

# EgoPush: Learning End-to-End Egocentric Multi-Object Rearrangement for Mobile Robots

Boyuan An, Zhexiong Wang\*, Yipeng Wang\*, Jiaqi Li, Sihang Li†, Jing Zhang†, Chen Feng†  
New York University

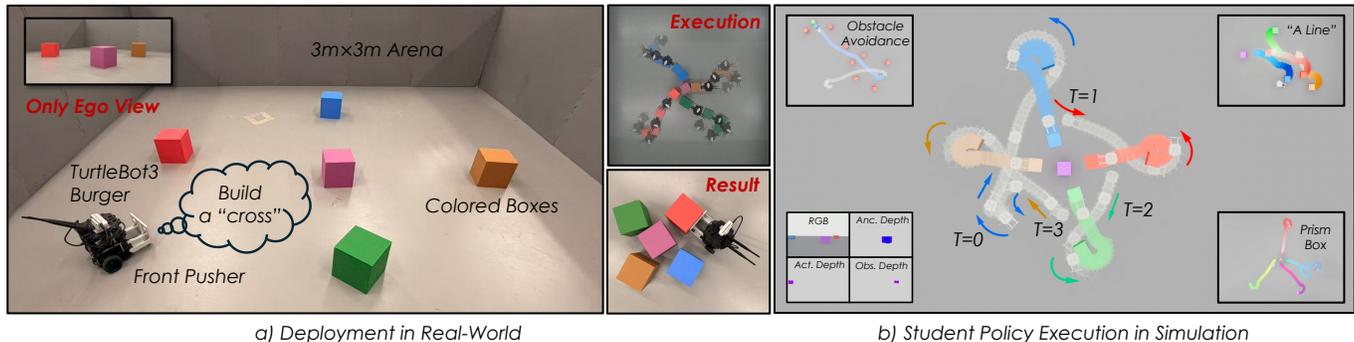


Fig. 1: **EgoPush**: From only egocentric observations, a mobile robot builds goal object configurations in the real world (left) and demonstrates structured behaviors in simulation (right), including obstacle avoidance, forming diverse target configurations (e.g., cross/line), and handling objects with varied geometries (cube/prism/cylinder).

**Abstract**—Humans can rearrange objects in cluttered environments using egocentric perception, navigating occlusions without global coordinates. Inspired by this capability, we study long-horizon multi-object non-prehensile rearrangement for mobile robots using a single egocentric camera. We introduce EgoPush, a policy learning framework that enables egocentric, perception-driven rearrangement without relying on explicit global state estimation that often fails in dynamic scenes. EgoPush designs an object-centric latent space to encode relative spatial relations among objects, rather than absolute poses. This design enables a privileged reinforcement-learning (RL) teacher to jointly learn latent states and mobile actions from sparse keypoints, which is then distilled into a purely visual student policy. To reduce the supervision gap between the omniscient teacher and the partially observed student, we restrict the teacher’s observations to visually accessible cues. This induces active perception behaviors that are recoverable from the student’s viewpoint. To address long-horizon credit assignment, we decompose rearrangement into stage-level subproblems using temporally decayed, stage-local completion rewards. Extensive simulation experiments demonstrate that EgoPush significantly outperforms end-to-end RL baselines in success rate, with ablation studies validating each design choice. We further demonstrate zero-shot sim-to-real transfer on a mobile platform in the real world. Code and videos can be found at <https://ai4ce.github.io/EgoPush/>.

## I. INTRODUCTION

Humans routinely perform long-horizon rearrangement in cluttered spaces without consulting precise *global* coordinates. Instead, they maintain an egocentric sense of *relative* spatial relations and continuously coordinate motion to keep task-relevant cues in view under intermittent occlusions. For instance, one can reposition several chairs into a symmetric

arrangement around a table by repeatedly stepping around to re-check alignment when chairs occlude one another.

Replicating this capability on robotic platforms is highly impactful, yet far from trivial. We introduce a controlled testbed: multi-object non-prehensile rearrangement from purely egocentric visual observations. The task challenges a mobile robot to operate over long horizons: it must navigate and physically nudge multiple robot-scale objects precisely into predefined arrangements relative to an anchor object (e.g., cross-shaped or linear patterns in Fig. 1). Our environment is intentionally texture sparse, containing no external landmarks or visual references beyond the objects themselves.

This setting poses several coupled challenges for egocentric mobile rearrangement: texture-sparse observations provide no reliable landmarks. During contact-rich interaction, objects are continuously displaced and frequently occlude one another, further challenging egocentric perception. Consequently, the agent cannot assume reliable global localization or external references, and must instead coordinate actions using only perceived relative relations between the robot and objects.

Existing methodologies struggle in this setting, as most non-prehensile manipulation frameworks rely on global state or external tracking [2, 42, 26, 11, 23]. Model-based approaches are particularly brittle: texture-sparse scenes challenge SLAM or visual odometry *to maintain consistent localization during object motion*; and during contact-rich interactions, the continual object motion violates the static-world assumptions commonly used in geometric mapping. Without stable global localization, planners lack a consistent reference frame and can degrade substantially. In contrast, standard end-to-end reinforcement learning (RL) directly [15, 32, 38, 25] from egocentric pixels avoids explicit mapping, but often suffers

\*Equal Contribution

†Corresponding Authors, {s110496, z.jing, cfeng}@nyu.edu

from poor sample efficiency and brittleness under partial observability caused by occlusions. When key cues repeatedly disappear from view, policies can fail to maintain coherent strategies for egocentric mobile manipulation.

A natural direction to improve sample efficiency is cross-modal distillation: training a privileged teacher with low-dimensional environment states via online RL, then distilling its behavior into an egocentric visual student [47, 43, 39, 24, 9, 1, 6, 40]. However, in our setting, this direction faces two core challenges: (1) the observability gap in distillation, and (2) long-horizon credit assignment. First, conventional pipeline typically optimizes an omniscient teacher under global ground-truth observations [47, 43, 39] and uses its behavior as supervision. In our setting, this can be counterproductive: because perception and action are tightly coupled, an all-seeing teacher may execute trajectories that are irrecoverable from the student’s egocentric view—for example, moving backward toward a target object behind the robot without turning to bring the object into view, yielding action labels that are inexplicable given the student’s limited field of view. The student then receives inconsistent supervision, becomes brittle, and fails to ground its policy in visually accessible evidence.

Beyond the observability gap, the long-horizon nature of multi-object tasks presents another bottleneck: temporal credit assignment. Without explicit temporal decomposition, a vanilla RL formulation treats these tasks as monolithic sequences, where sparse rewards fail to distinguish efficient maneuvers from aimless exploration. As the episode unfolds, the learning signal becomes increasingly delayed and attenuated, making it difficult for the agent to acquire the precise, tightly coordinated behaviors required for task completion. Consequently, without an explicit mechanism to provide dense and comparable progress signals throughout the entire sequence, optimization can fail or converge to suboptimal strategies.

To address these challenges, we present EgoPush, a policy learning framework for long-horizon multi-object rearrangement from egocentric vision. It learns object-centric representations that capture task-relevant *relative* spatial relations among the objects, enabling rearrangement without explicit global state estimation. To make teacher–student distillation viable under partial observability, we introduce Constrained Teacher RL, which limits the privileged teacher to egocentric, visibility-limited observations so that its behavior remains reproducible from the student’s viewpoint and induces learnable active perception. Finally, it improves temporal credit assignment for long-horizon tasks via stage-wise training with temporally decayed, stage-local completion rewards.

We evaluate EgoPush in extensive simulation benchmarks and further demonstrate successful zero-shot sim-to-real transfer on a Turtlebot mobile platform. Across baseline comparisons and ablation studies, EgoPush consistently improves both success rate and sample efficiency over alternative designs. We achieve robust transfer by explicitly modeling depth noise in simulation and mitigating sensor noise on real hardware.

To summarize our contributions of EgoPush, a policy training framework using cross-modal teacher–student distillation:

- It enables multi-object rearrangement from purely egocentric vision without relying on explicit maps or global positioning, by learning object-centric latent representations that encode relative spatial relations;
- It induces learnable active perception behaviors that jointly handle mobility, perception, and object interaction, by constraining the privileged teacher to egocentric, visibility-limited observations;
- It facilitates long-horizon learning with improved temporal credit assignment through a stage-aligned reward that decomposes complex tasks into manageable subproblems;
- Its key designs are validated through extensive baseline comparisons and ablation studies, with successful zero-shot sim-to-real transfers.

## II. RELATED WORKS

### A. *Non-prehensile Mobile Manipulation*

Non-prehensile manipulation is a practical cornerstone for mobile robots in clutter, as it allows interaction with objects that are infeasible to grasp due to size, weight, density, or accessibility constraints. For example, it can facilitate *navigation* by pushing aside movable obstacles to unlock a traversable passage or reduce local congestion [45, 11, 12], accomplish *rearrangement* by relocating objects to satisfy task goals [2], and perform *reconfiguration* by driving an object into a more favorable pose for subsequent operations (*e.g.*, improving stability or exposing a useful face) [19, 24].

Classic solutions typically frame non-prehensile rearrangement as planning/optimization under contact dynamics. Song et al. [35] formulates planar sorting with Monte Carlo Tree Search to reason over contact-induced transitions, while Ren et al. [30] structures the problem from an object-centric viewpoint to facilitate decision-making in cluttered scenes. Hybrid pipelines further combine task-and-motion planning with learned non-prehensile skills to handle complex interaction modes [20]; other works [18, 27] incorporate additional sensing modalities (*e.g.*, force or tactile feedback) to support pushing with a mobile base. Despite their strong long-horizon performance, these systems commonly assume access to accurate robot/object states or an explicit state estimator—requirements that can be brittle or infeasible when only egocentric sensing is available.

Vision-based learning improves generalization across object properties [11, 5], yet many non-prehensile rearrangement pipelines still rely on global state. Wu et al. [41] relies on top-down reconstructions together with a known global map to build spatial action representations. Multi-agent pushing is likewise commonly studied under global-information assumptions to enable coordination [36, 37]. Other systems explicitly introduce privileged viewpoints, such as aerial bird’s-eye-view guidance for ground robots [21]. Even in purely “visual” rearrangement, some setups effectively provide global coverage by keeping all objects within the camera frustum [16]. Consequently, a key open challenge remains: achieving long-horizon non-prehensile rearrangement under *genuinely* partial

observability, where egocentric views are narrow, occlusions are frequent, and global state is not accessible.

### B. Visual Policy Learning of Mobile Robots

Learning robot control from visual observations is a long-standing direction for reducing reliance on explicit state estimation. A straightforward paradigm is end-to-end RL [15, 32, 38, 25], which maps high-dimensional images (often fused with proprioception) directly to actions. Although RL has made notable progress in state-based learning of mobile robots [7, 8, 3, 44], pixel-based RL learning remains sample-inefficient and sensitive to reward sparsity and partial observability, motivating a wide range of techniques to improve stability and data efficiency (*e.g.*, stronger visual encoders [46, 13, 10], asymmetric actor-critic [14], parallel differentiable simulation [48], using low-dimensional visual features [17], etc.). However, contact-rich mobile manipulation introduces a more severe form of partial observability. Visual signals are inherently sparse during execution: pushing actions not only result in frequent self-occlusions but also often displace objects outside the camera frustum. This lack of persistent visual grounding causes purely reactive pixel-based policies to fail, as they overfit to transient cues and lack the robustness to recover from prolonged periods of visual vacancy.

A complementary and often more data-efficient pipeline is privileged RL teacher–visual student distillation [47, 43, 39, 24, 9, 1, 6, 40]. The common recipe is to train a teacher policy with access to privileged signals available in simulation—*e.g.*, global ground-truth state [47, 43, 39], noiseless geometry [24, 9, 1], or full-scene information unaffected by occlusion [6, 40]—and then transfer its behavior to a visual student via imitation learning (behavior cloning) or interactive variants (DAgger [31]). This strategy has been effective across various visuomotor control problems because the teacher provides stable supervision that mitigates high-variance credit assignment from raw images. Nevertheless, such benefits typically rely on an implicit assumption: the teacher’s demonstrations are *recoverable* from the student’s observation stream. For egocentric mobile manipulation tasks where perception and action are tightly coupled, an omniscient teacher optimized in the full state space can exploit off-screen or globally available information to execute trajectories that provide little visual evidence for a camera-based student. As a result, the student may face an ill-posed imitation problem, where multiple teacher actions are consistent with the same partial observation, leading to ambiguous supervision and brittle downstream policies.

