D435i camera for visual feedback, as shown in Fig. 4. We deploy all models on a single workstation with an Intel Core i9 CPU and NVIDIA RTX 6000 Ada GPU. The policy infers at 30 Hz and outputs the joint position targets that are tracked by a low-level joint PD controller at 300 Hz. Since the learned pose estimator relies on segmented object inputs, we fine-tune SAM2 [21] per object to generate prompt-free segmentation masks in real-time. From the masked RGB observations, the pose network predicts 2.5D keypoints, which are subsequently resolved into a 6D object pose via the Rigid Procrustes algorithm [27]. This estimated pose then serves as the real-time state feedback for the control policy. The policy operates asynchronously with respect to the estimator, querying the most recent pose via a zero-order hold.

We evaluate our system on five objects (see Fig. 4) exhibiting diverse geometric and physical properties, spanning primitives shapes (*Cube*, *Globe*) to complex, non-convex items (*Tablet Bottle*, *3D Printed Toy*, *Rubber Duck*). For these

objects, high-fidelity meshes are obtained using Polycam [18]. These are then rendered in NVIDIA Isaac Lab [16] to generate pose-annotated images for Gaussian Splatting. For each object, we train a pose estimator and control policy in NVIDIA Isaac Lab. Additional training details are provided in the Appendix.

We consider two lighting regimes: nominal (white distant light) and adversarial (low-illumination point sources with dynamic hue shifts). As shown in Fig. 4, adversarial conditions present significant visual challenges, including low contrast, specular highlights, and strong color casts, providing a rigorous test of system robustness.

IV. Results

A. Pose Estimation using Different Rendering Pipelines

To evaluate our pose estimation pipeline, we create a real-world test set for each object using FoundationPose [29] as a ground-truth pose labeler. To ensure high-quality labels, we provide FoundationPose with privileged inputs, including object masks, CAD meshes, and high-resolution RGB-D images, and perform multiple refinement iterations. We discard frames where the rendered pose does not visually align with the input image. Additional details on test dataset is in the Appendix.

We evaluate our proposed pose estimator training pipeline against three baselines, keeping network architecture, training hyperparameters, and dataset sizes constant. The baselines differ solely in their image generation approach:

1) *Standard Tiled Rendering:* Isaac Lab's default RTX renderer combined with standard post-process image augmentations.
2) *Domain Randomized (DR) Tiled Rendering*: Extends the above with randomized scene attributes (background HDRI lighting, material properties such as albedo tint, roughness, and metallic). We use the randomization parameters for material and background from [25].
3) *Naïve Gaussian Splatting:* Our GS-based pipeline without pre-rasterization augmentations, and using only standard post-process image augmentations.

All methods use the same source meshes for geometry-based rendering and Gaussian Splat scene optimization to ensure a fair comparison. We report the Average Distance of Model Points (ADD) and a prediction accuracy metric (error < 10mm and $< 10^\circ$), averaged over five training seeds.

a) Nominal Conditions: Table II details the pose estimation results under nominal lighting. Our method achieves the highest overall performance with a mean accuracy of 65.4% and a mean ADD of 10.2 mm, surpassing standard tiled rendering (53.3%) and the randomized tiled baseline (55.6%). While geometrically simple objects (*e.g.*, *Cube*) show similar performance across methods, our approach yields the largest gains on geometrically and texturally complex objects, such as the *3D Printed Toy* ($+24\%$ improvement over randomized tiled rendering) and the *Rubber Duck*.

b) Adversarial Conditions: Under adversarial lighting, the performance differences between methods become even more pronounced (see Table II). Our method demonstrates superior robustness, with a mean accuracy of $\mathbf{56.3}\%$, significantly outperforming the strongest baseline, Randomized Tiled Rendering (47.2%). A critical observation is that the weak performance of the Naïve GS baseline achieves only 36.5% accuracy. Since this baseline uses the same underlying rendering engine but lacks our specific randomization strategy, this failure highlights that high-fidelity rendering alone is insufficient for generalizing to out-of-distribution visual domains. In contrast, our approach leverages explicit control over scene attributes to generate diverse, challenging training samples. By tailoring these pre-rasterization augmentations to simulate physical lighting variations, which standard 2D image augmentations fail to capture, we maintain performance even in difficult visual scenarios.

c) Computational Efficiency: Beyond data quality, our integrated 3DGS pipeline provides significant computational advantages over standard rendering solutions. Compared to Isaac Lab's tiled renderer, it achieves a $1.6\times$ faster rendering throughput on an RTX 6000 Ada. Furthermore, it is substantially more memory-efficient: rendering a batch of 1,024 environments consumes only 12 GB of VRAM, compared to the prohibitive 34 GB required by the tiled renderer. Critically, our proposed pre-rasterization augmentations introduce negligible overhead, executing in less than 2 ms per batch, and representing only $\approx 4\%$ of the total frame rendering time.

