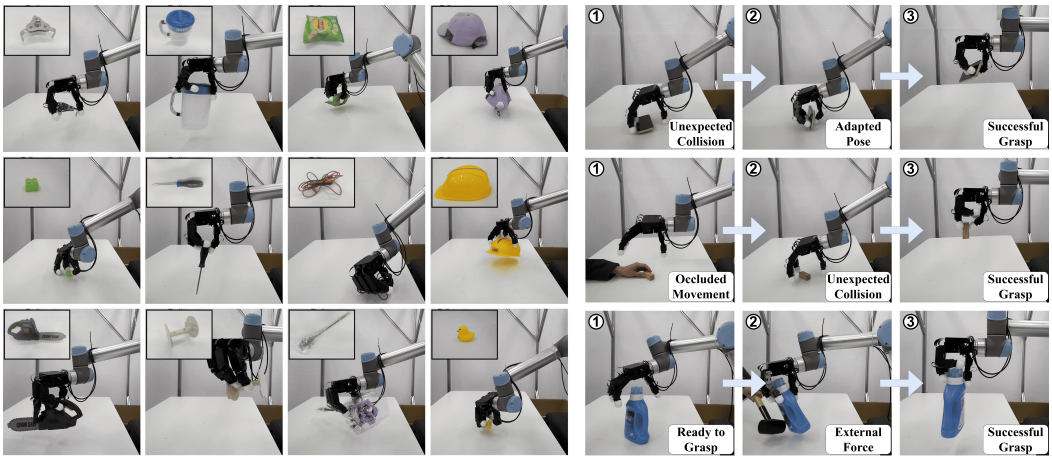
# Robust Dexterous Grasping of General Objects

**Hui Zhang**[1,*] **Zijian Wu**[2,*] **Linyi Huang**[2] **Sammy Christen**[1] **Jie Song**[2,3]

[1]ETH Zurich, Switzerland [2] HKUST (Guangzhou), China [3] HKUST, Hong Kong (China)

https://zdchan.github.io/Robust_DexGrasp/




**(a) Grasping of challenging objects** **(b) Robustness to disturbances**

Figure 1: Our method can robustly grasp 500+ unseen objects with various shapes, materials, and masses, including challenging thin, small, heavy, deformable, or transparent objects (a). It can keep robust grasping under unexpected collision, imprecise object observation, or external forces (b).

**Abstract:** The ability to robustly grasp a variety of objects is essential for dexterous robots. In this paper, we present a framework for zero-shot dynamic dexterous grasping using single-view visual inputs, designed to be resilient to various disturbances. Our approach utilizes a hand-centric object shape representation based on dynamic distance vectors between finger joints and object surfaces. This representation captures the local shape around potential contact regions rather than focusing on detailed global object geometry, thereby enhancing generalization to shape variations and uncertainties. To address perception limitations, we integrate a privileged teacher policy with a mixed curriculum learning approach, allowing the student policy to effectively distill grasping capabilities and explore for adaptation to disturbances. Trained in simulation, our method achieves success rates of 97.0% across 247,786 simulated objects and 94.6% across 512 real objects, demonstrating remarkable generalization. Quantitative and qualitative results validate the robustness of our policy against various disturbances. Additional results and supplementary videos are provided on our project website.

**Keywords:** Dexterous Grasping, Reinforcement Learning, Robot Manipulation

## 1 Introduction

Grasping is a fundamental capability in robotic manipulation, forming the foundation for higher-level tasks such as pick-and-place, articulating hinged objects, tool use, and in-hand manipulation. Dexterous hands, with their high degrees of freedom, enable more flexible and adaptive grasping. Realizing this potential, however, requires precise closed-loop control with real-time feedback to ensure robustness under environmental variations and disturbances. Previous research on dexterous

*Equal Contribution.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

grasping has mainly focused on settings with fully observable or pre-scanned known objects [1, 2], human-to-robot demonstration transfer [3–5], or the execution of predicted static grasping poses or fingertip positions [6–8]. However, these approaches often rely on pre-scanned objects or object-specific human demonstrations, which limit their efficient deployment to novel objects. Moreover, grasping strategies based on static pose execution limit real-time adaptability, resulting in reduced robustness to disturbances, especially when predefined poses become infeasible.

In this paper, we propose a framework for zero-shot dexterous grasping of diverse objects using single-view visual inputs. Similar to human capabilities, our system can adapt in real-time to disturbances such as unexpected collisions arise from noisy observations or external perturbations. Our approach utilizes a hand-centric object shape representation with dynamic distance vectors between finger joints and object surfaces, focusing on potential interactions rather than global shapes. This design improves robustness to shape variations and perceptual uncertainty caused by limited viewpoints. Robust adaptation to disturbances requires continuous observation feedback, yet perception during grasping is often constrained by occlusion and the lack of tactile sensing. We address this challenge through a two-step training paradigm: first training a teacher policy using privileged visual-tactile information, then employing a mixed curriculum learning approach to train a student policy. The curriculum starts with imitation learning to efficiently distill the core grasping behaviors of the teacher policy, and gradually transitions to reinforcement learning to promote exploration of adaptation to disturbances. During student policy training, observation noise and dynamic randomization are introduced to simulate real-world perturbations and encourage robust adaptation.

We conduct comprehensive experiments to evaluate the generalization and robustness of our method. Our approach shows strong generalizability in grasping a wide range of novel objects with random poses on a tabletop. Trained entirely in simulation, it achieves success rates of **97.0%** on **247,786** simulated objects and **94.6%** on **512 real-world** objects. Furthermore, our method facilitates real-time adaptation during grasping, exhibiting superior robustness against disturbances such as unexpected external forces. An ablation study further confirms the importance of key components, validating their contributions to overall performance.

In summary, our contributions are 1) A robust dexterous grasping framework designed for general objects, equipped with real-time adaptation capabilities to handle disturbances effectively. 2) A sparse hand-centric object shape representation tailored for real-world dexterous grasping, which captures interaction potential and facilitates robustness against shape variations and perceptual uncertainty. 3) A mixed curriculum learning method that integrates imitation learning for efficient grasping behavior distillation and reinforcement learning for adaptive motion exploration, under limited perception. 4) Our method, trained in simulation, demonstrates exceptional generalization across 247,786 unseen objects in simulation and 512 unseen objects on a real robot. It also exhibits robust adaptive motions in response to disturbances such as external forces. Additionally, we qualitatively showcase the practical applications enabled by our robust grasping capability.

## 2 Related Work

We list our differences with existing dexterous grasping methods in Tab. 1 for a better comparison.