## III. METHOD

In this section, we first introduce an object-centric latent representation that abstracts the scene into task-relevant roles (Sec. III-A), serving as a shared interface for both teacher and student policies. Then we introduce EgoPush, a two-phase distillation framework for long-horizon, multi-object non-prehensile rearrangement under egocentric observations. Phase 1 performs online reinforcement learning un-

der constrained privileged observations to train the teacher (Sec. III-B). Phase 2 distills the teacher’s behavior and intermediate representations into an egocentric visual student via imitation learning (Sec. III-C).

### A. Object-Centric Latent Representation

For this rearrangement task, not all objects are equally important for decision-making. Within a pushing episode, the policy primarily needs to attend to the currently manipulated object (**active object**) and the object that indicates the target (**anchor**), while treating all remaining objects as obstacles (**obstacle**). Therefore, we partition scene objects into three semantic categories and encode them with a shared-weight state estimator. This produces group-wise latent embeddings, which we concatenate to form a **object-centric latent representation**. Weight sharing aligns the geometric features of different groups in a common embedding space, enabling the policy to reason over *relative spatial relations between group latents* rather than modeling each object in isolation.

### B. Phase 1 (Teacher): Online Reinforcement Learning

**Action Space.** The robot is a differential-drive base, and the policy outputs a 2D continuous action  $a_t = [v_t, \omega_t]$ .  $v_t$  is the base linear velocity and  $\omega_t$  is the angular velocity, which are then converted into left/right wheel velocities via differential-drive kinematics and executed by PD controller.

**Privileged Information: Sparse Keypoints.** The teacher observes sparse keypoints [43, 39] for the active object, anchor, and obstacles that capture object geometry and their relative poses, which substantially reduces observation dimensionality and improves sample efficiency, while remaining sufficient for contact-rich pushing decisions. Since the objective is to push the active object near the anchor and align its pose, we additionally provide a target reference keypoint set  $P_t^{\text{ref}}$  sampled from the keypoints of the predefined target configuration; for each task category, the relative pose between the anchor and the reference target is invariant across episodes, so  $P_t^{\text{ref}}$  serves as a consistent spatial guide for pose alignment.

**Constrained Observations.** To ensure the teacher trajectories rely only on task-relevant information that is recoverable from the egocentric viewpoint and thus reproducible by the student, we impose two constraints. (1) **Virtual egocentric FOV masking:** we define a robot-pose-based viewing frustum and uniformly mask points outside the frustum or beyond a maximum range. This approximates the camera-visible field of view. (2) **Center-gated visibility for  $P_t^{\text{ref}}$ :**  $P_t^{\text{ref}}$  is visible only when the anchor lies within the virtual FOV and lies inside a central gated region in the view, *i.e.*, its normalized image-plane coordinates within  $(\pm u_{\text{gate}}, \pm v_{\text{gate}})$ . Otherwise,  $P_t^{\text{ref}}$  is masked to prevent the policy from exploiting the reference while ignoring the anchor. Under these constraints, the teacher observes grouped sparse keypoints  $\{P_t^{\text{act}}, P_t^{\text{anc}}, P_t^{\text{obs}}, P_t^{\text{ref}}\}$  and the previous action  $a_{t-1}$ .

**Rewards.** Our rewards include: (1) **Stage-aligned completion reward:** In long-horizon sequential rearrangement, the agent repeatedly solves similar subproblems (reach the active

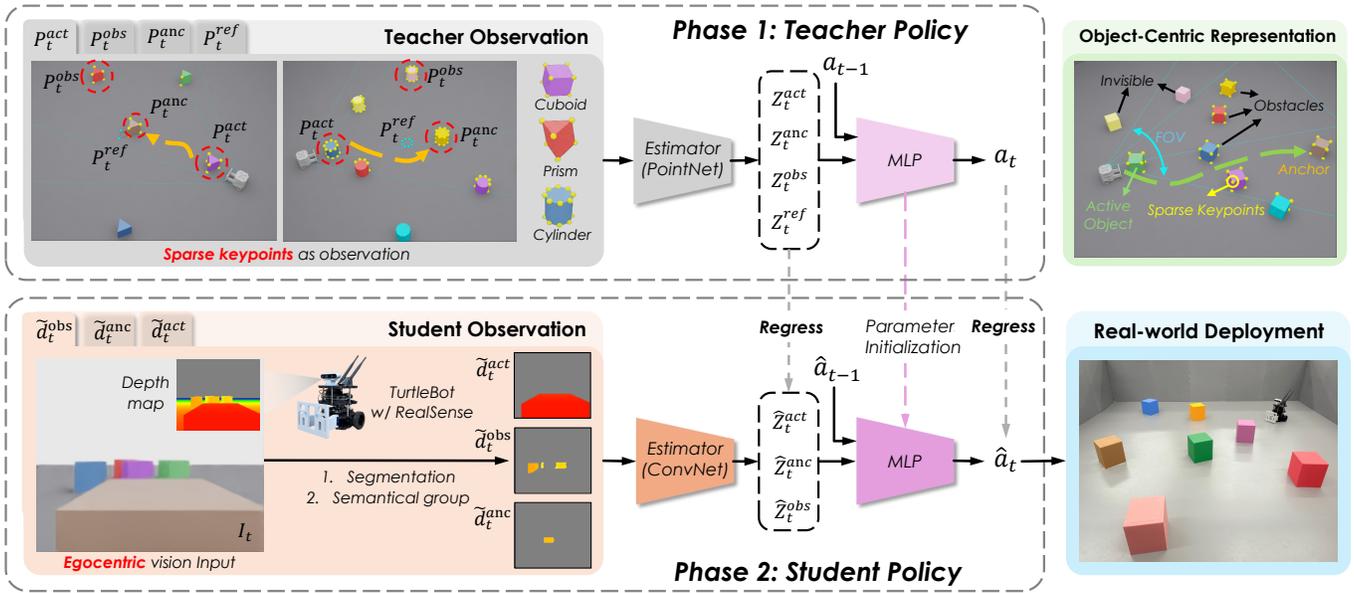


Fig. 2: **EgoPush Overview**. EgoPush is a two-phase learning framework for long-horizon, multi-object non-prehensile rearrangement under egocentric observations: in Phase 1, a privileged teacher policy is trained from sparse keypoints while enforcing egocentric, visibility-limited sensing so its behaviors remain visually recoverable; in Phase 2, an egocentric student uses RGB only for instance grouping and receives group-wise depth inputs, and is distilled online from the teacher via latent and action regression, enabling zero-shot sim-to-real deployment on a TurtleBot with a RealSense camera.

object  $\rightarrow$  place it near anchor) under delayed and sparse feedback. Consequently, a binary completion reward provides limited guidance and tends to blur efficient and inefficient solutions. We address this by (i) imposing a per-stage step budget  $T_s$ , and (ii) weighting completion rewards by the fraction of remaining budget, yielding stage-aligned supervision. Concretely, the *reach* completion reward is

$$r_{\text{reach},t} = \eta_t \cdot \mathbb{I}[\|\mathbf{p}_{\text{rbt},t} - \mathbf{p}_{\text{act},t}\|_2 < \varepsilon_{\text{reach}}], \quad (1)$$

where  $\mathbf{p}_{\text{rbt},t}$  and  $\mathbf{p}_{\text{act},t}$  denote the 2D positions of the robot and the active object at time  $t$ ,  $\varepsilon_{\text{reach}}$  is the reaching threshold, and  $\mathbb{I}[\cdot]$  is the indicator function that equals 1 if the condition holds and 0 otherwise. We set

$$\eta_t = \frac{T_s - \tau_t}{T_s + \epsilon_0}, \quad (2)$$

where  $\tau_t$  is the stage-local elapsed steps (reset at each stage boundary), and  $\epsilon_0$  is a small constant for numerical stability. This time-weighted completion reward shortens effective credit assignment within a stage and provides a consistent incentive for timely completion, which empirically stabilizes optimization and improves success in our long-horizon setting. Similarly, the *place* completion reward is

$$r_{\text{place},t} = \eta_t \mathbb{I} \left[ \begin{array}{l} \|\mathbf{p}_{\text{act},t} - \mathbf{p}_{\text{ref},t}\|_2 < \varepsilon_{\text{align}} \quad \text{and} \\ |\Delta\psi_t| < \varepsilon_{\Phi} \quad \text{and} \\ \|\mathbf{v}_{\text{act},t}\|_2 < \varepsilon_v \end{array} \right], \quad (3)$$

where  $\mathbf{p}_{\text{ref},t}$  is the 2D reference target position (defined by the virtual  $P_t^{\text{ref}}$ ),  $\varepsilon_{\text{align}}$  is the position-alignment threshold,  $\Delta\psi_t$  is the yaw error between the active object and its target orientation, and  $\mathbf{v}_{\text{act},t}$  denotes the planar velocity vector of the active object. The thresholds  $(\varepsilon_{\Phi}, \varepsilon_v)$  ensure that the reward

is only granted when the object is both precisely oriented and physically stable at the target location. (2) **Progress shaping (Phase-Gated Distance Decrease)**:

$$r_{\text{dist},t} = w_{\text{rbt}} (d_{\text{rbt},t-1} - d_{\text{rbt},t}) \mathbb{I}[g_t = 0] + w_{\text{ref}} (d_{\text{ref},t-1} - d_{\text{ref},t}) \mathbb{I}[g_t = 1], \quad (4)$$

where  $g_t \in \{0, 1\}$  is a phase gate within each stage:  $g_t=0$  denotes the reach phase and  $g_t=1$  denotes the place phase, switching to 1 at the reach event. The distances are defined as  $d_{\text{rbt},t} = \|\mathbf{p}_{\text{rbt},t} - \mathbf{p}_{\text{act},t}\|_2$  and  $d_{\text{ref},t} = \|\mathbf{p}_{\text{act},t} - \mathbf{p}_{\text{ref},t}\|_2$ , with weights  $(w_{\text{rbt}}, w_{\text{ref}})$ . (3) **Smoothness**: we penalize sudden changes between consecutive actions to smooth the policy output. (4) **Slowdown Near Target**: we encourage the object to settle by rewarding low object speed when it is close to the reference target. Full definitions and hyperparameters are provided in the supplementary materials.

**Termination.** (1) Task success: The episode terminates upon task success; We also apply early-termination conditions to improve training stability and prune unproductive explorations: (2) Outside: the robot moves outside a predefined world box; (3) Collision: an obstacle is collided with by the robot or the active box. We avoid a dense collision penalty to avoid reward hacking and tedious weight tuning; early termination truncates future returns and thus provides a strong negative signal; (4) Time limit:  $\tau_t$  in Eq. 2 reaches its budget  $T_s$ .

**EgoPush-Teacher.** We use a PointNet [28] as the teacher state estimator. Due to occlusions and FOV cropping, the number of visible points varies over time; PointNet handles variable-sized point sets via a shared per-point MLP and symmetric pooling, making it permutation-invariant. At time  $t$ , let  $P_t^k$  denote the visible point set of semantic group  $k \in \{\text{act}, \text{anc}, \text{obs}, \text{ref}\}$ .

We encode each group into a group-wise latent  $Z_t^k \in \mathbb{R}^d$ :

$$Z_t^k = \text{PointNet}_\theta(P_t^k), \quad k \in \{\text{act}, \text{anc}, \text{obs}, \text{ref}\}. \quad (5)$$

We then pass them together with the previous action  $a_{t-1}$  to a MLP:

$$a_t = \text{MLP}(a_{t-1}, Z_t^{\text{act}}, Z_t^{\text{anc}}, Z_t^{\text{obs}}, Z_t^{\text{ref}}). \quad (6)$$

We train the teacher policy using Proximal Policy Optimization (PPO) [33] and apply domain randomization to key physical parameters.