B. Effect of Pre-Rasterization Augmentations

To isolate the contributions of the pre-rasterization augmentations introduced in Section III-C3, we perform a "leave-one-out" ablation study. For each experiment, a pose estimator is trained with data generated with one augmentation group (defined in Table I) removed and evaluated against the full pipeline. Results on the nominal and adversarial test set are reported in Table III.

a) Nominal Conditions: We observe that the complete augmentation pipeline yields the highest overall robustness. The removal of localized augmentations, specifically *Random Noise* and *Structured Clustering* (Means/SH0), results in moderate performance dips (the accuracy dropping to 57-59%). This suggests that while fine-grained perturbations refine the decision boundary, they are not the sole drivers of performance under nominal conditions, where lighting is relatively standard.

However, a critical insight emerges from the exclusion of *Global Shift* augmentations. Removing these environmental perturbations causes a significant performance collapse, with mean accuracy dropping to 51.2% and ADD error nearly doubling to 16.9 mm. This degradation is particularly acute for the *Tablet Bottle*, which has reflective surfaces. This indicates that even under "nominal" real-world conditions, simulating global variations in exposure, color temperature, and ambient intensity is essential for bridging the sim-to-real gap.

b) Adversarial Conditions: First, removing *Random Noise* results in a comparatively moderate performance drop (Mean Accuracy: 48.6%), suggesting that while unstructured noise aids general robustness, it is not the primary driver of feature learning. In contrast, omitting the structured aug-

TABLE II: Evaluation of learned pose estimator on real-world data under nominal and adversarial lighting. Our method is compared against three baselines: *Standard Tiled*, *Randomized Tiled*, and *Naive GS* rendering. We report Average Distance of Model Points (ADD) in mm and a strict accuracy metric (< 10mm and $< 10^\circ$), averaged over 5 random seeds.