**Pose-based Dexterous Grasping**. Dexterous grasping is a long-standing research topic [9–11]. Traditional methods typically predict contact points or fingertip positions by optimizing analytical metrics for stable grasping, such as the differentiable approximations of shape closure [2] and force closure [12] metrics. These works mostly require accurate object models [9, 13–15] which limits their generalization ability in real-world deployment. Recently, some works have explored dealing with object shape uncertainty with compliant control algorithms [7, 16–19]. They usually utilize analytical dynamic models to calculate joint torque commands according to static target poses, limiting their robustness to unmodeled disturbances such as inaccurate joint actuator models or external forces. Instead of optimizing grasping poses or fingertip positions, some other methods learn to predict grasping poses from datasets [8, 20]. Overall, these methods usually focus on generating static grasping poses and executing them in an open-loop manner without adaptation, which limits hand

Table 1: Comparison with other dexterous grasping works

| Method | Single-view Observation | Dynamic Grasping | Zero-shot Generalization | Unseen Object Number (Sim) | Unseen Object Number (Real) |
|---|---|---|---|---|---|
| GraspXL [35] | × | ✓ | ✓ | 503,409 | - |
| Agarwal et al.[37] | × | ✓ | × | 2 | 6 |
| SpringGrasp [7] | ✓ | × | ✓ | - | 14 |
| DextrAH-G [38] | ✓ | ✓ | ✓ | - | 30 |
| DexGraspNet2.0 [8] | ✓ | × | ✓ | 1319 | 32 |
| Ours | ✓ | ✓ | ✓ | 247,786 | 512 |

dexterity and robustness to external disturbances, as the hand cannot adaptively change poses when the object moves unexpectedly. In contrast, our method predicts real-time joint actions according to current status, leading to adaptive motions to disturbances and more robust grasping.

**Dynamic Dexterous Grasping**. Rather than predicting static grasping poses, recent works explore real-time joint action prediction for dynamic dexterous grasping. Many of these approaches leverage human demonstrations [3–5, 21, 22] or robot demonstrations [23–27] to learn such actions. However, collecting real-world data remains costly, limiting generalization to out-of-distribution scenarios. Reinforcement learning (RL) has shown promise in handling disturbances across environments, particularly in robot locomotion [28–30]. Through exploration in simulation under disturbances, policies can learn dynamic actions and perform real-time adaptation. However, due to complex hand-object interactions that are challenging to explore and simulate, current RL-based dexterous grasping methods are often limited in specific settings (e.g., category-level generalization [31]) and suffer from sim-to-real gaps. Overall, most RL-based dynamic dexterous grasping methods are still confined to simulation [32–36]. Some RL-based methods leverage human data to simplify exploration and enable deployment on real robots. For example, [37–39] simplify RL exploration with a lower-dimensional action space using PCA components derived from human grasping data, limiting hand dexterity as admitted in their papers. [37] relies on a multi-camera system with only category-level generalization, while [38, 39] use analytical dynamic models for control commands, limiting adaptation to unmodeled disturbances. Huang et al. [1] leverage human grasping poses while relying on known object meshes. Overall, requirements for specific human data, accurate dynamic models, and known objects limit their potential to scale up or adapt to unmodeled disturbances. Our method, in contrast, can be deployed on real robots with a single camera, achieving zero-shot generalization to 512 unseen real-world objects without any human data or known object meshes, while performing real-time proprioception-based adaptive motions to external disturbances.

## 3 Method

We aim to tackle dexterous grasping of various unseen objects using single-view visual perception. Given a single-view object point cloud, we control a robotic arm with a dexterous hand to grasp the object while adapting to disturbances. Fig. 2 illustrates our pipeline. We first train a teacher policy using reinforcement learning (RL) with access to real-time, fully observable object point clouds and hand-object contacts and impulses, noted as a visual-tactile policy. Then we train a student policy with perception available on real robots, including single-view point clouds and noisy joint states without tactile perception. The student policy training is driven by a mixed curriculum learning method, starting with imitation learning (IL) to efficiently distill the teacher's grasping capabilities, then gradually transitioning to RL to explore for adaptive motions to noises and disturbances. Both policies output target hand and arm joint positions for low-level PD controllers.

### 3.1 Hand-centric Object Shape Representation

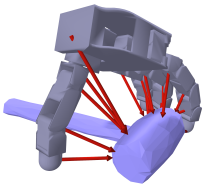


Figure 3: Hand-centric Shape Representation

Effectively capturing shape features that generalize for grasping diverse objects remains a fundamental challenge, especially under shape uncertainty arising from limited viewpoints. To address this, we introduce a sparse hand-centric object representation. As shown in Fig. 3, we construct a 51-dimensional vector by concatenating the distance vectors from

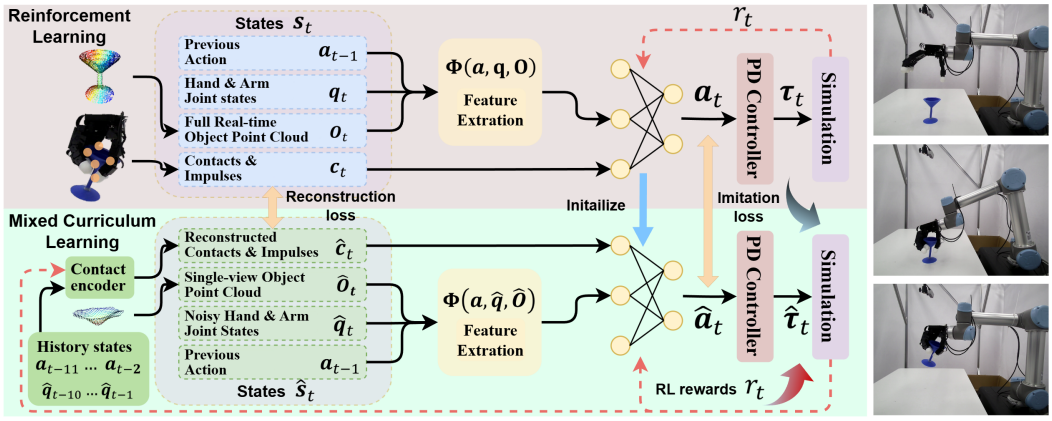



Figure 2: **Framework overview**. We first train a teacher policy with privileged visual-tactile perception driven by RL. Then we train a student policy with single-view object point clouds and noisy proprioception, driven by our mixed curriculum learning method, which starts with IL for efficient teacher policy distillation, and gradually transitions to RL for exploration under disturbances.

each finger joint (including the wrist and fingertips) to their nearest points from the object point clouds. This compact representation efficiently encodes interaction-relevant local shape features instead of fine-grained global geometry, which filters out irrelevant shape variations in non-contact areas, facilitates generalization to novel object shapes, and improves robustness under perceptual uncertainty. Its effectiveness is verified in Sec. 4.4.