### C. Phase 2 (Student): Supervised Learning

**Observation.** The student is equipped with an egocentric RGB-D camera. RGB is used only to obtain instance masks for semantic grouping while the policy network takes depth-only inputs. Specifically, we run instance-level segmentation  $S_{inst}(\cdot)$  on the RGB image  $I_t^{\text{rgb}} \in \mathbb{R}^{H \times W \times 3}$  to obtain a binary mask  $M_t^{(i)}$  for each visible object instance  $i$ , and assign each instance to a semantic group  $\mathcal{I}_t^k$ , based on its task role (active / anchor / obstacle). We then apply these masks to the depth map  $d_t \in \mathbb{R}^{H \times W \times 1}$  and aggregate instances within each group by pixel-wise summation, producing three fixed-dimensional depth layers  $\tilde{d}_t^k$ , as shown in Fig. 2:

$$\{M_t^{(i)}\}_{i=1}^{N_t} = S_{inst}(I_t^{\text{rgb}}), \quad (7)$$

$$\tilde{d}_t^{(i)} = M_t^{(i)} \odot d_t \in \mathbb{R}^{H \times W \times 1}, \quad (8)$$

$$\tilde{d}_t^k = \sum_{i \in \mathcal{I}_t^k} \tilde{d}_t^{(i)}, \quad \forall k \in \{\text{act}, \text{anc}, \text{obs}\}, \quad (9)$$

where  $N_t$  is the number of visible instances at time  $t$ .

Since instance masks  $\{M_t^{(i)}\}$  are spatially disjoint, the summation in Eq. 9 aggregates all instances within the same semantic category into a single depth layer  $\tilde{d}_t^k$  without pixel-wise interference, ensuring a constant input dimensionality for the visual backbone. Recent progress in zero-shot segmentation models [29, 34] suggests that obtaining such masks from RGB can be reliable in real scenes; however, to focus on the core problems, we use color-coded objects and HSV-threshold segmentation in our experiments (see Appendix D). We further address the sim-to-real gap of depth by simulating the noise pattern when training and applying the Navier-Stokes inpainting algorithm [4, 49] to denoise in real world.

**EgoPush-Student.** For the student, we use a CNN as the state estimator to encode the masked depth for each semantic group; the rest of the network architecture is kept the same as the teacher (i.e., the same MLP policy head):

$$\hat{Z}_t^k = \text{CNN}_\phi(\tilde{d}_t^k), \quad k \in \{\text{act}, \text{anc}, \text{obs}\}, \quad (10)$$

$$\hat{a}_t = \text{MLP}(a_{t-1}, \hat{Z}_t^{\text{act}}, \hat{Z}_t^{\text{anc}}, \hat{Z}_t^{\text{obs}}). \quad (11)$$

We avoid pure BC to minimize the mean squared error (MSE) between student and teacher actions, because contact-rich tasks such as pushing are extremely sensitive to closed-loop errors: once the student exhibits a small deviation early on, subsequent states quickly drift away from the teacher’s

demonstration distribution, making pure BC difficult to recover; instead, we adopt an online DAgger-style distillation procedure: in each iteration, we query the teacher online at the currently visited states to produce action labels, and the student immediately performs one supervised parameter update.

Beyond supervised action cloning, we introduce a relational distillation loss to bridge the representation gap between the privileged PointNet-based teacher and the vision-based student. Given the teacher’s latent space contains a privileged reference embedding  $Z_t^{\text{ref}}$ , whereas the student must operate solely on egocentric observations  $\{\hat{Z}_t^{\text{act}}, \hat{Z}_t^{\text{anc}}, \hat{Z}_t^{\text{obs}}\}$  without explicit target indicators. Due to this dimensional and structural misalignment, a direct global MSE loss between the teacher and student latent spaces is mathematically ill-defined.

To bridge this gap, we propose to align the invariant relational structure between the common semantic entities. Since the relative pose between the anchor and the reference target is invariant across episodes of the same task category, the student policy does not require an explicit goal observation. We compute the pairwise cosine similarity matrix  $\mathbf{S}$  for the shared groups  $\mathcal{K}_{\text{shared}} = \{\text{act}, \text{anc}, \text{obs}\}$ :

$$S_{i,j} = \frac{Z^i \cdot Z^j}{\|Z^i\|_2 \|Z^j\|_2}, \quad i, j \in \mathcal{K}_{\text{shared}}, \quad (12)$$

By minimizing  $\mathcal{L}_{\text{rel}} = \|\mathbf{S}_t^{\text{tea}} - \hat{\mathbf{S}}_t^{\text{stu}}\|_F^2$ , we force the student to mimic the teacher’s perception of relative spatial configurations. Crucially, although the student lacks the explicit  $Z_t^{\text{ref}}$ , it learns to satisfy the task objective by inheriting the teacher’s refined understanding of the active-anchor-obstacle relationship, which implicitly encodes the target-seeking behavior demonstrated by the teacher. We warm-start the student by initializing the MLP policy network with the teacher’s learned weights, so the student trains from an informed prior rather than from scratch, substantially accelerating convergence.

During distillation, beyond the physical-parameter randomization in Phase 1, we further apply domain randomization to camera-pose-related observations to improve the student’s robustness during real-world deployment.

## IV. EXPERIMENTS

We evaluate EgoPush in both simulation and real-world and demonstrate that: (1) Our method achieves precise multi-object rearrangement for items with diverse geometries into various target formations in both simulation and real world, as shown in Fig. 1; (2) Restricting Observation Space for RL Teacher ensures that the student receives a more accessible and distillable supervision signal, leading to superior performance; (3) Decomposing long-horizon tasks into sequential sub-tasks and employing time-decayed task completion rewards at the stage level significantly accelerates RL convergence; (4) Incorporating an auxiliary Relational Distillation loss enables the student to effectively inherit the teacher’s spatial reasoning.

### A. Experimental Setup

The RL policy is trained in NVIDIA Isaac Lab. We utilize an AMD EPYC Turin 9355 CPU and an NVIDIA RTX A6000

TABLE I: Ablation Studies of RL teacher observations. **SR** denotes success rate, **ExecTime** denotes execution time per episode, and **TrajLen** denotes trajectory length. The **ExecTime** and **TrajLen** are only computed for successful episodes.

	Teacher			Student		
	SR / % $\uparrow$	ExecTime $\downarrow$	TrajLen $\downarrow$	SR / % $\uparrow$	ExecTime $\downarrow$	TrajLen $\downarrow$
Ours	<b>99.31</b>	395.59	12.11	<b>70.70</b>	460.60	12.60
w/o C-GV	98.34	394.39	12.01	21.09	<b>453.33</b>	<b>9.85</b>
w/o FOV (global)	99.22	<b>373.20</b>	<b>11.06</b>	0	–	–

Ada GPU for both RL training and policy distillation. During the RL phase, we deploy 8,192 parallel environments for training, while the distillation phase employs 512 parallel environments for online supervised learning. In simulation, the manipulated objects consist of three primitive shapes: a cube ( $15 \times 15 \times 15$  cm), a cylinder (diameter 15 cm, height 15 cm), and a triangular prism (equilateral base with 15 cm side length, height 15 cm). The robot’s collision body is modeled as a  $14 \times 14 \times 14.3$  cm cube. To account for the minimum depth sensing range of approximately 15 cm of the real-world camera, we equipped the robot with a pusher of 7.5 cm length. We utilize an RGB-D camera with a  $69^\circ$  horizontal Field of View (FOV) and a resolution of  $240 \times 180$  pixels.

From our designed tasks (Fig. 1), we initially utilize a relatively straightforward cross-shaped formation task involving cubes. The ablated components already exhibit a significant performance gap compared to our full method. Consequently, we adopt this task as the standard evaluation benchmark for the first two distillation experiments. During testing, the teacher model performs across 1,024 episodes, while the student model is tested across 256 episodes. We further evaluate our model on objects with different geometries in appendix E.1. However, for the Relational Distillation ablation, the ablated version achieves performance comparable to our model on the cross-shape task. To further differentiate their capabilities, we introduce a more challenging line-shape formation task. Finally, for baseline comparisons, we simplify the environment to a two-object setup where the goal is to push one object towards another. Even in this simplified scenario, the baselines perform poorly, failing to yield satisfactory results.

### B. Ablation Studies

**Restricted Observations for RL Teacher.** We constrain the RL teacher’s observations to mitigate the observability mismatch between the privileged teacher and the partially-observed student. This approach incorporates two key designs: virtual egocentric FOV masking and center-gated visibility for privileged reference keypoints. To evaluate the efficacy of these designs, we consider two relaxed teacher variants:

- (i) **w/o FOV Masking (global)**, which removes the virtual egocentric FOV constraint and exposes global keypoints; and
- (ii) **w/o Center-Gated Visibility (w/o C-GV)**, which retains the FOV but removes the center-gated reference visibility.

All teacher variants are trained with the same reinforcement learning algorithm, reward functions, and hyperparameters, and are distilled into students with the same visual architecture. Furthermore, all policies are trained for a sufficiently

extensive duration to ensure full convergence.

Table I reports teacher and student performance together with metrics for the above variants. Analysis of the results in the table reveals that the Teacher Success Rate (SR) remains consistently high across all three settings ( $> 98\%$ ), validating the soundness of our reinforcement learning framework. In terms of execution time and trajectory length, the Global setting significantly outperforms the other two, while the w/o C-GV variant is slightly better than our policy. This trend is expected: as the teacher’s observations become more constrained, its performance degrades accordingly.

However, the student results tell a drastically different story. The w/o C-GV variant attains only a 21% success rate, whereas our student reaches 70%. Even though the Global teacher is near-perfect, its student fails to achieve a single successful episode. For execution time and trajectory length (computed over successful episodes only), w/o C-GV appears shorter than our student. This is mainly due to a success-selection effect: w/o C-GV succeeds only on trivial instances (*e.g.*, the active object starts close to and already facing the anchor), which naturally yield short trajectories. Our student succeeds on harder instances that require longer navigation and pushing, thereby increasing the mean ExecTime and trajectory length among successful runs.

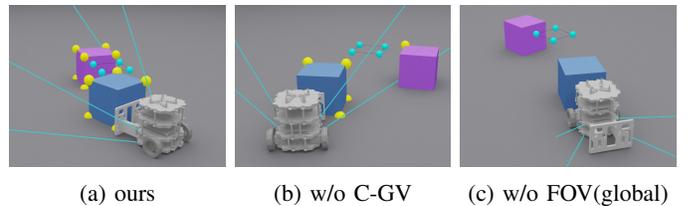


Fig. 3: Comparative analysis of teacher observation spaces.

To provide an intuitive explanation for the observed teacher–student capability misalignment, Fig. 3 visualizes the teacher behaviors under the three observation settings. Under our setting, the teacher predominantly pushes while facing the anchor to keep it within the gated FOV, thereby maintaining visibility of the C-GV–constrained reference point cloud (Fig. 3a); In contrast, the w/o C-GV teacher often sees only a partial set of the anchor keypoints and effectively follows a ‘lazy’ strategy that tracks the reference point cloud without explicitly controlling anchor visibility; consequently, the student receives sparse actionable cues and fails to recover the teacher’s performance (Fig. 3b); Under Global observation, the teacher is unconstrained and frequently pushes while

TABLE II: Credit assignment ablation on Cross Rearrangement Task. Starting from an episode-terminal reward baseline, we progressively add stage-wise rewards (SWR), temporal decay using a global episode timer (TD), and a stage timer (ST) that resets the decay schedule at stage boundaries.

Method	SR / % $\uparrow$	ExecTime $\downarrow$	TrajLen $\downarrow$
Base	16.02	1076.42	13.78
SWR	87.50	505.13	14.24
SWR + TD	97.95	455.46	13.77
SWR + TD + ST (Ours)	<b>98.63</b>	<b>443.26</b>	<b>12.83</b>

facing away from the anchor, producing behaviors that are fundamentally not learnable from the student’s egocentric observations (Fig. 3c).