Objects	**Cube**		**3D Printed Toy**		**Rubber Duck**		**Tablet Bottle**		**Globe**		**Mean**	
Method	ADD	Accuracy	ADD	Accuracy	ADD	Accuracy	ADD	Accuracy	ADD	Accuracy	ADD	Accuracy
NOMINAL CONDITIONS												
Standard Tiled	$11.9_{\pm 0.86}$	$57.2_{\pm 8.13}$	$14.1_{\pm 0.92}$	$37.7_{\pm 2.84}$	$9.5_{\pm 0.23}$	$67.0_{\pm 1.47}$	$10.0_{\pm 0.65}$	$45.7_{\pm 4.78}$	$15.0_{\pm 2.41}$	$59.1_{\pm 3.68}$	$12.1_{\pm 1.01}$	$53.3_{\pm 4.18}$
DR Tiled	$10.5_{\pm 0.75}$	$69.7_{\pm 9.05}$	$15.9_{\pm 0.64}$	$32.5_{\pm 2.78}$	$9.1_{\pm 0.23}$	$73.7_{\pm 2.22}$	$\mathbf{9.0_{\pm 0.52}}$	$50.7_{\pm 3.56}$	$16.7_{\pm 1.20}$	$51.3_{\pm 4.21}$	$12.2_{\pm 0.67}$	$55.6_{\pm 4.36}$
Naïve GS	$10.4_{\pm 1.35}$	$59.0_{\pm 4.66}$	$18.7_{\pm 0.95}$	$23.3_{\pm 3.18}$	$11.4_{\pm 0.26}$	$44.3_{\pm 4.88}$	$11.2_{\pm 0.63}$	$48.3_{\pm 4.59}$	$20.2_{\pm 1.45}$	$17.2_{\pm 4.61}$	$14.4_{\pm 0.93}$	$38.4_{\pm 4.38}$
Ours	$\mathbf{9.1_{\pm 0.51}}$	$\mathbf{73.1_{\pm 4.73}}$	$\mathbf{11.3_{\pm 0.82}}$	$\mathbf{56.5_{\pm 5.77}}$	$\mathbf{7.9_{\pm 0.24}}$	$\mathbf{78.0_{\pm 3.97}}$	$10.2_{\pm 0.84}$	$\mathbf{51.7_{\pm 5.06}}$	$\mathbf{12.3_{\pm 0.89}}$	$\mathbf{67.7_{\pm 3.06}}$	$\mathbf{10.2_{\pm 0.66}}$	$\mathbf{65.4_{\pm 4.52}}$
ADVERSARIAL CONDITIONS												
Standard Tiled	$12.6_{\pm 0.74}$	$56.8_{\pm 4.83}$	$20.3_{\pm 2.10}$	$29.1_{\pm 7.05}$	$15.6_{\pm 0.86}$	$51.3_{\pm 4.52}$	$24.1_{\pm 1.58}$	$25.0_{\pm 3.95}$	$18.7_{\pm 2.04}$	$41.9_{\pm 2.68}$	$18.3_{\pm 1.46}$	$40.8_{\pm 4.61}$
DR Tiled	$11.5_{\pm 0.89}$	$57.6_{\pm 4.69}$	$\mathbf{13.8_{\pm 0.54}}$	$39.0_{\pm 2.54}$	$\mathbf{11.5_{\pm 0.61}}$	$57.7_{\pm 3.89}$	$14.4_{\pm 1.68}$	$39.1_{\pm 5.14}$	$18.6_{\pm 1.07}$	$42.7_{\pm 4.44}$	$14.0_{\pm 0.96}$	$47.2_{\pm 4.14}$
Naïve GS	$14.3_{\pm 0.85}$	$44.3_{\pm 2.30}$	$21.9_{\pm 0.85}$	$27.9_{\pm 4.85}$	$13.9_{\pm 1.25}$	$57.3_{\pm 4.29}$	$22.1_{\pm 1.39}$	$25.0_{\pm 3.35}$	$21.0_{\pm 1.52}$	$28.1_{\pm 5.95}$	$18.6_{\pm 1.17}$	$36.5_{\pm 4.15}$
Ours	$\mathbf{10.6_{\pm 0.38}}$	$\mathbf{60.6_{\pm 5.47}}$	$14.4_{\pm 0.77}$	$\mathbf{45.9_{\pm 4.58}}$	$12.2_{\pm 0.53}$	$\mathbf{62.7_{\pm 3.09}}$	$\mathbf{13.5_{\pm 0.99}}$	$\mathbf{46.5_{\pm 3.43}}$	$\mathbf{13.8_{\pm 0.78}}$	$\mathbf{65.6_{\pm 2.52}}$	$\mathbf{12.9_{\pm 0.69}}$	$\mathbf{56.3_{\pm 3.82}}$

TABLE III: Ablation study of pre-rasterization augmentations under nominal and adversarial conditions for pose estimation. We compare our method with models trained with specific augmentation groups removed. We report Average Distance of Model Points (ADD) in mm and a strict accuracy metric (< 10mm and $< 10^\circ$), averaged over 5 random seeds.