### 3.2 Visual-tactile Policy Training

We first train a policy with RL to control the robot arm and dexterous hand, which has access to real-time full object point clouds and hand-object binary contacts and impulses of each finger link.

**Observation**. The observation space of the policy is represented with $\mathbf{s}_t = (\mathbf{a}_{t-1}, \mathbf{q}_t, \mathbf{O}_t, \mathbf{c}_t)$, where $t$ is the current time step, $\mathbf{a}_{t-1}$ is the action of the previous step, $\mathbf{q}_t$ is the current arm and finger joint angles, $\mathbf{O}_t$ is the real-time fully observable object point cloud, $\mathbf{c}_t = \{\mathbf{b}_t, \mathbf{f}_t\}$ includes the binary contact states $\mathbf{b}_t$ and impulses $\mathbf{f}_t$ of each finger link with the object. We omit the notation "$_t$" for simplicity in this section. Given $\mathbf{a}$, $\mathbf{q}$ and $\mathbf{O}$, we extract the following features with a function $\Phi(\mathbf{a}, \mathbf{q}, \mathbf{O}) = (\mathbf{d}, \mathbf{h}, \mathbf{T}, \Delta\mathbf{q})$. Specifically, $\mathbf{d}$ is the vector from each finger joint to the closest point of $\mathbf{O}$. $\mathbf{h}$ is the vertical distance of each arm and hand link to the table. $\mathbf{T}$ is the wrist orientation and position. $\Delta\mathbf{q}$ is the tracking error, which means the difference between current joint angles $\mathbf{q}_t$ and previous action $\mathbf{a}_{t-1}$. The extracted features are then fed to the policy together with $\mathbf{q}$ and $\mathbf{c}$.

**Reward Design**. We design the reward function to encourage robust and safe grasping with the formulation $r = r_{\text{dis}} + r_{\text{contact}} + r_{\text{height}} + r_{\text{reg}}$.

The distance reward $r_{\text{dis}}$ encourages the hand to approach the object by penalizing the link-object distances with $r_{\text{dis}} = -\sum_{i=1}^{L} w_i^d \cdot ||\mathbf{d}_i||^2$, where $w_i^d$ is the weights and $L$ is the number of hand links.

The contact reward $r_{\text{contact}}$ promotes desired hand-object contacts while penalizing undesired contacts, including self-collision, robot-table collision, and arm-object collision, with the formulation $r_{\text{contact}} = \sum_{i=1}^{L} b_i(w_i^{cd} + w_i^{fd} f_i^o) - \sum_{j=1}^{L+M} b_j(w_j^{cu} + w_j^{fu} f_j^u)$. $w_i^{cd}, w_j^{cu}, w_i^{fd}, w_j^{fu}$ are the weights for desired and undesired binary contact states and impulses, $b_i$ and $b_j$ are the desired binary contact states of the $i_{th}$ hand link and undesired binary contact states of the $j_{th}$ hand or arm link, $f_j^o$ and $f_j^u$ are the magnitude of desired hand impulses on the $i_{th}$ link and undesired hand or arm impulses on the $j_{th}$ link, and $M$ is the number of arm links.

The height reward $r_{\text{height}}$ promotes table collision avoidance by discouraging the robot links from excessively approaching the table, achieved by penalizing link-table distances when they are smaller than 2 cm with $r_{\text{height}} = \sum_{i=1}^{L+M} w_i^h \cdot \log(\min\{h_i, 0.02\})$, where $w_i^h$ is the weight and $h_i$ is the vertical distance to the table of the $i_{th}$ link.

The regularization reward $r_{\text{reg}}$ penalizes unnecessary object motion and extreme robot movements with $r_{\text{reg}} = w_h||\dot{\mathbf{T}}_h||^2 + w_o||\dot{\mathbf{T}}_o||^2 + w_l||\mathbf{l}_o|| + w_q||\dot{\mathbf{q}}_a||^2$, where $w_h, w_o, w_l, w_q$ are the weights, $\dot{\mathbf{T}}_h$ and $\dot{\mathbf{T}}_o$ are the linear and angular velocities of the hand and object, $\mathbf{l}_o$ is the object displacement, and $\dot{\mathbf{q}}_a$ is the arm joint velocities.

All weight values can be found in SupMat.

### 3.3 Student Policy Training with Limited Perception

With the teacher policy trained using real-time visual-tactile perception, we further train a student policy with single-view object point clouds and noisy proprioception without tactile information. We randomize the friction coefficients and proprioception of robot joint angles when training the student policy. Besides, we also randomize the PD gains of the low-level joint controllers to simulate unstable hardware actuator stiffness and damping, potentially caused by factors like overheating (especially for finger joint actuators). All randomization parameters can be found in SupMat.

**Observation**. The visual-tactile policy requires real-time fully observable object point clouds $\mathbf{O}_t$ and contact states $\mathbf{c}_t$, which are privileged information inaccessible on the real robots. For hardware deployment, the student policy should utilize the single-view object point cloud $\hat{\mathbf{O}}_t$ and noisy joint state proprioception $\hat{\mathbf{q}}_t$. Specifically, we utilize an LSTM-based encoder to reconstruct the contact states $\hat{\mathbf{c}}_t = \{\hat{\mathbf{b}}_t, \hat{\mathbf{f}}_t\}$ from joint state and action histories. Intuitively, the actions correspond to the joint actuator torques, while the misalignment between the actuator torques and joint state changes can indicate external forces induced by contacts.

**Mixed Curriculum Learning**. While the teacher policy focuses on stable grasping with visual-tactile perception, the student policy must learn to grasp and adapt to disturbances with limited perception. To deal with this challenge, we propose a mixed curriculum learning approach. The training begins with IL using two losses: a contact reconstruction loss $L_{re} = w_{re}(||\hat{\mathbf{b}}_t - \mathbf{b}_t||^2 + ||\hat{\mathbf{f}}_t - \mathbf{f}_t||^2)$ to reconstruct contacts $\mathbf{c}_t$, and an action imitation loss $L_{act} = w_{act}||\hat{\mathbf{a}}_t - \mathbf{a}_t||^2$ to mimic the teacher policy actions. This can help the student policy efficiently distill the grasping capability of the teacher policy. Training then gradually transitions to RL by decreasing $w_{act}$ with a factor $\lambda$ and increasing RL rewards by $1-\lambda$ (keeping $w_{re}$ fixed). This transition encourages the student policy to keep exploration for adaptation to disturbances from observation noises and actuator inaccuracies. The student policy network is initialized with teacher policy weights to accelerate training.