**Credit Assignment Ablation.** To validate the learning difficulty of the long-horizon sequential rearrangement task and to assess the effectiveness of our credit-assignment design, we ablate the reward shaping components used for credit assignment in a progressive manner. Starting from a sparse baseline (**Base**) that provides only episode-terminal success feedback, we introduce stage-wise rewards computed per stage (**SWR**) to encourage sub-goal attainment, using the reaching and placing terms in Eq. 1 and Eq. 3 without the temporal decay factor in Eq. 2. We then add temporal decay based on a global episode timer (**SWR+TD**) by applying Eq. 2 with episode-level timing. Finally, **Ours** uses a stage timer that resets the decay schedule at each stage boundary, so that each stage receives comparable temporal pressure.

The training curves (Fig. 4) show that **Ours** converges faster and more stably, suggesting that stage-aligned temporal shaping provides a more consistent learning signal across sequential stages. **This comparison uses an unmatched training budget: Ours** is trained only until it has largely converged (about  $\sim 45k$  steps), whereas to better isolate the effect of individual ablated components, we continue training the other variants for an additional steps, resulting in a total of  $\sim 90k$  steps. We report results using the best checkpoint achieved by each variant under its respective training budget.

Quantitative results are reported in Table II. *Base* learns unreliably in this long-horizon setting, indicating that sparse terminal feedback is inadequate for temporal credit assignment. Adding stage-wise rewards (*SWR*) substantially improves learnability, highlighting the benefit of stage decomposition. Introducing temporal decay with an episode-level timer (*SWR+TD*) further boosts performance, especially success rate, by encouraging earlier, decision-critical actions. Finally, **Ours** reaches near-saturated performance by resetting the decay schedule at each stage boundary with a stage timer—**despite using only half the training steps**, underscoring its more effective long-horizon credit assignment.

**Relational Distillation Ablation.** To evaluate the effect of relational supervision, we ablate the additional supervision in Eq. 12 and perform distillation with action loss only (*w/o Relational Distillation*). In CROSS, the student remains

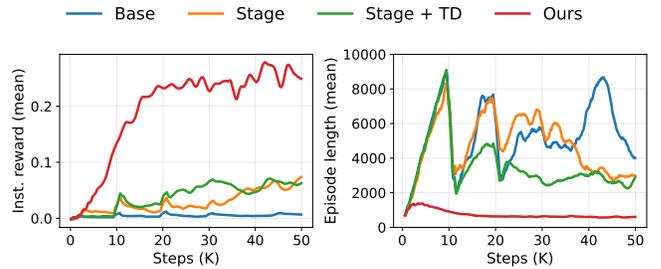


Fig. 4: Training curves for the credit assignment ablations.

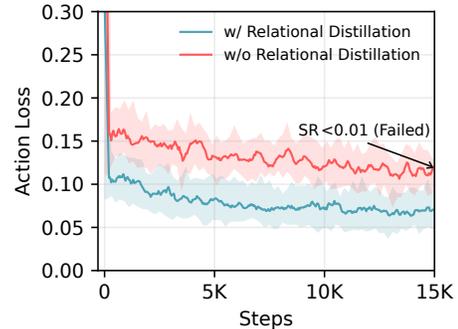


Fig. 5: Distillation training curve for distillation ablation.

competitive with **Ours**, likely because the anchor is fixed and the per-stage spatial relations are close to symmetric. However, LINE-SHAPE is more sensitive: small errors compound over the sequential execution, and the target configuration is not symmetric. Under this setting, *w/o Relational Distillation* yields a noticeably larger converged action loss and completely fails in final performance (Fig. 5).

### C. Baseline Comparisons

To demonstrate the difficulty of our task setting and the effectiveness of EgoPush, we compare against both a classical mapping-based baseline and a suite of end-to-end (E2E) deep reinforcement learning baselines with different observation modalities. Specifically, we include a traditional approach based on spatial intention maps [42], as well as E2E visual RL variants that use RGB, RGB-D, oracle semantic masks (RGBD-Sem), and recurrent policies (RGBD-RNN).

We include Spatial Intention Maps (SIM) [42] as a representative classical mapping-and-planning baseline. To align with our egocentric setup, we construct a 2D top-down semantic map from egocentric ground truth (*GT*) RGB-D observations with *GT* segmentation for semantics (to avoid confounding SIM with perception errors). For fairness, we do not provide robot ground-truth pose; instead, SIM uses an odometry-style pose obtained by integrating the *executed* local velocities, which may drift over long-horizon, contact-rich episodes and degrade map consistency. SIM outputs a local spatial target (pixel/waypoint), which we convert to the same local velocity action interface via a waypoint-tracking controller with identical action bounds. For RGBD-Sem, we assume perfect semantic segmentation and provide the ground-truth mask together with depth as input. All learning-based methods

TABLE III: **Baseline comparisons on a simplified task.** We report reached the active object rate (**Reach**), Success Rate (**SR**; counted as place the active object near anchor), and Trajectory Length (**Traj**) for successful episodes.

Method	Input	Reach / % $\uparrow$	SR / % $\uparrow$	Traj $\downarrow$
<i>Classical baseline</i>				
SIM [42]	RGB-D + map + odom	30.94	19.26	9.65
<i>End-to-End Visual RL Baselines</i>				
E2E-RGB (CNN)	RGB	5.57	0.00	–
E2E-RGBD (CNN)	RGB-D	6.54	0.29	1.30
RGBD-Sem	RGB-D + GT Seg	39.12	0.10	3.26
E2E-Curriculum	RGB-D	32.75	0.78	19.66
RGBD-RNN	RGB-D	15.43	0.10	3.03
<b>EgoPush (ours)</b>	<b>RGB-D</b>	<b>100.00</b>	<b>100.00</b>	4.66

are trained from scratch using the same reward definition, termination conditions, and training budget.

Table III summarizes the comparative results. Across all E2E variants, learning from egocentric observations remains brittle: although several baselines achieve non-trivial **Reach**, their success rates remains below 1%, indicating a failure to complete the long-horizon push-and-align objective. Notably, even with GT semantic masks, RGBD-Sem still fails to solve the task, suggesting that perfect object-level perception alone does not resolve the underlying difficulty. These results point to long-horizon geometric reasoning and maintaining spatial consistency under partial observability as the dominant bottlenecks, especially when contact-rich interactions perturb object configurations and key objects may temporarily leave the FOV. SIM also performs poorly: without GT pose, action-integrated odometry drift accumulates over long horizons and undermines mapping and planning consistency, leading to compounding errors in dynamic, contact-rich scenes.

In contrast, EgoPush achieves a success rate of 100% with 100% reach rate, demonstrating robust long-horizon capability under the same egocentric sensing constraints. These results support our central claim that visually grounded distillation and structured supervision are crucial for solving long-horizon, contact-rich rearrangement from egocentric vision. We provide qualitative analyses of representative success and failure cases in the appendix A.

#### D. Real World Experiments

Real world experiments are conducted in a  $3\text{m} \times 3\text{m}$  arena (shown in the left of Fig. 1), containing a single mobile robot (TurtleBot3 Burger) equipped with an Intel RealSense D435i camera and multiple boxes of distinct colors. The arena floor is gray and surrounded by foam walls of the same color to facilitate segmentation of the boxes. The boxes are  $15\text{cm} \times 15\text{cm} \times 15\text{cm}$  cubes covered with colored paper. At the beginning of each trial, the robot and all boxes are placed with random initial positions and orientations inside the arena. Fig. 6 visualizes the experiment setup and robot configuration. The onboard Jetson Nano transmits captured RGB-D frames to a remote RTX 5080 server via WebSocket, which performs real-time policy computation and returns control signals to the robot’s low-level controller.

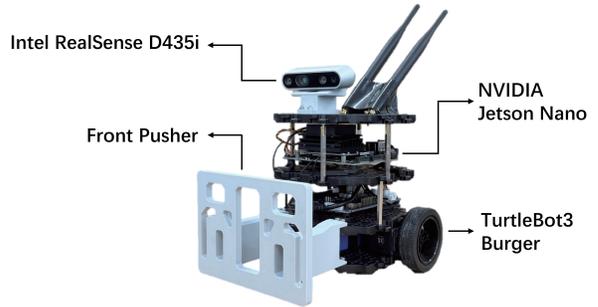


Fig. 6: Real-world hardware setup.

We zero-shot deployed the best student policy into this real-world setup and tested 10 episodes of the cross-shape task with 5 boxes. The real robot was able to finish pushing all four boxes to the anchor box within 2 minutes, achieving an 80 % success rate under a loose evaluation metric (allowing for minor orientation and centering deviations). In general, we observed that the policy behavior was qualitatively similar to that in simulation. However, the policy’s action scale was restricted to accommodate the servo torque limits of the real robot, resulting in a slight degradation of task robustness and efficiency in training in simulation.

## V. DISCUSSIONS

In this work, we presented **EgoPush**, a framework that enables mobile robots to solve long-horizon multi-object rearrangement tasks using solely egocentric vision. While our constrained teacher and structured supervision substantially improve distillability and robustness, several limitations remain and motivate future work. Our student is primarily reactive: it conditions on the current egocentric observation (group-wise depth) and a short action history, but does not maintain an explicit belief over objects that are temporarily unobserved. In scenes with *consecutive* obstacles, we observe a characteristic dilemma: facing the goal (anchor/target cues) can occlude the feasible corridor, while turning to inspect a feasible corridor may lose goal cues altogether. Without memory, the policy can enter an oscillatory or deadlock mode—alternating between “goal-seeking” and “path-seeking” views without committing to a globally consistent maneuver. This failure mode is especially pronounced when the robot must thread through narrow passages while continuously coordinating contact with the active object.

Importantly, our object-centric latent interface suggests a promising direction to mitigate this limitation: leveraging the latent space as a compact spatial state and integrating it with recurrent sequence models (*e.g.*, GRU/LSTM) to learn *spatial memory*. Since the latent features are structured around task roles and encode relative relations rather than absolute poses, they provide a natural substrate for temporal aggregation across occlusions, enabling the agent to maintain a persistent belief over traversable corridors and goal cues even when either becomes temporarily out of view.

## ACKNOWLEDGMENTS

We would like to thank Eugene Vinitsky, Tianyue Wu, and Tianyu Zhao for valuable discussions. We also thank Zeyu Jiang for assistance with the real-world experiments, Zhejun Cui and Zhichao Han for valuable feedback on early drafts of the manuscript, and Bowen Hu and Weijie Kong for help with figure preparation. This work was supported by NSF Grant 2238968, and by NYU IT High Performance Computing resources, services, and staff expertise.