Objects	**Cube**		**3D Printed Toy**		**Rubber Duck**		**Tablet Bottle**		**Globe**		**Mean**	
Method	ADD	Accuracy	ADD	Accuracy	ADD	Accuracy	ADD	Accuracy	ADD	Accuracy	ADD	Accuracy
NOMINAL CONDITIONS												
w/o Random Noise	$11.9_{\pm 3.24}$	$61.5_{\pm 11.21}$	$13.5_{\pm 1.14}$	$37.2_{\pm 9.13}$	$8.2_{\pm 0.39}$	$\mathbf{80.9_{\pm 2.88}}$	$10.7_{\pm 0.84}$	$46.7_{\pm 3.33}$	$14.5_{\pm 1.17}$	$66.5_{\pm 5.32}$	$11.8_{\pm 1.36}$	$58.6_{\pm 6.37}$
w/o Spatial Clustering	$9.2_{\pm 0.91}$	$\mathbf{74.9_{\pm 8.49}}$	$12.9_{\pm 0.86}$	$44.7_{\pm 7.06}$	$8.0_{\pm 0.22}$	$73.5_{\pm 3.27}$	$16.3_{\pm 1.38}$	$39.3_{\pm 5.64}$	$13.8_{\pm 2.15}$	$51.3_{\pm 5.73}$	$12.0_{\pm 1.10}$	$56.7_{\pm 6.04}$
w/o Color Clustering	$9.2_{\pm 0.41}$	$71.0_{\pm 5.94}$	$13.6_{\pm 1.60}$	$49.4_{\pm 2.42}$	$8.2_{\pm 0.59}$	$77.4_{\pm 5.64}$	$20.4_{\pm 0.56}$	$34.3_{\pm 4.03}$	$14.5_{\pm 1.69}$	$64.8_{\pm 3.09}$	$13.2_{\pm 0.97}$	$59.4_{\pm 4.22}$
w/o Global Shift	$11.2_{\pm 1.26}$	$67.2_{\pm 3.85}$	$\mathbf{10.9_{\pm 1.13}}$	$55.7_{\pm 6.58}$	$9.1_{\pm 0.41}$	$70.9_{\pm 3.85}$	$39.5_{\pm 2.84}$	$10.3_{\pm 2.45}$	$13.9_{\pm 0.25}$	$52.1_{\pm 5.17}$	$16.9_{\pm 1.18}$	$51.2_{\pm 4.38}$
Ours	$\mathbf{9.1_{\pm 0.51}}$	$73.1_{\pm 4.73}$	$11.3_{\pm 0.82}$	$\mathbf{56.5_{\pm 5.77}}$	$\mathbf{7.9_{\pm 0.24}}$	$78.0_{\pm 3.97}$	$\mathbf{10.2_{\pm 0.84}}$	$\mathbf{51.7_{\pm 5.06}}$	$\mathbf{12.3_{\pm 0.89}}$	$\mathbf{67.7_{\pm 3.06}}$	$\mathbf{10.2_{\pm 0.66}}$	$\mathbf{65.4_{\pm 4.52}}$
ADVERSARIAL CONDITIONS												
w/o Random Noise	$13.2_{\pm 0.77}$	$49.7_{\pm 5.46}$	$15.8_{\pm 1.34}$	$40.7_{\pm 4.62}$	$12.9_{\pm 0.39}$	$52.7_{\pm 4.29}$	$13.9_{\pm 0.78}$	$41.5_{\pm 1.10}$	$16.3_{\pm 0.86}$	$58.4_{\pm 3.83}$	$14.4_{\pm 0.83}$	$48.6_{\pm 3.86}$
w/o Spatial Clustering	$13.5_{\pm 1.40}$	$46.2_{\pm 7.00}$	$16.4_{\pm 1.20}$	$39.3_{\pm 7.13}$	$14.0_{\pm 0.69}$	$51.7_{\pm 4.22}$	$17.6_{\pm 0.84}$	$35.6_{\pm 3.00}$	$16.2_{\pm 1.59}$	$39.6_{\pm 5.92}$	$15.5_{\pm 1.14}$	$42.5_{\pm 5.45}$
w/o Color Clustering	$14.4_{\pm 0.68}$	$47.3_{\pm 6.63}$	$16.7_{\pm 1.60}$	$41.2_{\pm 4.03}$	$14.8_{\pm 0.80}$	$46.7_{\pm 3.16}$	$19.9_{\pm 1.15}$	$29.4_{\pm 4.46}$	$16.2_{\pm 1.46}$	$58.9_{\pm 3.93}$	$16.4_{\pm 1.14}$	$44.7_{\pm 4.44}$
w/o Global Shift	$24.9_{\pm 1.29}$	$17.8_{\pm 3.16}$	$22.0_{\pm 1.54}$	$25.4_{\pm 2.54}$	$20.0_{\pm 1.89}$	$27.7_{\pm 1.70}$	$30.2_{\pm 1.63}$	$12.6_{\pm 2.39}$	$17.5_{\pm 0.52}$	$34.4_{\pm 2.42}$	$22.9_{\pm 1.37}$	$23.6_{\pm 2.44}$
Ours	$\mathbf{10.6_{\pm 0.38}}$	$\mathbf{60.6_{\pm 5.47}}$	$\mathbf{14.4_{\pm 0.77}}$	$\mathbf{45.9_{\pm 4.58}}$	$\mathbf{12.2_{\pm 0.53}}$	$\mathbf{62.7_{\pm 3.09}}$	$\mathbf{13.5_{\pm 0.99}}$	$\mathbf{46.5_{\pm 3.43}}$	$\mathbf{13.8_{\pm 0.78}}$	$\mathbf{65.6_{\pm 2.52}}$	$\mathbf{12.9_{\pm 0.69}}$	$\mathbf{56.3_{\pm 3.82}}$