## 4 Experiments

### 4.1 Experimental Setup

**Hardware Setup**. The robot platform used in this work comprises a UR5 [40] robot arm paired with an Allegro [41] dexterous robot hand. We utilize a top-view RealSense D435i camera. An overview of the hardware setup is shown in Fig. 4. The hand and arm are initialized with a partially opened hand heading to object centers from 25cm away, and enclosing the objects with the fingers. The policy runs at 5 Hz as a high-level controller, while the low-level PD controllers for the hand and arm operate at 100 Hz. More details are explained in SupMat.

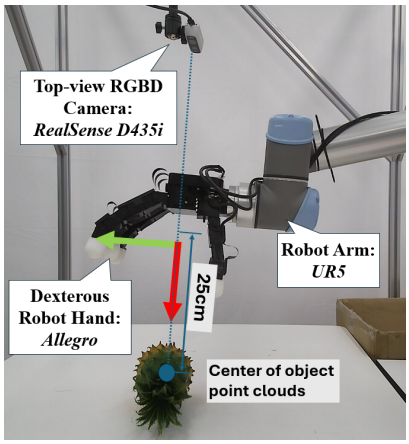


Figure 4: Hardware setup

**Training Details**. Our policy is trained in Raisim [42] simulation, using 35 objects including 3D assets from [43] and scanned objects from [1].

**Metric**. Since our focus is on robust grasping, we use the grasping success rate (Suc. Rate) as our evaluation metric. Specifically, a grasp is considered successful only if the object can be lifted to a height of 0.1 meters and remains stable without falling for at least 5 seconds.

Table 2: Large-scale evaluation (Sim)

| Size | Small | Medium | Large | Total |
|---|---|---|---|---|
| Object Number | 38,493 | 100,435 | 108,858 | 247,786 |
| Success Rate | 0.949 | 0.972 | 0.976 | 0.970 |

Table 3: Large-scale evaluation (Real)

| Category | Suc. Rate | Category | Suc. Rate | Category | Suc. Rate | Category | Suc. Rate |
|---|---|---|---|---|---|---|---|
| Picnic Models | 0.902 | Building Blocks | 0.963 | Fruit & Vegetable Models | 0.962 | Tool Models | 0.875 |
| Animal Models | 0.907 | Toy Cars | 0.979 | Wooden Models | 0.940 | Snacks | 0.974 |
| Bottles & Boxes | 0.970 | Real Tools | 0.893 | Deformable Objects | 0.957 | Other Daily Objects | 0.971 |
| Average | 0.946 | | | | | | |

### 4.2 Large Scale Evaluation

To demonstrate the strong generalization ability of our method, we evaluate it using 512 real objects and 247,786 simulated objects, which are all unseen during training.

**Simulation Evaluation**. We begin by evaluating our method on objects from the Objaverse [44] dataset. Following [35], we scale the objects into three sizes: small, medium, and large. Objects that are infeasible to grasp from a tabletop setting (i.e., diameter $<$ 1cm, height $<$ 2cm, or width $>$ 15cm) are excluded, resulting in a final test set of 247,786 objects. Preprocessing details are provided in SupMat. Objects are randomly placed on the table for grasping. As shown in Tab. 2, our method achieves an overall success rate of 0.970, with consistently high performance across different object sizes. Notably, while grasping smaller objects is more challenging due to the relatively large link dimensions of the Allegro hand, our method still achieves a 0.949 success rate on small objects. These results highlight the strong generalization capability of our approach across diverse object shapes and scales, which we attribute to the proposed sparse hand-centric shape representation.

**Hardware Evaluation**. We further assess the effectiveness of our method in grasping a wide range of novel real-world objects. To this end, we collect a set of 512 objects from different categories, with variations in shape, weight (7 g to 610 g), material (e.g., plastic, styrofoam, rubber, wood, glass, paper, metal, cloth, sponge), and size (from 3.5×3.5×1.5 cm to 40×20×13 cm). Detailed object specifications are provided in SupMat. Each object is evaluated in three random table-top poses. The average success rate per category is reported in Tab. 3. Using only single-view depth observation without any prior knowledge of the objects, our method achieves an overall success rate of 0.946. Notably, despite being trained exclusively on rigid objects in simulation, the learned policy generalizes to deformable objects, demonstrating robust sim-to-real transfer. We attribute this generalization to the reconstructed tactile information along finger links, which enables adaptive finger closure until a stable force-closure grasp is formed. Most failure cases are caused by two factors: noisy point cloud observations for thin or small objects, and insufficient finger torques for heavy or slippery objects due to hardware limitations of the actuators.

### 4.3 Method Comparison

One key advantage of RL-based grasping is its real-time adaptation ability, especially under disturbances. To verify this, we first evaluate our method in real-world using 30 3D-printed ShapeNet [45] objects (shown in SupMat), compared with the following baselines: 1) a naive controller that moves the hand above the objects and gradually closes the fingers. 2) DexGraspNet [2] (with an extra table collision penalty term), with the optimized grasping poses executed by PD controllers. We also evaluate our method with two state-of-the-art methods using their settings and objects: 3) SpringGrasp [7], an optimization-based compliant grasping method. 4) DextrAH-G [38], an RL-based dynamic grasping method. To further verify our robustness to disturbances, we compare our method with DexGraspNet under external forces in both simulation and real-world settings. Each object is tested across five random poses. More details of the baselines are in SupMat.