## REFERENCES

- [1] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged Locomotion in Challenging Terrains using Egocentric Vision. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 403–415. PMLR, 14–18 Dec 2023.
- [2] Jeeho Ahn and Christoforos Mavrogiannis. ReloPush: Multi-Object Rearrangement in Confined Spaces with a Nonholonomic Mobile Robot Pusher. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.
- [3] Tianxu An, Joonho Lee, Marko Bjelonic, Flavio De Vincenti, and Marco Hutter. Scalable Multi-Robot Cooperation for Multi-Goal Tasks Using Reinforcement Learning. *IEEE Robotics and Automation Letters*, 10(2):1585–1592, 2025. doi: 10.1109/LRA.2024.3521183.
- [4] M. Bertalmio, A.L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001. doi: 10.1109/CVPR.2001.990497.
- [5] Hien Bui\*, Yufeyang Gao\*, Haoran Yang\*, Eric Cui, Siddhant Mody, Brian Acosta, Thomas Stephen Felix, Bibit Bianchini, and Michael Posa. Push Anything: Single- and Multi-Object Pushing From First Sight with Contact-Implicit MPC. *arXiv preprint arXiv:2510.19974*, 2025.
- [6] 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. doi: 10.1126/scirobotics.adc9244.
- [7] Yiyu Chen and Quan Nguyen. Learning Agile Locomotion and Adaptive Behaviors via RL-augmented MPC. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11436–11442, 2024. doi: 10.1109/ICRA57147.2024.10610453.
- [8] Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive Whole-Body Control for Humanoid Robots. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.107.
- [9] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme Parkour with Legged Robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11443–11450, 2024. doi: 10.1109/ICRA57147.2024.10610200.
- [10] Lin Cong, Hongzhuo Liang, Philipp Ruppel, Yunlei Shi, Michael Görner, Norman Hendrich, and Jianwei Zhang. Reinforcement learning with vision-proprioception model for robot planar pushing. *Frontiers in Neurobotics*, 16:829437, 2022.
- [11] Ioannis Dadiotis, Mayank Mittal, Nikos Tsagarakis, and Marco Hutter. Dynamic Object Goal Pushing with Mobile Manipulators Through Model-Free Constrained Reinforcement Learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13363–13369, 2025. doi: 10.1109/ICRA55743.2025.11128166.
- [12] Cunxi Dai, Xiaohan Liu, Koushil Sreenath, Zhongyu Li, and Ralph Hollis. Interactive navigation with adaptive non-prehensile mobile manipulation. *arXiv preprint arXiv:2410.13418*, 2024.
- [13] Nils Dengler, David Großklaus, and Maren Bennewitz. Learning Goal-Oriented Non-Prehensile Pushing in Cluttered Scenes. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1116–1122, 2022. doi: 10.1109/IROS47612.2022.9981873.
- [14] Ismail Geles, Leonard Bauersfeld, Angel Romero, Jiaxu Xing, and Davide Scaramuzza. Demonstrating Agile Flight from Pixels without State Estimation. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.082.
- [15] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning Latent Dynamics for Planning from Pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2555–2565. PMLR, 09–15 Jun 2019.
- [16] Dan Haramati, Tal Daniel, and Aviv Tamar. Entity-Centric Reinforcement Learning for Object Manipulation from Pixels. In *International Conference on Learning Representations (ICLR)*, 2024.
- [17] Johannes Heeg, Yunlong Song, and Davide Scaramuzza. Learning Quadrotor Control from Visual Features Using Differentiable Simulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4033–4039, 2025. doi: 10.1109/ICRA55743.2025.11128641.
- [18] Adam Heins and Angela P. Schoellig. Force Push: Robust Single-Point Pushing With Force Feedback. *IEEE Robotics and Automation Letters*, 9(8):6856–6863, 2024. doi: 10.1109/LRA.2024.3414180.
- [19] Yongpeng Jiang, Yongyi Jia, and Xiang Li. Contact-Aware Non-Prehensile Manipulation for Object Retrieval

- in Cluttered Environments. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10604–10611, 2023. doi: 10.1109/IROS55552.2023.10341476.
- [20] Gaoyuan Liu, Joris de Winter, Denis Steckelmacher, Roshan Kumar Hota, Ann Nowe, and Bram Vanderborght. Synergistic Task and Motion Planning With Reinforcement Learning-Based Non-Prehensile Actions. *IEEE Robotics and Automation Letters*, 8(5):2764–2771, 2023. doi: 10.1109/LRA.2023.3261708.
- [21] Haokun Liu, Zhaoqi Ma, Yunong Li, Junichiro Sugihara, Yicheng Chen, Jinjie Li, and Moju Zhao. Hierarchical Language Models for Semantic Navigation and Manipulation in an Aerial-Ground Robotic System. *Advanced Intelligent Systems*, n/a(n/a):e202500640, 2025. doi: <https://doi.org/10.1002/aisy.202500640>.
- [22] Minghuan Liu, Zhengbang Zhu, Xiaoshen Han, Peng Hu, Haotong Lin, Xinyao Li, Jingxiao Chen, Jiafeng Xu, Yichu Yang, Yunfeng Lin, Xinghang Li, Yong Yu, Weinan Zhang, Tao Kong, and Bingyi Kang. Manipulation as in Simulation: Enabling Accurate Geometry Perception in Robots. *arXiv preprint arXiv:2406.10759*, 2025.
- [23] John Lloyd and Nathan F. Lepora. Goal-Driven Robotic Pushing Using Tactile and Proprioceptive Feedback. *IEEE Transactions on Robotics*, 38(2):1201–1212, 2022. doi: 10.1109/TRO.2021.3104471.
- [24] Jiangran Lyu, Ziming Li, Xuesong Shi, Chaoyi Xu, Yizhou Wang, and He Wang. Dywa: Dynamics-adaptive world action model for generalizable non-prehensile manipulation. In *International Conference on Computer Vision (ICCV)*, 2025.
- [25] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharmashan Kumar, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. 518(7540):529–533. ISSN 1476-4687. doi: 10.1038/nature14236.
- [26] João Moura, Theodoros Stouraitis, and Sethu Vijayakumar. Non-prehensile Planar Manipulation via Trajectory Optimization with Complementarity Constraints. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 970–976, 2022. doi: 10.1109/ICRA46639.2022.9811942.
- [27] Idil Ozdamar, Doganay Sirintuna, Robin Arbaud, and Arash Ajoudani. Pushing in the Dark: A Reactive Pushing Strategy for Mobile Robots Using Tactile Feedback. *IEEE Robotics and Automation Letters*, 9(8):6824–6831, 2024. doi: 10.1109/LRA.2024.3414279.
- [28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.
- [29] 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, 2024.
- [30] Kejia Ren, Gaotian Wang, Andrew S. Morgan, Lydia E. Kavradi, and Kaiyu Hang. Object-Centric Kinodynamic Planning for Nonprehensile Robot Rearrangement Manipulation. *IEEE Transactions on Robotics*, 41:5761–5780, 2025. doi: 10.1109/TRO.2025.3613532.
- [31] Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.
- [32] Fereshteh Sadeghi and Sergey Levine. CAD2RL: Real Single-Image Flight Without a Single Real Image. In *Proceedings of Robotics: Science and Systems*, Cambridge, Massachusetts, July 2017. doi: 10.15607/RSS.2017.XIII.034.
- [33] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 2017.
- [34] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025.
- [35] Haoran Song, Joshua A. Haustein, Weihao Yuan, Kaiyu Hang, Michael Yu Wang, Danica Kragic, and Johannes A. Stork. Multi-Object Rearrangement with Monte Carlo Tree Search: A Case Study on Planar Nonprehensile Sorting. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9433–9440, 2020. doi: 10.1109/IROS45743.2020.9341532.
- [36] Zili Tang, Yuming Feng, and Meng Guo. Collaborative Planar Pushing of Polytopic Objects with Multiple Robots in Complex Scenes. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.063.
- [37] Zili Tang, Ying Zhang, and Meng Guo. PushingBots: Collaborative Pushing via Neural Accelerated Combinatorial Hybrid Optimization. *IEEE Transactions on Robotics*, 42:579–599, 2026. doi: 10.1109/TRO.2025.3647767.

- [38] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World, 2017.
- [39] Shagun Uppal, Ananye Agarwal, Haoyu Xiong, Kenneth Shaw, and Deepak Pathak. Spin: Simultaneous perception interaction and navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18133–18142, 2024.
- [40] Jun Wang, Ying Yuan, Haichuan Che, Haozhi Qi, Yi Ma, Jitendra Malik, and Xiaolong Wang. Lessons from Learning to Spin “Pens”. In *Conference on Robot Learning (CoRL)*, 2024.
- [41] Jimmy Wu, Xingyuan Sun, Andy Zeng, Shuran Song, Johnny Lee, Szymon Rusinkiewicz, and Thomas Funkhouser. Spatial Action Maps for Mobile Manipulation. In *Proceedings of Robotics: Science and Systems*, Corvallis, Oregon, USA, July 2020. doi: 10.15607/RSS.2020.XVI.035.
- [42] Jimmy Wu, Xingyuan Sun, Andy Zeng, Shuran Song, Szymon Rusinkiewicz, and Thomas Funkhouser. Spatial Intention Maps for Multi-Agent Mobile Manipulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, page 8749–8756. IEEE, May 2021. doi: 10.1109/icra48506.2021.9561359.
- [43] Tianyue Wu, Yeke Chen, Tianyang Chen, Guangyu Zhao, and Fei Gao. Whole-Body Control Through Narrow Gaps from Pixels to Action. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11317–11324, 2025. doi: 10.1109/ICRA55743.2025.11128088.
- [44] Guangtong Xu, Tianyue Wu, Zihan Wang, Qianhao Wang, and Fei Gao. Flying on Point Clouds with Reinforcement Learning. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7231–7238, 2025. doi: 10.1109/IROS60139.2025.11246821.
- [45] Taegeun Yang, Jiwoo Hwang, Jeil Jeong, Minsung Yoon, and Sung-Eui Yoon. Efficient Navigation Among Movable Obstacles using a Mobile Manipulator via Hierarchical Policy Learning. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 17130–17137, 2025. doi: 10.1109/IROS60139.2025.11245985.
- [46] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving Sample Efficiency in Model-Free Reinforcement Learning from Images, 2020.
- [47] Shaofeng Yin, Yanjie Ze, Hong-Xing Yu, C. Karen Liu, and Jiajun Wu. VisualMimic: Visual Humanoid Locomanipulation via Motion Tracking and Generation. *arXiv preprint arXiv:2509.20322*, 2025.
- [48] Haoxiang You, Yilang Liu, and Ian Abraham. Accelerating Visual-Policy Learning through Parallel Differentiable Simulation. *arXiv preprint arXiv:2505.10646*, 2025.
- [49] Jiarui Zhang, Chengyong Lei, Chengjiang Dai, Lijie Wang, Zhichao Han, and Fei Gao. High-Speed Vision-Based Flight in Clutter with Safety-Shielded Reinforcement Learning. *arXiv preprint arXiv:2602.08653*, 2026.
- [50] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher Atkeson, Sören Schwertfeger, Chelsea Finn, and Hang Zhao. Robot Parkour Learning. In *Conference on Robot Learning (CoRL)*, 2023.

## APPENDIX

### Contents

- A. Qualitative Analysis of Baseline Policies ..... 12
- B. Training Details ..... 14
- C. Real World Robot System Setup ..... 17
- D. Image Processing ..... 17
- E. Additional Experiment Result ..... 18

#### A. QUALITATIVE ANALYSIS OF BASELINE POLICIES

This section documents qualitative failure modes observed in our baseline comparisons. We first summarize the **shared training protocol and timing** used for all baselines, and then analyze representative failure cases.

##### A.1 Shared Training Protocol and Timing (Fairness)

All learning-based baselines reported in Section IV-C (Baseline Comparisons), Table III of the main manuscript are trained from scratch under identical simulator timing, action bounds, and interaction budget. Concretely, the physics time step is fixed at 1/60 s and actions are applied with a decimation of 6, so each policy step corresponds to 0.1 s of simulated time (10 Hz control). Unless explicitly stated, we keep the same termination conditions and total PPO training budget across E2E variants; the primary differences are the observation modality (RGB, RGB-D, or oracle semantics) and the policy backbone (feed-forward or recurrent). For the SIM baseline, we follow the original observation-space design [42], using a map-based observation, a spatial waypoint action, and a SIM-style reward composed of progress, success, and collision-penalty terms. Table IV lists the full reproducibility settings.

##### A.2 End-to-End Visual RL Baselines

Before analyzing qualitative failure cases, we briefly summarize the configuration differences among E2E baselines. All E2E variants are **visuomotor policies** that take egocentric observations and output **diff-drive** commands  $(v, \omega)$ , without teacher supervision or explicit mapping. Concretely, **E2E-RGB** uses only ego RGB images; **E2E-RGBD** additionally provides depth; **RGBD-Sem** augments RGB-D with *oracle* semantic grouping via semantic patches and previous action input, to test whether explicit semantic segmentation alone is sufficient to solve the task; and **RGBD-RNN** adds a GRU head to test whether short-term memory alleviates partial observability. We further include an **E2E-Curriculum** variant trained in two stages: the first stage uses only reaching-related rewards (orientation, distance reduction, and touch), while goal-completion and safety terms are disabled; the second stage restores the full reward set for pushing and resumes from the first-stage checkpoint. The two stages use equal training budgets. We discuss the **SIM** map-based baseline separately in Sec. A.3. Across E2E variants (RGB, RGB-D, RGBD-Sem, and recurrent policies), the most salient limitation is **short-horizon reactivity under partial observability**. Even when the agent transiently achieves the *Reach* event (touching the active box at least once), it often fails to maintain a stable

interaction long enough to reliably transport the active box to the anchor region. We summarize several recurring failure patterns below.