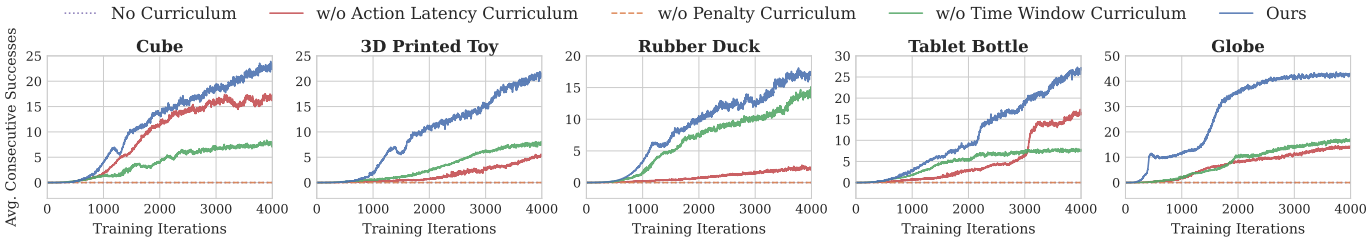

Fig. 5: Impact of performance-based curricula on training efficiency. Curves show learning progress for full curriculum compared with ablations with individual components removed. Note that *No Curriculum* and *w/o Penalty Curriculum* coincide at zero.

mentations leads to significant degradation. The removal of *Spatial Clustering* (3D means) and *Color Clustering* (SH0) drops accuracy to 42.5% and 44.7%, respectively, validating that correlated perturbations are essential for modeling localized effects. Most critically, the exclusion of *Global Shift* augmentations causes a catastrophic collapse in performance, with mean accuracy plummeting to just 23.6%. This underscores that simulating macro-level environmental changes is the single most important factor for generalizing to diverse real-world lighting conditions.

C. In-hand Reorientation Policy Learning

1) Resource-Efficient Policy Training: The teacher RL training phase learns in-hand manipulation skills efficiently, scaling across diverse object geometries while using minimal GPU resources. Policies are trained using 24,576 parallel environments. For primitive geometries (*e.g.*, the Cube), the training converges in 26 hours on a single consumer-grade NVIDIA RTX 4090 (24GB VRAM). More complex objects are trained on a dual-GPU setup, requiring 30 GB VRAM and converging in roughly 90 hours due to the increased simulation fidelity of complex contact geometry. The subsequent student-teacher distillation phase completes in 16 hours for $4,096$ environments on a single NVIDIA RTX 4090 GPU. Compared to prior work [6], which requires eight A40 GPUs over 60 hours, our pipeline achieves an order-of-magnitude improvement in VRAM efficiency and substantially reduced training time, making high-fidelity RL more accessible for real-world robotics.

2) Impact of Performance-based Curriculum: We evaluate the sample efficiency of our performance-based curriculum by selectively enabling different curriculum components (Section III-A3). Fig. 5 reports the consecutive successes averaged across all environments during training. Applying all the curriculum components leads to the fastest convergence and highest CS. Removing either the *Action Latency* or *Time Window* components significantly slows learning, particularly for complex geometries such as the *3D Printed Toy* and *Rubber*

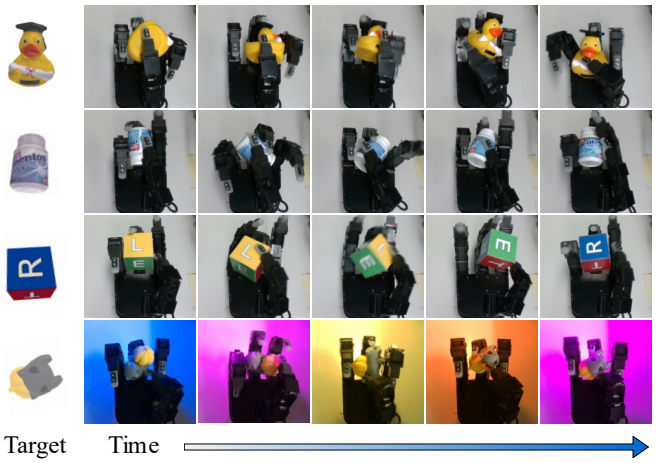

Fig. 6: Rollout sequence of the hand reorienting an object to the target pose. The images are from the robot camera feed.