Table 4: Method comparison (Real)

| Method | Suc. Rate | Method | Suc. Rate | Method | Suc. Rate | Method | Suc. Rate |
|---|---|---|---|---|---|---|---|
| Naive baseline | 0.633 | DexGraspNet [2] | 0.607 | SpringGrasp [7] | 0.771 | DextrAH-G [38] | 0.927 |
| Ours | 0.920 | Ours | 0.920 | Ours | 0.957 | Ours | 0.964 |

Table 5: Robustness to external forces (Sim & Real)

| Method | No Disturbance | 2.5 N Force | Method | No Disturbance | 2.5 N Force |
|---|---|---|---|---|---|
| DexGraspNet [2] (Sim) | 0.667 | 0.553 | DexGraspNet [2] (Real) | 0.607 | 0.480 |
| Ours (Sim) | 0.953 | 0.920 | Ours (Real) | 0.920 | 0.840 |

**Real-world Comparison without External Disturbances**. We first compare the real-world performance of different methods without explicit external disturbances. It is worth noting, however, that hardware experiments inherently involve disturbances due to noisy joint states, object observations, and actuator inaccuracies (e.g., caused by overheating). Results are shown in Tab. 4. Our method consistently outperforms all baselines, regardless of whether being evaluated on our or their settings and object sets, highlighting the enhanced robustness of our RL-based policy. We argue that robust grasping requires real-time adaptive control over finger selection, timing, and joint torques to accommodate diverse object properties, which are effectively learnt by our RL-based policy. In contrast, the naive baseline and DexGraspNet often fail due to limited adaptability, particularly with thin, smooth, or round objects, and may knock objects over due to misaligned finger contact. SpringGrasp optimizes fingertip positions, initial hand poses, and controller gains, allowing the hand to grasp the object compliantly. However, it still lacks the capability to adjust grasping poses according to real-time status, leading to lower performance compared with our method. Among all baselines, DextrAH-G achieves the closest performance to ours, benefiting from its RL-based dynamic grasping. However, as acknowledged in their paper, its PCA-constrained action space restricts dexterity, and its model-based table collision avoidance reduces performance on thin objects. In contrast, our method preserves the full degree-of-freedom of the action space and actively handles rather than rigidly avoids table contacts, leading to improved performance for thin objects as Fig. 1 shows.

**Robustness to External Forces**. We further evaluate the robustness of our method against external forces applied to objects after they have been grasped, comparing the performance with DexGraspNet. In simulation, we apply an external force of 2.5 N with random directions at a random point on the object. In the hardware experiment, we place a 250 g weight at a randomly selected location on the object to introduce a 2.5 N downward force as a controlled external disturbance. As shown in Tab. 5, our method achieves superior success rates with smaller performance degradation under disturbances in both simulation and real-world environments, indicating enhanced robustness against external forces. We argue the robustness comes from our policy's ability to consistently generate stable force-closure grasps and adapt in real time to in-hand object slips. This capability is reinforced through randomized PD gains and friction coefficients during training, requiring the student policy to maintain stable grasps under dynamic disturbances. In contrast, the baseline struggles to consistently achieve robust force-closure grasps and perform real-time adaptation, especially for thin objects. We observe that heavier objects are more susceptible to external forces for both methods, especially in hardware experiments due to the physical torque limits of the finger joint actuators.

### 4.4 Ablation and Analysis

We conduct an ablation study to assess the impact of individual components in both simulation and the real world. We use the same 30 ShapeNet objects as in Sec. 4.3, with 5 random poses per object.

**Simulation Ablation**. In simulation, we first evaluate the effectiveness of our mixed curriculum learning approach by comparing against the following variants that train the student policy: 1) without the RL rewards (W.o. RL rewards), 2) without the action imitation loss (W.o. IL loss), 3) using a fixed ratio for the RL rewards and imitation loss (W.o. Curriculum). Secondly, to examine the necessity of a privileged teacher policy, we include a variant that directly trains the student policy

Table 6: Ablation (Sim & Real)

| | Setting | Suc. Rate | Setting | Success Rate |
|---|---|---|---|---|
| Sim | W.o. RL rewards | 0.907 | W.o. IL loss | 0.933 |
| | W.o. Curriculum | 0.913 | W.o. Priv. learning | 0.773 |
| | Ours | 0.953 | Teacher policy | 0.960 |
| Real | Ours | 0.920 | Full Point Cloud | 0.933 |

from scratch using only RL rewards and contact reconstruction loss (W.o. Priv. learning). Finally, we compare our student policy with the teacher policy to validate its effectiveness (Teacher policy).

The results are presented in Tab. 6. Compared to our full method, the variants without RL rewards and curriculum exhibit a similar decline in performance, underscoring the necessity of the exploration enabled by RL. The marginally lower success rate achieved without IL loss indicates its contribution to student policy training. Overall, starting with a larger IL loss facilitates a rapid distillation of the grasping capabilities from the teacher policy, while subsequently increasing RL rewards encourages effective exploration for adaptation to disturbances. This illustrates the effectiveness of our mixed curriculum learning approach. Training the student policy from scratch (W.o. Priv. learning) yields substantially lower performance and increased sensitivity to hyperparameters, underscoring the importance of the privileged teacher in providing effective initialization and supervision during early training stages. The comparable success rates between our student and teacher policies further validate the effectiveness of our mixed curriculum learning method.

**Real-world Ablation**. We further compare our original setting with a variant that utilizes real-time, fully observable object point clouds in the real world (Full Point Cloud), which we get from FoundationPose [46] along with known object meshes. In contrast, our approach uses single-view object point clouds captured by the camera without object meshes, which can be directly applied to novel objects for zero-shot generalization. The results are shown in Tab. 6, where the highly comparable performances between the two settings highlight the effectiveness of our sparse object-centric representation in extracting meaningful shape features from single-view object observations.

### 4.5 Qualitative Results

We qualitatively demonstrate the generalization and robustness of our method in Fig. 1. Our policy can deal with various challenging objects, and adaptively adjust grasping poses in response to unexpected collisions caused by noisy joint states, inaccurate object observations, or actuator imprecision. It also effectively compensates for disturbances and maintains stable grasps under significant external forces. Additional qualitative results are provided in the Suppl. Video, where we further showcase various applications enabled by our robust grasping capability with different modules integrated, such as grasping in cluttered scenes with a segmentation model [47], task-driven grasping with a VLM planner [48], and grasping moving objects with an object tracking module [49].

## 5 Conclusion

In this paper, we propose an RL-based framework for robust dexterous grasping from single-view perception. Our method exhibits strong generalization capability, achieving success rates of 94.6% on 512 real-world objects and 97.0% on 247,786 simulated objects. Compared to existing methods, our RL-based dynamic grasping approach exhibits superior robustness and adaptability, especially under randomly applied external forces and inherent internal disturbances such as observation noises or actuator inaccuracies, which can lead to unexpected collisions and require pose adaptation. We show that a sparse, hand-centric object shape representation enables effective generalization across diverse object geometries under perceptual uncertainty. Furthermore, our mixed curriculum learning framework facilitates the policy to learn real-time adaptive motions under varying disturbances with limited perception. These components together constitute a robust, low-level grasping controller, providing the foundation for diverse downstream applications as demonstrated in our Suppl. Video.