**(A) Control collapse: spinning in place or freezing.** The most common degenerate outcome is that the robot either spins in place for the majority of the episode or remains nearly stationary after a brief initial motion. This behavior is consistent with the difficulty of long-horizon credit assignment from egocentric pixels: early in training, stochastic policies can occasionally obtain partial rewards by chance (e.g., briefly facing or touching the box), yet the resulting gradients may drive the policy toward a locally stable but task-irrelevant attractor.

**(B) Early stopping before meaningful progress.** We frequently observe policies that stop prematurely, either before making contact with the active box, or after pushing the box partway toward the goal but before satisfying the success condition. Qualitatively, this resembles learning a short-horizon sub-skill (e.g., approach or initiate pushing) without learning the full conditional logic required for completing the task (e.g., verifying box-at-goal and continuing to correct residual errors).

**(C) Goal chasing without contact (“pushing air”).** Another common pattern is that the policy learns to orient toward the goal region (or the anchor) but fails to maintain contact with the active box. Once contact is lost, the robot continues to drive toward the goal as if the box were attached, effectively pushing empty space.

**(D) Premature pushing without reaching.** Finally, some policies start executing push-like motions toward the goal without first reaching the active box, indicating that the policy has overfit to goal-directed cues while missing the prerequisite contact dynamics.

Even with **oracle semantic masks** (RGBD-Sem), performance remains poor. This indicates that *recognizing* objects is not sufficient; what is missing is **persistent spatial consistency**—the ability to remember where objects were, re-acquire them after occlusion, and coordinate navigation and pushing while the configuration changes. Representative E2E failure cases are shown in Figure 7 (first three rows).

##### A.3 Spatial Intention Maps (SIM) Baseline

SIM represents a classical mapping-and-planning baseline built around a **map-centric** state representation and an explicit **spatial action** (selecting a target location on a map), rather than directly regressing low-level robot velocities from pixels. To adapt SIM to our benchmark, we retain its core ingredient—a robot-centric, heading-aligned **spatial intention map**—as the policy input, and integrate it into our PPO pipeline. Concretely, we construct a local semantic map from oracle segmentation and let the policy output a waypoint  $(x, y)$  in the local map frame; a geometric controller then converts this waypoint into diff-drive commands  $(v, \omega)$  under the same control rate and bounds as other baselines (Table IV). For training, we use a **SIM-style reward** consisting of a progress term (delta distance decrease), a sparse one-shot

TABLE IV: **Reproducibility details for baseline comparisons for Section IV-C (Baseline Comparisons), Table III in the main manuscript.** All baselines share the same environment timing, action bounds, and PPO hyperparameters. E2E baselines use the same reward set, with an optional two-stage curriculum schedule, while the SIM baseline uses a SIM-style reward and waypoint macro-actions.

Category	Setting	Value	Applies to	Notes	
Env / timing	Physics step	$dt = 1/60$ s	all	Policy step = $dt \times$ decimation	
	Action decimation	6 (10 Hz)	all		
	Episode limit	100 s (1000 policy steps)	all	finite-horizon	
	Arena bound	radius 1.5 m	all	for out-of-bounds checks	
	Parallel envs (train)	512	all baselines	Isaac Lab vectorized envs	
Sensors	Ego camera FOV	$70^\circ \times 60^\circ$ (H $\times$ V)	all vision baselines	Camera updates once per policy step	
	Ego camera rate	10 Hz	all vision baselines		
	Camera resolution	$210 \times 180$ (raw), $\text{downsample} \times 2 \rightarrow 104 \times 90$	all vision baselines		
	SIM map size	$96 \times 96$	SIM baseline	robot-centric local map	
Actions	Action space	$\mathbf{a} \in [-1, 1]^2$	all	PPO Gaussian mean is $\tanh$ -squashed	
	Low-level command	$(v, \omega)$ for diff-drive	E2E baselines	$v = 0.4a_0, \omega = 2.84a_1$	
	SIM macro-action	waypoint $(x, y)$ in robot-local map frame	SIM baseline	executed by geometric controller producing $(v, \omega)$	
	Controller gains	$k_v = 1.0, k_\omega = 2.0$	SIM baseline	yaw gating ( $20^\circ$ ) and goal tolerance (5cm)	
Reward	Angle-to-box	Same as Ours	E2E baselines	distance-gated (0.5 m)	
	Progress shaping	Same as Ours	E2E baselines	phase-gated: reach phase (robot-to-box), place phase (box-to-anchor)	
	Reach completion	Same as Ours	E2E baselines	reach threshold 0.2 m; scaled by $\eta_t$ (Eq. 2 in the main manuscript)	
	Box-at-Goal completion reward	Same as Ours	E2E baselines	Curriculum stage-1 sets to 0	
	Slowdown near target	Same as Ours	E2E baselines	Curriculum stage-1 sets to 0	
	Out-of-bounds penalty	Same as Ours	E2E baselines	Curriculum stage-1 sets to 0	
	Exploration bonus	Same as Ours	E2E baselines	Curriculum stage-1 sets to 0	
	Smoothness penalty	Same as Ours	E2E baselines	Curriculum stage-1 sets to 0	
	Progress reward	$10.0 \times \Delta\text{distance}$ (m)	SIM baseline	dense progress term (robot to box / box to anchor)	
	Success reward	+10.0 per success event	SIM baseline	one-shot box-at-goal event	
	Out-of-bounds penalty	-0.25 per OOB step	SIM baseline	OOB proxy term in collision model	
Non-target disturbance	-1.0 per disturbance	SIM baseline	rising-edge non-target box motion penalty		
PPO (train)	Algorithm	PPO (skrl)	all baselines	Same optimizer/budget across baselines	
	Total steps	150k policy steps	all baselines		
	Rollout length	32	all baselines		vectorized rollout horizon
	Epochs / sbatches	8 / 32	all baselines		PPO update schedule
	Learning rate	$2.5 \times 10^{-4}$	all baselines		Adam
Eval	Episodes	1024 episodes	all methods	vectorized play-eval	
	Parallel envs	256	all methods		
	Metrics	Reach (%), SR (%), TrajLen	all methods		SR counts <i>Box-at-Goal</i> at least once

success term, and a lightweight collision penalty, matching the reward structure used in the original SIM work.

The main failure mode we observe is **state inconsistency caused by accumulated odometry and contact errors over time** (the real robot’s odometry drift is considerably worse than in simulation due to wheel slip, imperfect ground contact, and etc.). Since we do not provide GT robot pose, SIM updates its map using an odometry-style pose obtained by integrating *executed* local velocities. Over long episodes, small integration errors accumulate; under repeated contacts, the discrepancy between assumed and true pose can grow large enough that the semantic map becomes misaligned with reality. This undermines both (i) local planning (targets chosen on a distorted map) and (ii) the feedback loop between map updates

and action selection, yielding compounding errors.

In addition, the task requires **tight coupling between perception and interaction**. Even with a reasonably consistent map, pushing success depends on fine-grained alignment and maintaining stable contact, which is difficult to achieve with a purely map-based decision layer without structured, stage-aware supervision. In the supplementary videos, this often appears as repeated attempts to re-approach the active object or to push from unfavorable angles, leading to wasted steps and increased disturbances. Representative SIM failure cases are shown in Figure 7 (SIM-Odom row) and Figure 8, where GT-map mismatch increases from left to right.

TABLE V: **Per-method obs and policy specification for Section IV-C (Baseline Comparisons), Table III in the main manuscript.** All methods share the common protocol in Table IV. Here we list the *only* method-specific differences: obs tensor layout and policy backbone.

Method	Input	Policy obs (tensor)	Backbone / head	Action parameterization
SIM [42]	RGB-D ego obs local map odometry pose	Robot-centric semantic map shape: $4 \times 96 \times 96$	Conv(32, $5 \times 5$ , s2) Conv(64, $5 \times 5$ , s2) Conv(64, $3 \times 3$ , s1) MLP(256, 256) Gaussian head	Policy outputs local waypoint $(x, y)$ Geometric controller tracks waypoint Simulator receives diff-drive $(v, \omega)$
E2E-RGB	RGB ego obs	Ego RGB tensor shape: $3 \times 90 \times 104$ flattened size: 28,080	Conv(32, $5 \times 5$ , s2) Conv(64, $5 \times 5$ , s2) Conv(64, $3 \times 3$ , s1) MLP(256, 256) Gaussian head	Policy directly predicts $(v, \omega)$ both channels normalized to $[-1, 1]$
E2E-RGBD E2E-Curriculum	RGB-D ego obs	Ego RGB-D tensor shape: $4 \times 90 \times 104$ flattened size: 37,440	Conv(32, $5 \times 5$ , s2) Conv(64, $5 \times 5$ , s2) Conv(64, $3 \times 3$ , s1) MLP(256, 256) Gaussian head	Policy directly predicts $(v, \omega)$ both channels normalized to $[-1, 1]$
RGBD-Sem	RGB-D ego obs semantic group	3 sem RGB-D patches each patch: $4 \times 90 \times 104$ flattened size: 112,322	patch Conv(32, $5 \times 5$ , s2) patch Conv(64, $5 \times 5$ , s2) patch Conv(64, $3 \times 3$ , s1) MLP(256, 256) Gaussian head	Policy directly predicts $(v, \omega)$ both channels normalized to $[-1, 1]$
RGBD-RNN	RGB-D ego obs	Ego RGB-D tensor shape: $4 \times 90 \times 104$ flattened size: 37,440	Conv(32, $5 \times 5$ , s2) Conv(64, $5 \times 5$ , s2) Conv(64, $3 \times 3$ , s1) MLP(256, 256) GRU(256, seq 32) Gaussian head	Policy directly predicts $(v, \omega)$ both channels normalized to $[-1, 1]$

#### A.4 Qualitative Visualizations

To make the above failure modes concrete, we provide **filmstrip-style qualitative case studies** over full episodes. Figure 7 summarizes four representative baseline failure patterns (three E2E rows and one SIM-Odom row), while Figure 8 shows a paired GT-vs-map sequence in which alignment degrades over time.

### B. TRAINING DETAILS

#### B.1 Reward Function Details

(3) **Smoothness:** To smooth the output of the neural network, a penalty is implemented to discourage sudden changes by comparing the current action to the previous action:

$$r_{\text{smooth},t} = (\Delta v_t)^4 \mathbb{I}[\Delta v_t > \varepsilon_{\text{smooth}}] + (\Delta \omega_t)^4 \mathbb{I}[\Delta \omega_t > \varepsilon_{\text{smooth}}], \quad (13)$$

where  $v_t$  and  $\omega_t$  are the linear and angular action components;  $\Delta v_t = v_t - v_{t-1}$  and  $\Delta \omega_t = \omega_t - \omega_{t-1}$  denote consecutive action differences;  $\varepsilon_{\text{smooth}}$  is the (pre-scaling) threshold.

(4) **Slowdown Near Target:**

$$r_{\text{slow},t} = g_d(t) g_v(t), \quad (14)$$

$$g_d(t) = \left( 1 - \frac{\|\mathbf{p}_{\text{act},t} - \mathbf{p}_{\text{ref},t}\|_2}{\varepsilon_{d,\text{th}}} \right) \mathbb{I}[\|\mathbf{p}_{\text{act},t} - \mathbf{p}_{\text{ref},t}\|_2 < \varepsilon_{d,\text{th}}], \quad (15)$$

$$g_v(t) = \max \left( -1, 1 - \frac{\|\mathbf{v}_{\text{act},t}\|_2}{\varepsilon_{v,\text{th}}} \right) \quad (16)$$

where  $\varepsilon_{d,\text{th}}$  defines the near-target region (distance threshold),  $\varepsilon_{v,\text{th}}$  is the speed threshold, and  $\mathbf{v}_{\text{act},t}$  is the planar velocity vector of the active object.