Duck. Removing the *Regularization Penalty* curriculum causes complete failure, since the policy becomes more conservative and deprioritizes task completion. Similarly, having no curriculum also yields near-zero success. Beyond accelerating convergence, the curriculum also acts as a self-regulating mechanism for reward balancing, eliminating the need for per-object tuning. All teacher RL training and student distillation experiments use the same reward weights and hyperparameters across objects, demonstrating the approach's generality.

D. Hardware Deployment

Similar to the pose estimation experiments, we validate the sim-to-real deployment of the in-hand reorientation system under nominal and adversarial lighting, as shown in Fig. 6. Consistent with prior work [2, 6], we define successful in-hand reorientation when the object orientation error is within 0.4 radians. We measure *consecutive successes* (CS), *i.e.*, the number of goals reached before a failure occurs (object falls out of the hand). Table IV reports the consecutive successes (CS) averaged over five runs per object and lighting condition.

a) Comparison to Baselines: Under nominal lighting, our system achieves a mean of 37.6 consecutive reorientations across objects. It substantially outperforms the only prior vision-based baseline reported on hardware, DeXtreme [6], on the shared *Cube* object (35.4 vs. 27.8).

To further validate the necessity of our tailored perception pipeline, we replaced our estimator with FoundationPose [29] during deployment. This configuration resulted in near-total failure, achieving only 0.4 consecutive successes (CS) on average. We attribute this collapse to two factors: (i) FoundationPose operates at approximately 4 Hz, which is too slow for the rapid control loop compared to the $\sim$18 Hz throughput of our estimator, and (ii) frequent tracking loss caused by rapid object motion and severe finger occlusions inherent to in-hand manipulation. This experiment underscores that robust manipulation performance depends not only on policy quality but critically on high-frequency, occlusion-tolerant pose estimation.

TABLE IV: Real-world deployment on different objects.

	Consecutive Successes (Over Five Runs)		
Objects	DeXtreme [6]	Ours (Nominal)	Ours (Adversarial)
Cube	$27.8_{\pm 19.0}$	$35.4_{\pm 13.8}$	$25.6_{\pm 8.9}$
3D Printed Toy	-	$28.2_{\pm 12.6}$	$12.0_{\pm 6.9}$
Rubber Duck	-	$24.2_{\pm 15.3}$	$9.0_{\pm 5.0}$
Tablet Bottle	-	$12.6_{\pm 8.8}$	$4.2_{\pm 0.7}$
Globe	-	$87.6_{\pm 41.4}$	$76.2_{\pm 66.2}$
Mean	-	$37.6_{\pm 21.8}$	$25.4_{\pm 30.1}$

b) Object-Specific Performance: The learned policy demonstrates exceptional robustness on primitive geometries, with the *Globe* exceeding 200 CS in one of the trials. The performance extends to highly non-convex objects, such as the *3D Printed Toy* and *Rubber Duck*, obtaining more than 20 CS on average. We qualitatively observe smoother manipulation behaviors with compliant objects (e.g., *Globe*, *Duck*), suggesting that inherent material damping aids stability despite not being explicitly modeled in simulation. However, we observe a notable sim-to-real gap for the *Tablet Bottle*. We attribute this degradation to unmodeled friction effects, particularly the extremely low surface friction introduced by its label.

c) Robustness to Adversarial Lighting: Under severe adversarial lighting conditions, the policy maintained an average of consecutive successes of $\sim$25. To our knowledge, this is the first demonstration of sustained dexterous in-hand manipulation under such extreme visual perturbations. Although performance decreases relative to nominal lighting, the decline highlights the tight coupling between perception and control. The reduction in overall task success disproportionately exceeds the degradation in pose estimation accuracy (Section IV-A). This indicates that even small perceptual errors can compound into significant control failures.

V. Conclusion

In this work, we presented a framework for robust in-hand reorientation using monocular RGB vision. By integrating 3D Gaussian Splatting directly into the simulation loop, we substantially reduced the computational cost of high-fidelity rendering. Our novel *pre-rasterization augmentations* introduce structured diversity into static Gaussian scenes, effectively bridging the visual sim-to-real gap. This approach enabled the training of a perception module for object pose estimation that is robust to both nominal and severe adversarial lighting conditions. Deploying the resulting system on a multi-fingered hand, we achieved an average of over 25 consecutive successful reorientations across diverse object geometries, even under adversarial lighting.