## 6 Limitations

Although our method demonstrates strong generalization and robustness across a wide range of objects, it still has several limitations that we plan to address in future work. First, despite its capability to grasp objects with diverse sizes and physical properties, the large link dimensions of the Allegro Hand make it difficult to grasp very small objects (e.g., with diameters less than 1.5 cm). Solving this limitation relies on a smaller dexterous robot hand with a similar scale to the human hand. Second, although our policy can grasp moving objects as demonstrated in the Suppl. Video, it struggles in highly dynamic environments such as catching a flying object. This limitation could potentially be addressed by incorporating an explicit object motion prediction module. Finally, our framework focuses specifically on robust grasping without addressing non-grasping interactions like pushing objects, which are also essential in many manipulation tasks. Such interactions may be enabled in the future through task-specific reward designs and policy training.

## References


[1] L. Huang, H. Zhang, Z. Wu, S. Christen, and J. Song. FunGrasp: Functional Grasping for Diverse Dexterous Hands. *arXiv preprint arXiv:2411.16755*, 2024.

[2] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, and H. Wang. DexGraspNet: A Large-Scale Robotic Dexterous Grasp Dataset for General Objects Based on Simulation. *International Conference on Robotics and Automation (ICRA)*, 2023.

[3] Z. Q. Chen, K. Van Wyk, Y.-W. Chao, W. Yang, A. Mousavian, A. Gupta, and D. Fox. Dextransfer: Real world multi-fingered dexterous grasping with minimal human demonstrations. *arXiv preprint arXiv:2209.14284*, 2022.

[4] Y. Chen, C. Wang, Y. Yang, and K. Liu. Object-Centric Dexterous Manipulation from Human Motion Data. In *Conference on Robot Learning (CoRL)*, 2024.

[5] K. Li, P. Li, T. Liu, Y. Li, and S. Huang. ManipTrans: Efficient Dexterous Bimanual Manipulation Transfer via Residual Learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2025.

[6] M. Attarian, M. A. Asif, J. Liu, R. Hari, A. Garg, I. Gilitschenski, and J. Tompson. Geometry Matching for Multi-Embodiment Grasping. In *Conference on Robot Learning (CoRL)*, 2023.

[7] S. Chen, J. Bohg, and C. K. Liu. SpringGrasp: An optimization pipeline for robust and compliant dexterous pre-grasp synthesis. *Robotics: Science and Systems (RSS)*, 2024.

[8] J. Zhang, H. Liu, D. Li, X. Yu, H. Geng, Y. Ding, J. Chen, and H. Wang. DexGraspNet 2.0: Learning Generative Dexterous Grasping in Large-scale Synthetic Cluttered Scenes. *Conference on Robot Learning (CoRL)*, 2024.

[9] C. Ferrari and J. Canny. Planning optimal grasps. *International Conference on Robotics and Automation (ICRA)*, 1992.

[10] S. Kim, A. Shukla, and A. Billard. Catching Objects in Flight. *Transactions on Robotics (T-RO)*, 2014.

[11] T. Takahashi, T. Tsuboi, T. Kishida, Y. Kawanami, S. Shimizu, M. Iribe, T. Fukushima, and M. Fujita. Adaptive grasping by multi fingered hand with tactile sensor based on robust force and position control, 2008.

[12] T. Liu, Z. Liu, Z. Jiao, Y. Zhu, and S.-C. Zhu. Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator. *Robotics and Automation Letters (RA-L)*, 2021.

[13] M. T. Ciocarlie, C. Goldfeder, and P. K. Allen. Dexterous Grasping via Eigengrasps: A Low-dimensional Approach to a High-complexity Problem. 2007. URL https://api.semanticscholar.org/CorpusID:6853822.

[14] A. Miller and P. Allen. Graspit! A versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 2004.

[15] M. A. Roa and R. Suárez. Grasp quality measures: review and performance. *Auton. Robots*, 2015.

[16] M. Kazemi, J.-S. Valois, J. A. Bagnell, and N. Pollard. Robust Object Grasping using Force Compliant Motion Primitives. *Robotics: Science and Systems (RSS)*, 2012.

[17] M. Li, K. Hang, D. Kragic, and A. Billard. Dexterous grasping under shape uncertainty. *Robotics and Autonomous Systems*, 2016.

[18] M. Pfanne, M. Chalon, F. Stulp, H. Ritter, and A. Albu-Schäffer. Object-Level Impedance Control for Dexterous In-Hand Manipulation. *Robotics and Automation Letters (RA-L)*, 2020.

[19] S. Puhlmann, J. Harris, and O. Brock. RBO Hand 3: A Platform for Soft Dexterous Manipulation. *Transactions on Robotics (T-RO)*, 2022.

[20] L. Zhang, K. Bai, G. Huang, Z. Chen, and J. Zhang. Multi-fingered Robotic Hand Grasping in Cluttered Environments through Hand-object Contact Semantic Mapping. *arXiv preprint arXiv:2404.08844*, 2024.

[21] J. Ye, J. Wang, B. Huang, Y. Qin, and X. Wang. Learning continuous grasping function with a dexterous hand from human demonstrations. *Robotics and Automation Letters (RA-L)*, 2023.

[22] J. Li, Y. Zhu, Y. Xie, Z. Jiang, M. Seo, G. Pavlakos, and Y. Zhu. OKAMI: Teaching Humanoid Robots Manipulation Skills through Single Video Imitation. *Conference on Robot Learning (CoRL)*, 2024.

[23] S. Yang, M. Liu, Y. Qin, D. Runyu, L. Jialong, X. Cheng, R. Yang, S. Yi, and X. Wang. ACE: A Cross-platfrom Visual-Exoskeletons for Low-Cost Dexterous Teleoperation. *arXiv preprint arXiv:2408.11805*, 2024.

[24] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang. Open-TeleVision: Teleoperation with Immersive Active Visual Feedback. *Conference on Robot Learning (CoRL)*, 2024.

[25] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. Fan, and Y. Zhu. DexMimicGen: Automated Data Generation for Bimanual Dexterous Manipulation via Imitation Learning. *arXiv preprint arXiv:2410.24185*, 2024.

[26] H.-S. Fang, H. Yan, Z. Tang, H. Fang, C. Wang, and C. Lu. AnyDexGrasp: General Dexterous Grasping for Different Hands with Human-level Learning Efficiency. *arXiv preprint arXiv:2502.16420*, 2025.