#### B.2 World Box

Under the batch rendering architecture of Isaac Lab, all environments reside within a unified 3D spatial coordinate system. This proximity often leads to visual interference, as camera sensors may capture features from adjacent environments. To mitigate this visual crosstalk, we encapsulate each environment within a gray cylindrical world box (Fig. 9) to effectively isolate per-environment visual information.

#### B.3 FOV Design

We constrain the privileged teacher to use only information recoverable from an egocentric camera by (i) a virtual frustum

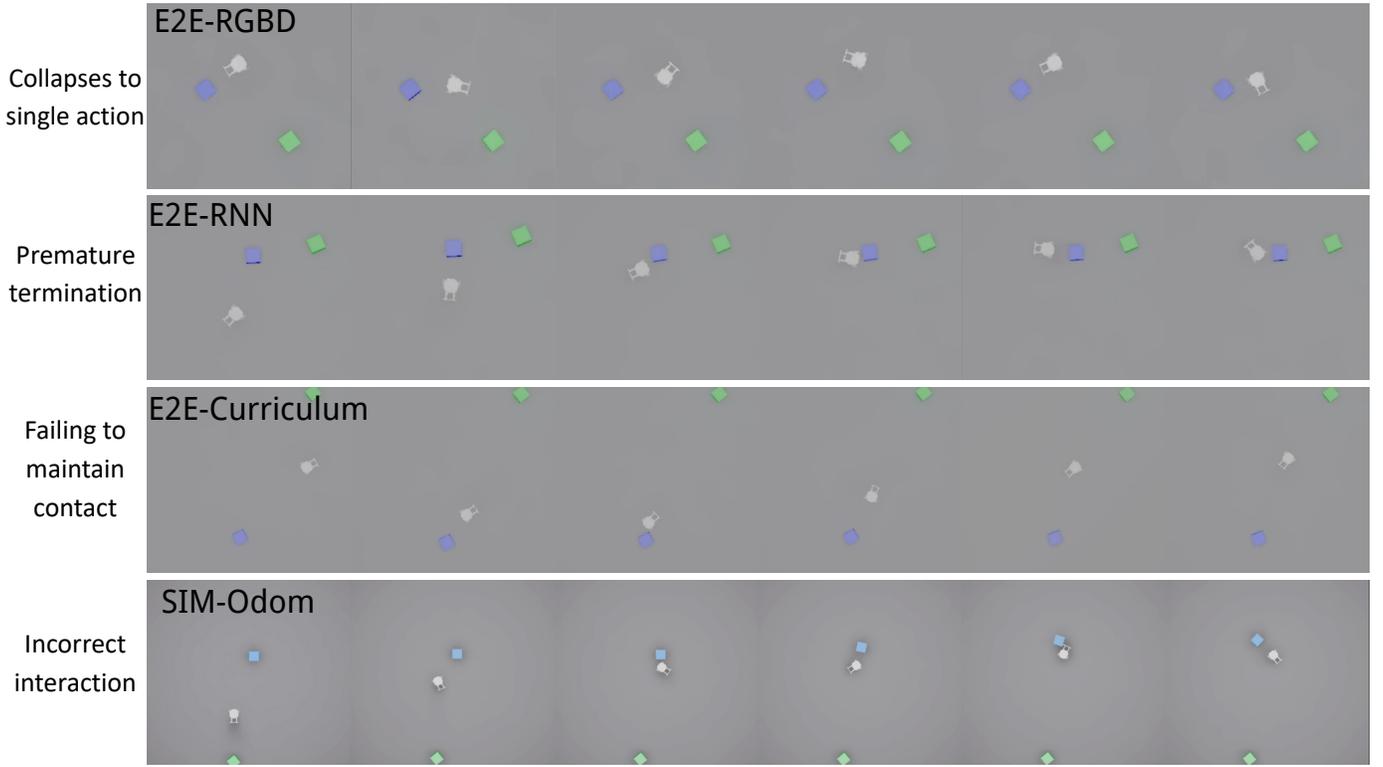


Fig. 7: **Representative baseline failure filmstrips.** From top to bottom: **E2E-RGBD** collapses to a near-single action mode, **E2E-RNN** exhibits premature termination, **E2E-Curriculum** fails to maintain stable contact after initial approach, and **SIM-Odom** shows incorrect interaction due to map/pose inconsistency.

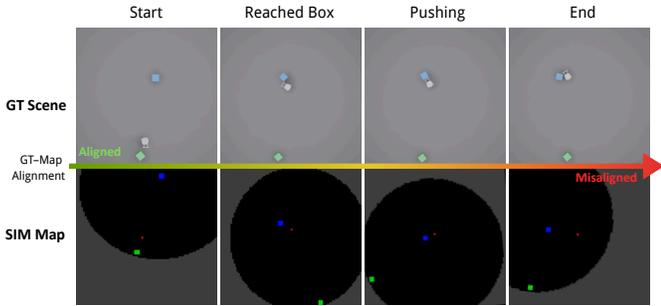
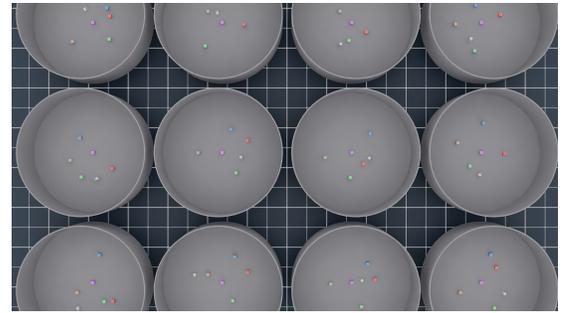


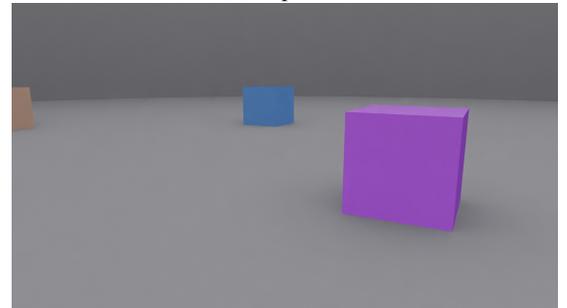
Fig. 8: **GT scene vs SIM map misalignment over time.** Top row: ground-truth scene. Bottom row: corresponding SIM local map used by the policy. From left to right, the estimated map evolves from *aligned* to *diverged*, yielding increasingly inconsistent interaction decisions under SIM-Odom.

mask applied to all 3D keypoints and (ii) a center-gated visibility (C-GV) that controls access to the reference target keypoints  $P_t^{ref}$ , as shown in Fig. 10.

a) *Coordinate conventions.*: Let  $\mathbf{T}_{wr} = [\mathbf{R}_{wr} | \mathbf{t}_{wr}] \in SE(3)$  denote the robot (camera) pose in the world, mapping robot/camera-frame points to world:  $\mathbf{p}_w = \mathbf{R}_{wr}\mathbf{p}_r + \mathbf{t}_{wr}$ . We use a camera-aligned robot frame where  $x$  points forward,  $y$  points left, and  $z$  points up. Normalized image-plane coordi-



(a) Top view



(b) Ego-View

Fig. 9: World Boxes for Parallel Training.

nates  $(u, v) \in [-1, 1]^2$  are defined with  $u$  right-positive and  $v$  down-positive.

## Restricted Observations

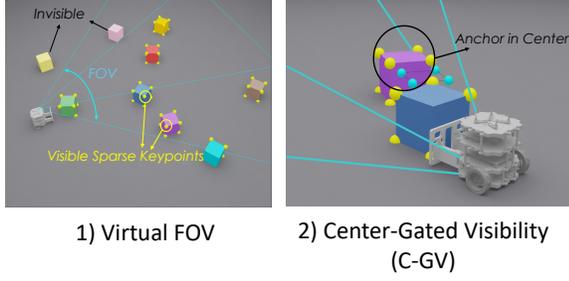


Fig. 10: Two Primary Mechanisms.

**B.3.1 Virtual Egocentric FOV Masking:** Given a world-frame keypoint  $\mathbf{p}_w \in \mathbb{R}^3$ , its camera-frame coordinates are

$$\text{vis}(\mathbf{p}_r) = \mathbb{I} \left[ \begin{array}{l} x > 0 \quad \text{and} \\ |y| \leq x \tan(\theta_h/2) \quad \text{and} \\ |z| \leq x \tan(\theta_v/2) \quad \text{and} \\ x \in [d_{\min}, d_{\max}] \end{array} \right]. \quad (17)$$

We use a constant mask value  $\epsilon \in \mathbb{R}^3$  (e.g.,  $\epsilon = [-10, -10, -10]^\top$ ). The masked 3D observation of a keypoint is

$$\tilde{\mathbf{p}}_r = \begin{cases} \mathbf{p}_r, & \text{if } \text{vis}(\mathbf{p}_r) = 1 \\ \epsilon, & \text{otherwise.} \end{cases} \quad (18)$$

For C-GV we also compute normalized image-plane coordinates (pinhole, using only FOV):

$$u = \frac{y}{x \tan(\theta_h/2)}, \quad v = \frac{-z}{x \tan(\theta_v/2)}. \quad (19)$$

Note that  $(u, v)$  is only used for gating; the teacher policy input uses the masked 3D keypoints in equation 18.

*a) Fixed-size grouping.:* Each object provides  $K$  canonical keypoints in its local frame; we transform them to world and then apply equation 17–equation 18. We keep a fixed maximum number of obstacles  $N_{obs}^{max}$ , sort obstacles by increasing distance to the robot (ties broken by object id), and pad missing obstacles with  $\epsilon$ ; extra obstacles are truncated.

**B.3.2 Center-gated Visibility (C-GV) for  $P_t^{ref}$ :**  $P_t^{ref}$  is a set of  $K$  reference keypoints attached to the *target configuration* of the active object relative to the anchor. At time  $t$ , we instantiate these reference keypoints in the world by composing the anchor pose with the predefined relative transform for the task category, and then convert them to the camera frame and mask them using equation 18.

To prevent exploiting  $P_t^{ref}$  while ignoring the anchor, we reveal  $P_t^{ref}$  *only if* the anchor lies (i) within the virtual FOV and (ii) inside a central gate in the image plane. Let  $\mathbf{c}_{anc,w}$  be the anchor centroid in world (we use the object’s pose center; if unavailable, the mean of its  $K$  keypoints). Compute  $\mathbf{c}_{anc,r}$  and its  $(u_{anc}, v_{anc})$  using equation 19. Define

$$\mathbb{I}_{gate} = \mathbb{I}[\text{vis}(\mathbf{c}_{anc,r}) = 1] \cdot \mathbb{I}[|u_{anc}| \leq u_{gate}] \cdot \mathbb{I}[|v_{anc}| \leq v_{gate}], \quad (20)$$

where  $(u_{gate}, v_{gate})$  specifies the central gated region. Finally, the teacher’s observable reference keypoints are

$$\tilde{P}_t^{ref} = \begin{cases} \mathcal{M}_{3d}(P_t^{ref}), & \text{if } \mathbb{I}_{gate} = 1 \\ \{\epsilon\}_{k=1}^K, & \text{otherwise,} \end{cases} \quad (21)$$

where  $\mathcal{M}_{3d}(\cdot)$  applies equation 18 element-wise to the  $K$  reference keypoints. Under these constraints, the teacher observes grouped masked keypoints  $\{\tilde{P}_t^{act}, \tilde{P}_t^{anc}, \tilde{P}_t^{obs}, \tilde{P}_t^{ref}\}$  and the previous action  $a_{t-1}$ .