Our findings underscore a critical insight for the field: the primary bottleneck in real-world dexterity often lies less in control complexity and more in perceptual fidelity. We show that when vision is modeled with sufficient physical grounding, it can support complex and precise manipulation tasks, previously dominated by proprioceptive or tactile-based approaches. Nevertheless, vision is fundamentally constrained by finger occlusions and its inability to directly sense contact

forces. A promising direction for future work is the integration of dense visual feedback with high-frequency tactile sensing to better handle unmodeled surface properties. Moreover, while our method excels at instance-specific manipulation, extending this framework toward broader object generalization will require reducing reliance on predefined object frames and incorporating richer, expressive geometric representations.

Acknowledgments

This work was funded by ETH Zurich (research grant no. 22-2 ETH-47), Swiss National Science Foundation (NCCR Automation grant no. 51NF40_225155), Swiss Federal Railways (SBB) via ETH Mobility Initiative, ETHAR and by grants from NVIDIA. The authors also acknowledge the use of NVIDIA RTX 6000 Ada Generation GPUs, which facilitated this research.

The authors would like to thank René Zurbrügg for insightful discussions regarding pose estimator training and system identification, Pascal Roth for the initial integration of Gaussian splatting rendering with IsaacLab, and Elena Krasnova for assistance with the robotic hardware.

References

[1] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[2] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39 (1):3–20, 2020.

[3] Aditya Bhatt, Adrian Sieler, Steffen Puhlmann, and Oliver Brock. Surprisingly robust in-hand manipulation: An empirical study. *Robotics: Science and Systems*, 2021.

[4] Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84):eadc9244, 2023.

[5] Timothy Chen, Ola Shorinwa, Joseph Bruno, Aiden Swann, Javier Yu, Weijia Zeng, Keiko Nagami, Philip Dames, and Mac Schwager. Splat-nav: Safe real-time robot navigation in gaussian splatting maps, 2024. URL https://arxiv.org/abs/2403.02751.

[6] Ankur Handa, Arthur Allshire, Viktor Makoviychuk, Aleksei Petrenko, Ritvik Singh, Jingzhou Liu, Denys Makoviichuk, Karl Van Wyk, Alexander Zhurkevich, Balakumar Sundaralingam, et al. Dextreme: Transfer of agile in-hand manipulation from simulation to reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5977–5984. IEEE, 2023.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 770–778, 2016.

[8] Huajian Huang, Longwei Li, Cheng Hui, and Sai-Kit Yeung. Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular, stereo, and rgb-d cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[9] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

[10] Yongseok Lee, Hyunsu Kim, Harim Ji, Jinuk Heo, Youngseon Lee, Jiseock Kang, Jeongseob Lee, and Dongjun Lee. Human-in-the-loop gaussian splatting for robotic teleoperation. *IEEE Robotics and Automation Letters*, 11(1):105–112, 2026. doi: 10.1109/LRA.2025.3632755.

[11] Xinhai Li, Jialin Li, Ziheng Zhang, Rui Zhang, Fan Jia, Tiancai Wang, Haoqiang Fan, Kuo-Kun Tseng, and Ruiping Wang. Robogsim: A real2sim2real robotic gaussian splatting simulator, 2024. URL https://arxiv.org/abs/2411.11839.

[12] Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. Gaussian Splatting SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[13] Jonathan Michaux, Seth Isaacson, Challen Enninful Adu, Adam Li, Rahul Kashyap Swayampakula, Parker Ewen, Sean Rice, Katherine A. Skinner, and Ram Vasudevan. Let's make a splan: Risk-aware trajectory optimization in a normalized gaussian splat. *IEEE Transactions on Robotics*, pages 1–19, 2025. doi: 10.1109/TRO.2025.3584559.

[14] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022. doi: 10.1126/scirobotics.abk2822. URL https://www.science.org/doi/abs/10.1126/scirobotics.abk2822.

[15] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: a framework and survey. *J. Mach. Learn. Res.*, 21(1), January 2020. ISSN 1532-4435.