[27] Y. Zhong, X. Huang, R. Li, C. Zhang, Y. Liang, Y. Yang, and Y. Chen. DexGraspVLA: A Vision-Language-Action Framework Towards General Dexterous Grasping. *arXiv preprint arXiv:2502.20900*, 2025.

[28] S. Choi, G. Ji, J. Park, H. Kim, J. Mun, J. H. Lee, and J. Hwangbo. Learning quadrupedal locomotion on deformable terrain. *Science Robotics*, 2023.

[29] X. Gu, Y.-J. Wang, X. Zhu, C. Shi, Y. Guo, Y. Liu, and J. Chen. Advancing Humanoid Locomotion: Mastering Challenging Terrains with Denoising World Model Learning. *Robotics: Science and Systems (RSS)*, 2024.

[30] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 2022.

[31] Y. Qin, B. Huang, Z.-H. Yin, H. Su, and X. Wang. DexPoint: Generalizable Point Cloud Reinforcement Learning for Sim-to-Real Dexterous Manipulation. *Conference on Robot Learning (CoRL)*, 2022.

[32] S. Christen, M. Kocabas, E. Aksan, J. Hwangbo, J. Song, and O. Hilliges. D-grasp: Physically plausible dynamic grasp synthesis for hand-object interactions. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[33] W. Wan, H. Geng, Y. Liu, Z. Shan, Y. Yang, L. Yi, and H. Wang. UniDexGrasp++: Improving Dexterous Grasping Policy Learning via Geometry-Aware Curriculum and Iterative Generalist-Specialist Learning. In *International Conference on Computer Vision (ICCV)*, 2023.

[34] H. Yuan, B. Zhou, Y. Fu, and Z. Lu. Cross-Embodiment Dexterous Grasping with Reinforcement Learning. *arXiv preprint arXiv:2410.02479*, 2024.

[35] H. Zhang, S. Christen, Z. Fan, O. Hilliges, and J. Song. GraspXL: Generating Grasping Motions for Diverse Objects at Scale. In *European Conference on Computer Vision (ECCV)*, 2024.

[36] H. Zhang, S. Christen, Z. Fan, L. Zheng, J. Hwangbo, J. Song, and O. Hilliges. ArtiGrasp: Physically Plausible Synthesis of Bi-Manual Dexterous Grasping and Articulation. In *International Conference on 3D Vision (3DV)*, 2024.

[37] A. Agarwal, S. Uppal, K. Shaw, and D. Pathak. Dexterous Functional Grasping. In *Conference on Robot Learning (CoRL)*, 2023.

[38] T. G. W. Lum, M. Matak, V. Makoviychuk, A. Handa, A. Allshire, T. Hermans, N. D. Ratliff, and K. V. Wyk. DextrAH-G: Pixels-to-Action Dexterous Arm-Hand Grasping with Geometric Fabrics. In *Conference on Robot Learning (CoRL)*, 2024.

[39] R. Singh, A. Allshire, A. Handa, N. Ratliff, and K. V. Wyk. DextrAH-RGB: Visuomotor Policies to Grasp Anything with Dexterous Hands. *arXiv preprint arXiv:2412.01791*, 2024.

[40] Universal Robots. UR5. `https://www.universal-robots.com/products/ur5-robot/`.

[41] Wonik Robotics. Allegro robot hand. `https://www.wonikrobotics.com/robot-hand`.

[42] J. Hwangbo, J. Lee, and M. Hutter. Per-contact iteration method for solving contact dynamics. *Robotics and Automation Letters (RA-L)*, 2018.

[43] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar. The YCB object and Model set: Towards common benchmarks for manipulation research. *International Conference on Advanced Robotics (ICAR)*, 2015.

[44] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi. Objaverse: A Universe of Annotated 3D Objects. *arXiv preprint arXiv:2212.08051*, 2022.

[45] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012, 2015.

[46] B. Wen, W. Yang, J. Kautz, and S. Birchfield. FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects. In *Computer Vision and Pattern Recognition (CVPR)*, 2024.

[47] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment Anything. *arXiv preprint arXiv:2304.02643*, 2023.

[48] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond, 2023.

[49] B. Wen and K. E. Bekris. BundleTrack: 6D Pose Tracking for Novel Objects without Instance or Category-Level 3D Models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[50] F. Jonathan. python_UR5_ikSolver.

[51] H. K. Cheng, S. W. Oh, B. Price, J.-Y. Lee, and A. Schwing. Putting the Object Back into Video Object Segmentation. *Computer Vision and Pattern Recognition (CVPR)*, 2024.

# A Experiment Details

## A.1 Large Scale Object Details

In simulation, we utilize the Objaverse [44] objects processed to graspable sizes by [35], which contain different scales: small, medium, and large. Furthermore, as [35] focuses on grasping motion generation without hand-table collision, we filter out the objects that are not feasible to be grasped from the table (diameter $<$ 1cm, height $<$ 2cm, or width $>$ 15cm), leading to 247,786 objects in total.

For hardware experiments, we choose 512 objects from 12 categories as explained in the main paper, which contains objects with various shapes, materials, masses, and sizes. The objects are visualized in Fig. 5, and the physical attributes of each category are listed in Tab. 7.

Figure 5: Real objects used for large-scale evaluation

Table 7: Physical Attributes of Real Objects

| Category | Num. | Mass (g) | Scale (mm) | Material |
|---|---|---|---|---|
| Picnic Models | 41 | 9-257 | 50*50*40 - 280*170*100 | Plastic |
| Building Blocks | 54 | 10-180 | 40*30*30 - 200*200*100 | Plastic/Styrofoam |
| Fruit & Vegetable Models | 35 | 7-196 | 70*50*50 - 330*80*80 | Plastic/Styrofoam/Rubber |
| Tool Models | 16 | 20-270 | 70*40*15 - 400*140*130 | Plastic |
| Animal Models | 18 | 26-165 | 100*50*30 - 230*120*100 | Rubber |
| Toy Cars | 16 | 40-117 | 90*35*30 - 110*100*70 | Plastic |
| Wooden Models | 78 | 25-218 | 50*30*30 - 170*90*90 | Wood |
| Snacks | 38 | 22-570 | 60*40*40 - 350*130*70 | - |
| Bottles & Boxes | 67 | 15-550 | 35*35*30 - 240*170*55 | Plastic/Glass/Paper |
| Real Tools | 50 | 16-610 | 40*40*30 - 130*100*90 | Metal/Plastic/Rubber |
| Deformable Objects | 31 | 8-142 | 70*50*50 - 220*180*90 | Rubber/Cloth/Sponge/Styrofoam |
| Other Daily Objects | 68 | 19-454 | 40*30*30 - 270*130*100 | - |
| Total | 512 | 7-610 | 35*30*15 - 400*200*130 | - |