## B.4 Front Pusher on Robot

To address the sensing dead zone (approx. 15 cm) of the depth camera, a pusher is integrated into the robot base as shown in Fig. 11. This end-effector features a 7.5 cm protrusion and a 14 cm  $\times$  10 cm frontal profile. This design ensures that objects remain within the reliable depth-sensing manifold during interaction, while maintaining a form factor consistent with the robot’s width.

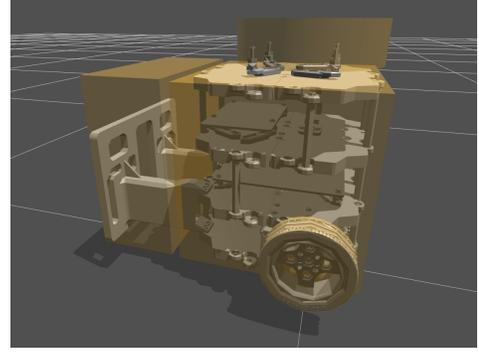


Fig. 11: Robot with Pusher and its Collider.

While the 7.5 cm protrusion effectively mitigates the depth camera’s sensing dead zone, it introduces a significant longitudinal offset between the contact interface and the robot’s Instantaneous Center of Rotation (ICR). This extension of the moment arm (lever arm) dictates that any off-center contact force results in a disproportionately large parasitic yaw moment. Consequently, even minor misalignments in the pushing vector are amplified into substantial rotational disturbances, complicating the maintenance of a stable heading during rectilinear manipulation.

Despite the inherent dynamical instabilities introduced by the extended pusher, our learning-based agent demonstrates remarkable performance and adaptability. This success underscores the robustness of the reinforcement learning framework in internalizing and compensating for complex, non-linear contact dynamics. The policy effectively learns to modulate its control signals to suppress the parasitic yaw moments, achieving precise manipulation even within a constrained dynamical manifold.

Interestingly, an emergent behavior was observed during our ablation study on Restricted Observations for the RL Teacher. When the teacher is granted unconstrained global perception (rendering the depth camera’s sensing dead zone irrelevant), it consistently converges to an unconventional strategy: pushing

the object with the robot’s rear chassis—the side without the pusher integrated.

Guided by temporally decayed completion rewards, the global teacher prioritizes execution efficiency and stability. By utilizing the rear side, the agent effectively minimizes the moment arm between the contact point and the robot’s center of rotation. This strategy reduces the sensitivity to alignment errors and eliminates the aforementioned ‘snaking’ effect, allowing for faster and more stable task completion. This phenomenon further validates that the pusher serves as a perceptual scaffold that, while essential for egocentric sensing, imposes a dynamical penalty that the agent must learn to navigate.

### C. REAL WORLD ROBOT SYSTEM SETUP

We used a TurtleBot3 Burger as the mobile robot platform. The robot uses two servo motors and one passive ball caster. The two actuators provide a maximal translation velocity of 0.22 m/s and a maximal rotation velocity of 2.84 rad/s. A 3D-printed pusher was mounted in front of the robot. The heading of the robot is reconfigured to the opposite of the default direction, such that the ball caster is in the front and the two drive motors are in the rear, allowing a longer lever arm for the actuator. Real-world pushers are equipped with bumper strips on both sides, as shown in Fig. 12, to ensure alignment between the real-world pusher and the pusher’s collider box in the simulation, avoid exploiting gaps along the pusher’s sides to manipulate the boxes.

The robot is equipped with an NVIDIA Jetson Nano, running Ubuntu 20.04 with ROS1 for camera and low-level action control. An Intel RealSense D435i camera is connected to Jetson, mounted on the top of the robot to provide RGB-D egocentric view. The RGB sensor of the camera is set to 20cm above and 5cm behind the ball caster, pitching down  $11.5^\circ$  from the horizon. We used a python wrapper of librealsense2 to handle RealSense Camera utilities. The RGB image is captured at a resolution of  $320 \times 240$ . The depth image is captured at a resolution of  $480 \times 270$ , and aligned with the RGB image before down-sampling together to a resolution of  $240 \times 180$ . The camera initially operates at a base frequency of 30Hz, with varying delays depending on the image processing method employed.

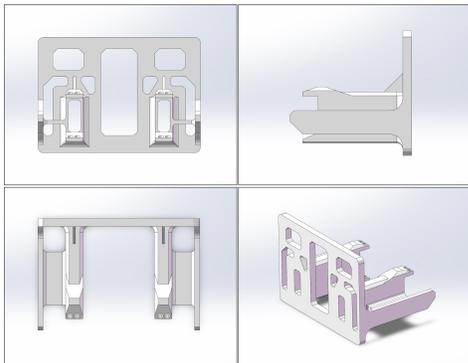


Fig. 12: Pusher of the real robot.

TABLE VI: HSV threshold ranges used for box segmentation in real-world experiments with OpenCV convention. Here, hue is represented on a circular scale with  $H \in [0, 179]$  (hence the red mask wraps around the boundary and is implemented as  $H \in [179, 179] \cup [0, 7]$ ). Saturation and value use  $S \in [0, 255]$  and  $V \in [0, 255]$ .

Color	Hue( $H$ )		Saturation( $S$ )		Value( $V$ )	
	Low	High	Low	High	Low	High
Red	179	7	100	255	100	255
Green	65	85	50	255	55	190
Blue	97	110	80	255	80	255
Violet	150	180	55	155	45	220
Brown	11	22	85	237	65	220

A laptop equipped with an Intel Ultra 9 275HX CPU and an NVIDIA RTX 5080 GPU runs Ubuntu 22.04 and serves as the server for computation. The communication between the robot and the server is established via WebSocket. Specifically, the robot continuously streams the egocentric RGB-D observations to the server over Wi-Fi. On the server side, the received image are decoded and undergo post-processing and segmentation. The resulting processed image patches are used as inputs to the control policy. The policy then performs real-time inference to generate velocity commands, which are transmitted back to the robot at  $\sim 25$ Hz. The robot executes the received velocity commands through its local ROS1 control stack.

### D. IMAGE PROCESSING

#### D.1 RGB Image

The experiments were conducted in a  $3\text{m} \times 3\text{m}$  gray arena containing five boxes of distinct colors: red, green, blue, violet, and brown. Upon receiving the egocentric RGB stream from the robot, the server performs color-based segmentation using HSV thresholding. This process isolates the boxes from the gray background, generating binary masks that are subsequently used to filter the depth images. The specific HSV threshold ranges adopted for each color are detailed in Table VI.

#### D.2 Depth Image

To reduce background clutter and sensor noise irrelevant to the target object, we apply the binary foreground masks obtained from the RGB image to the depth images. In the real-world experiments, the raw depth output from the Intel RealSense D435i suffers from significant noise across the sensing range, and depth measurements for the top surfaces of the boxes are frequently subject to dropouts. To mitigate these issues, we investigated four depth postprocessing strategies. Comparison videos of the different depth processing methods are available on our website: <https://ai4ce.github.io/EgoPush/>. **(A) Learning-based denoising.** We evaluated the state-of-the-art Camera Depth Model (CDM) for RealSense D435i from [22]. This model fuses RGB and depth modalities to guide depth denoising. While CDM yields fine-grained reconstruction and effectively preserves top-surface details, it incurs

a prohibitive inference latency ( $\sim 50$ ms per frame) despite server-side deployment, rendering it unsuitable for our real-time control pipeline.

**(B) Onboard postprocessing.** We followed the depth filtering stack in [50], applying hole-filling, spatial, and temporal filters sequentially to the depth image on Jetson Nano. However, this approach yields suboptimal results that the top surfaces exhibit strong temporal flickering at close range and often disappear entirely at longer distances. Furthermore, the processing latency on the robot’s onboard hardware is relatively high ( $\sim 15$ ms per frame).

**(C) Median-depth filling(ours).** As a baseline, we implemented a constant-fill method that replaces all pixels within the masked region with the median depth value. This approach produces highly stable depth maps and facilitates sim-to-real alignment, performing well in our setup with simple geometric objects. The method is deployed on the server with negligible latency ( $\sim 2$ ms per frame). However, because it discards all intra-object geometric details, its ability to generalize to objects with more complex geometric structures is inherently limited.

**(D) Navier–Stokes inpainting(ours).** Our final implemented solution employs Navier–Stokes inpainting [4] within the masked region, and during simulation training we inject random noise into the depth input to better approximate real-world sensing conditions. Although it recovers less geometric detail than the learning-based CDM, it provides greater stability compared to onboard postprocessing and improved generalization relative to median-depth filling. Empirically, this method achieves a task success rate comparable to the median-filling baseline while retaining significantly more geometric information. It is deployed on the server side with a computational cost similar to the median-fill method ( $\sim 2$ ms per frame).

## E. ADDITIONAL EXPERIMENT RESULTS

### E.1 Different Boxes shapes

We further evaluate the *student* policy on different kinds of object geometries to probe its generalization beyond the training cuboid. Specifically, we replace the active object with a cylinder or a triangular prism, while keeping the sensing setup, action bounds, termination conditions, and evaluation protocol unchanged. As reported in Table VII, the student maintains a near-perfect ReachBox rate for both geometries, indicating that the perception-and-approach sub-skill transfers well across shapes. However, the overall SR drops to 67.48% on the cylinder and 54.30% on the prism, with noticeably longer ExecTime and TrajLen for the prism. The gap between ReachBox and SR suggests that most failures occur in the interaction phase to maintain stable contact and completing the final alignment, where geometry-dependent contact dynamics can amplify small control errors over long horizons. Figure 13 provides representative qualitative rollouts for the two geometries.

TABLE VII: Evaluation Results of the Student Model on Different Object Geometries. **SR** denotes success rate, **Reach** denotes reach-box rate, **ExecTime** denotes execution time per episode, and **TrajLen** denotes trajectory length.

Metric	Cylinder	Prism
SR / % $\uparrow$	67.48	54.30
Reach / % $\uparrow$	99.41	100.00
TrajLen $\downarrow$	9.99	11.00
ExecTime $\downarrow$	381.41	545.55

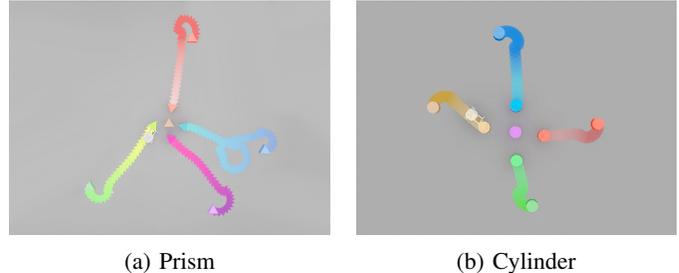


Fig. 13: Qualitative results on different box geometries.

### E.2 Ours accuracy

For the cuboid pushing task, the proposed student model is evaluated using a *nearest-distance-based accuracy metric*. Since the student policy does not have access to the target tag during execution, performance is assessed by measuring the Euclidean distance between the final box position and its corresponding (invisible) target tag in the workspace.

To normalize the error magnitude, the distance is bounded by the training-time distance threshold, which serves as the maximum admissible error. The normalized error is computed as the ratio between the nearest geometric distance and this threshold. Under this evaluation protocol, the student model achieves an average error rate of approximately 13.3% on the cuboid pushing task.

Let  $\mathbf{p}_{\text{box}} \in \mathbb{R}^2$  denote the final planar position of the pushed box, and  $\mathbf{p}_{\text{tag}} \in \mathbb{R}^2$  denote the position of the corresponding target tag. The nearest-distance error is defined as

$$d = \|\mathbf{p}_{\text{box}} - \mathbf{p}_{\text{tag}}\|_2. \quad (22)$$

Given the training-time distance threshold  $\varepsilon_{\text{train}}$ , the normalized accuracy score is computed as

$$\text{Acc} = 1 - \frac{\min(d, \varepsilon_{\text{train}})}{\varepsilon_{\text{train}}}. \quad (23)$$

Accordingly, the corresponding error rate is defined as

$$\text{Error} = \frac{\min(d, \varepsilon_{\text{train}})}{\varepsilon_{\text{train}}}. \quad (24)$$