[16] NVIDIA, :, Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, Lukasz Wawrzyniak, Milad Rakhsha, Alain Denzler, Eric Heiden, Ales Borovicka, Ossama Ahmed, Iretiayo Akinola, Abrar Anwar, Mark T. Carlson, Ji Yuan Feng, Animesh Garg, Renato Gasoto, Lionel Gulich, Yijie Guo, M. Gussert, Alex Hansen, Mihir Kulkarni, Chenran Li, Wei Liu, Viktor Makoviychuk, Grzegorz Malczyk, Hammad Mazhar, Masoud Moghani, Adithyavairavan Murali, Michael Noseworthy, Alexander Poddubny, Nathan

Ratliff, Welf Rehberg, Clemens Schwarke, Ritvik Singh, James Latham Smith, Bingjie Tang, Ruchik Thaker, Matthew Trepte, Karl Van Wyk, Fangzhou Yu, Alex Millane, Vikram Ramasamy, Remo Steiner, Sangeeta Subramanian, Clemens Volk, CY Chen, Neel Jawale, Ashwin Varghese Kuruttukulam, Michael A. Lin, Ajay Mandlekar, Karsten Patzwaldt, John Welsh, Huihua Zhao, Fatima Anes, Jean-Francois Lafleche, Nicolas Moënne-Loccoz, Soowan Park, Rob Stepinski, Dirk Van Gelder, Chris Amevor, Jan Carius, Jumyung Chang, Anka He Chen, Pablo de Heras Ciechomski, Gilles Daviet, Mohammad Mohajerani, Julia von Muralt, Viktor Reutskyy, Michael Sauter, Simon Schirm, Eric L. Shi, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. 2025. URL https://arxiv.org/abs/2511.04831.

[17] Johannes Pitz, Lennart Röstel, Leon Sievers, Darius Burschka, and Berthold Bäuml. Learning a shape-conditioned agent for purely tactile in-hand manipulation of various objects. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13112–13119. IEEE, 2024.

[18] Polycam Inc. Polycam - lidar & 3d scanner for iphone and android, 2024. URL https://poly.cam/. Accessed: 2024-05-20.

[19] Haozhi Qi, Brent Yi, Sudharshan Suresh, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. General In-Hand Object Rotation with Vision and Touch. In *Conference on Robot Learning (CoRL)*, 2023.

[20] Mohammad Nomaan Qureshi, Sparsh Garg, Francisco Yandun, David Held, George Kantor, and Abhishesh Silwal. Splatsim: Zero-shot sim2real transfer of rgb manipulation policies using gaussian splatting, 2024. URL https://arxiv.org/abs/2409.10161.

[21] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

[22] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[23] Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022.

[24] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017. URL https://api.semanticscholar.org/CorpusID:28695052.

[25] 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.

[26] Ritvik Singh, Jason Jingzhou Liu, Karl Van Wyk, Yu-Wei Chao, Jean-Francois Lafleche, Florian Shkurti, Nathan Ratliff, and Ankur Handa. Synthetica: Large scale synthetic data generation for robot perception. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7810–7817. IEEE, 2025.

[27] Jos M. F. ten Berge. The rigid orthogonal procrustes rotation problem. *Psychometrika*, 71(1):201–205, 2006. doi: 10.1007/s11336-004-1160-5.

[28] Jiaxu Wang, Qiang Zhang, Jingkai Sun, Jiahang Cao, Gang Han, Wen Zhao, Weining Zhang, Yecheng Shao, Yijie Guo, and Renjing Xu. Reinforcement learning with generalizable gaussian splatting. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 435–441, 2024. doi: 10.1109/IROS58592.2024.10801348.

[29] Bowen Wen, Wei Yang, Jan Kautz, and Stanley T. Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17868–17879, 2023. URL https://api.semanticscholar.org/CorpusID:266191252.

[30] Maximum Wilder-Smith, Vaishakh Patil, and Marco Hutter. Radiance fields for robotic teleoperation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13861–13868, 2024. doi: 10.1109/IROS58592.2024.10801345.

[31] Yuxuan Wu, Lei Pan, Wenhua Wu, Guangming Wang, Yanzi Miao, Fan Xu, and Hesheng Wang. Rl-gsbridge: 3d gaussian splatting based real2sim2real method for robotic manipulation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 192–198, 2025. doi: 10.1109/ICRA55743.2025.11128103.

[32] Max Yang, chenghua lu, Alex Church, Yijiong Lin, Christopher J. Ford, Haoran Li, Efi Psomopoulou, David A.W. Barton, and Nathan F. Lepora. Anyrotate: Gravity-invariant in-hand object rotation with sim-to-real touch. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=8Yu0TNJNGK.

[33] Zhao-Heng Yin, Binghao Huang, Yuzhe Qin, Qifeng Chen, and Xiaolong Wang. Rotating without seeing: Towards in-hand dexterity through touch. *Robotics: Science and Systems*, 2023.