## A.2 Baseline Details

**DexGraspNet**. Since DexGraspNet optimizes grasping poses without considering tables, we incorporate a table collision loss to generate collision-free poses for a fair comparison. Specifically, given the object pose and corresponding table surface with height $h_{table}$, we add a loss term with the formulation $l_{table} = \sum_{i=0}^{N} ||h_{table} - h_i||^2 \cdot \mathcal{I}_{h_{table} > h_i}$, where $h_i$ is the height of the $i_{th}$ hand joints

and N is the number of joints (including virtual joints for fingertips). To ensure a fair comparison, the initial hand pose for DexGraspNet is set identically to that used in our method. As DexGraspNet leverages full object point clouds, we provide such inputs by using known object meshes and pose estimations from FoundationPose [46] during its hardware experiments. In contrast, our method is evaluated with our original setting using only single-view point clouds without access to object meshes. The 3D-printed ShapeNet [45] objects used for the evaluation are shown in Fig. 6.

Figure 6: Objects used for comparisons and ablation

**SpringGrasp**. Since the exact objects used in SpringGrasp [7] are unavailable, we instead employ highly similar alternatives, as illustrated in Fig. 7. Given that both our method and SpringGrasp rely solely on depth information without RGB input, differences in object textures have minimal impact on performance. In fact, the objects used in our evaluation may pose greater challenges due to reflective surfaces or transparent materials, which introduce additional noise in the depth observations. Focusing on robust grasping, we compare the success rates based on strictly successful grasps. Following a consistent evaluation criterion, we label partially successful grasps defined in SpringGrasp (objects are lifted but subsequently slip or slide) as failures for both methods.

(a) (b)

Figure 7: Objects used for comparison with SpringGrasp [7] (a) and in their original experiment (b).

# B Implementation Details

## B.1 Initial Pose

We propose a simple yet effective method to initialize the arm and hand joints (see Fig. 4), which are easily reachable via inverse kinematics (IK) planning since the hand does not interact with the object during initialization. The hand is initialized with partially open fingers using fixed joint angles $\mathbf{q}_0 =$ [0.2, 0.6, 0.2, 0.5, 0.2, 0.6, 0.2, 0.5, 0.2, 0.6, 0.2, 0.5, 1.3, 0.0, -0.1, 0.2]. To initialize the arm pose, we define a heading direction $\mathbf{x}$ orthogonal to the palm (red arrow), pointing from the camera toward the object point cloud center $\mathbf{c}$. This allows the hand to approach the object from a direction with minimal shape uncertainty. Next, we define a palm direction $\mathbf{y}$ (green arrow) aligned with the final link of the arm. It is set to be orthogonal to $\mathbf{x}$ while minimizing the projection length of the object point cloud along $\mathbf{y}$. This helps the hand enclose the object from a narrow edge, which facilitates easier grasping. The 6D wrist pose is finally defined by $\mathbf{x}$, $\mathbf{y}$, and an offset 25cm from $\mathbf{c}$ along $\mathbf{x}$, which is used to initialize the arm joints by an Inverse Kinematics (IK) solver [50].

## B.2 Object Point Cloud

To extract object point clouds from a continuous visual stream, we first segment the object before grasping using SAM [47], obtaining an initial, unoccluded point cloud. During grasping, we estimate the real-time point cloud under hand occlusion by tracking object pose changes with Cutie [51] and BundleTrack [49]. To mitigate depth noise, particularly for transparent or reflective objects, we apply outlier removal and per-frame smoothing to the initial point cloud over the first 50 frames.

## B.3 Action Space

During grasping, our policy predicts both arm and hand target joint angles as actions. When the objects are grasped during deployment, we set fixed target angles for the arm joints to lift the objects.

## B.4 Domain Randomization

As explained in the main paper, we randomize the environment parameters when training the student policy for robust sim-to-real transfer. The randomized parameters are listed in Tab. 8.

Table 8: Domain randomization parameters.

| Variable | Randomization |
|---|---|
| Friction Coefficient | {0.5, 0.6, 0.7, 0.8,0.9} |
| Hand P Gain | [0.9, 1.1] * 600 |
| Hand D Gain | [0.9, 1.1] * 20 |
| Arm P Gain | [0.5, 1.05] * 1.6e4 |
| Arm D Gain | [0.5, 1.05] * 600 |
| Hand Joint Angles | [-0.02, +0.02]rad + GT |
| Arm Joint Angles | [-0.005, +0.005]rad + GT |
| Arm/Hand Link Position | [-0.01, +0.01]m + GT |
| Arm/Hand Link Orientation | [-0.02, +0.02]rad + GT |

## B.5 Training Details

Using a single NVIDIA RTX 3090 GPU and 12 CPU cores, the training of the teacher and student policies takes approximately 30 hours in total. An overview of important parameters and reward function weights are provided in Tab. 9 and Tab. 10.

Table 9: Hyperparameters

| Hyperparameters PPO | Value |
|---|---|
| Epochs | 1.5e4 |
| Steps per epoch | 70 |
| Environment steps per episode | 63 |
| Batch size | 2000 |
| Updates per epoch | 20 |
| Simulation timestep | 0.01s |
| Simulation steps per action | 20 |
| Discount factor $\gamma$ | 0.996 |
| Max. gradient norm | 0.5 |
| Value loss coefficient | 0.5 |
| Entropy coefficient | 0.0 |
| Hidden units | 128 |
| Hidden layers | 2 |

Table 10: Weights of the Reward Function

| Weights | Value |
|---|---|
| $w^d$ (fingertip) | 2.0 |
| $w^d$ (the other hand links) | 0.5 |
| $w^{cd}$ (fingertip) | 6.0 |
| $w^{cd}$ (the other hand links) | 1.5 |
| $w^{fd}$ (fingertip) | 4.0 |
| $w^{fd}$ (the other hand links) | 1.0 |
| $w^{cu}$ | -1.0 |
| $w^{fu}$ | -0.5 |
| $w^h$ | -0.05 |
| $w_h$ | -1.0 |
| $w_o$ | -15.0 |
| $w_l$ | -5.0 |
| $w_q$ | -1.0 |
| $w_{re}$ | 1.0 |
| $w_{act}$ | 1.0 |
| $\lambda$ | 1.0 - $iter_num$/2000 |