

Retrieval Dexterity: Efficient Object Retrieval in Clutters with Dexterous Hand

Fengshuo Bai^{1,2}, Yu Li^{2,3}, Jie Chu^{1,2}, Tawei Chou^{2,3}, Runchuan Zhu³, Ying Wen^{1,†}, Yaodong Yang^{2,3,†} and Yuanpei Chen^{2,3,†}

¹Shanghai Jiao Tong University, ²PKU-PsiBot Joint Lab, ³Peking University

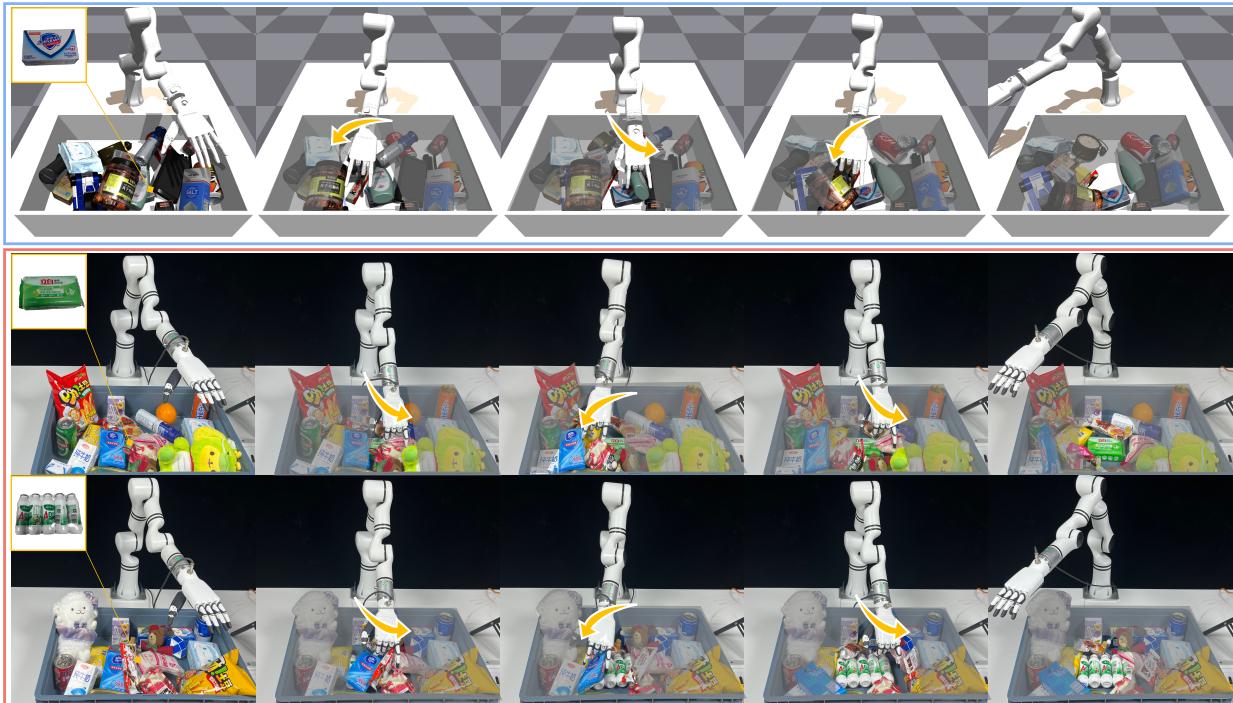


Figure 1 | We present **Retrieval Dexterity**, a system that learns efficient object retrieval in **simulation** and demonstrates zero-shot **real-world** deployment.

Abstract

Retrieving objects buried beneath multiple objects is not only challenging but also time-consuming. Performing manipulation in such environments presents significant difficulty due to complex contact relationships. Existing methods typically address this task by sequentially grasping and removing each occluding object, resulting in lengthy execution times and requiring impractical grasping capabilities for every occluding object. In this paper, we present a dexterous arm-hand system for efficient object retrieval in multi-object stacked environments. Our approach leverages large-scale parallel reinforcement learning within diverse and carefully designed cluttered environments to train policies. These policies demonstrate emergent manipulation skills (e.g., pushing, stirring, and poking) that efficiently clear occluding objects to expose sufficient surface area of the target object. We conduct extensive evaluations across a set of over 10 household objects in diverse clutter configurations, demonstrating superior retrieval performance and efficiency for both trained and unseen objects. Further-

more, we successfully transfer the learned policies to a real-world dexterous multi-fingered robot system, validating their practical applicability in real-world scenarios. Videos can be found on our project website <https://ChangWinde.github.io/RetrDex>.

1. Introduction

Imagine having a box full of miscellaneous items with your desired book at the bottom. Having to remove each overlying object one by one to reach your book is undoubtedly a tedious task. Similar cluttered environments, such as messy drawers, disorganized office desks, or packed warehouses, present significant challenges for object retrieval in robotics. Previous approaches often relied on sequential removal of occluding objects using parallel-jaw grippers before attempting to grasp the target object. However, this strategy is not only time-consuming but also presents significant challenges in achieving reliable grasping manipulation. In such scenarios, humans typically employ their hands to efficiently move other objects aside,

[†]Corresponding author(s): Yuanpei Chen{yuanpei.chen312@gmail.com}, Yaodong Yang{yaodong.yang@pku.edu.cn}, Ying Wen{ying.wen@sjtu.edu.cn}

retrieving the desired object in a relatively short time. This is because the hand, as a high-degree-of-freedom manipulator, offers numerous configurations for object manipulation. To enable efficient object retrieval across diverse objects and scenarios, we propose using a multi-finger dexterous hand as the end-effector.

Learning such skills brings multiple challenges: *(i) Time efficiency:* When objects are only partially visible or completely occluded in the perception system, sequentially grasping and placing objects is highly time-consuming. Determining how to quickly locate and expose sufficient pixels for grasping is a challenging yet important task. Hand-designed strategies often require unacceptably long execution times. *(ii) Diverse objects:* Training a policy that succeeds across various diverse object settings is even more challenging. Previous methods requiring pose estimation of surrounding objects can introduce new errors and struggle to generalize to unseen objects with limited training data. Some prior research typically assumes the existence of a universally successful grasping policy, which may be an even more difficult proposition. *(iii) High-dimensional action space and contact-rich environment:* Dexterous multi-finger hands introduce a high-dimensional state and action space, which increases the difficulty for optimization. Additionally, in cluttered stacking scenarios, object collisions are extremely rich, making environmental dynamics more challenging to model.

In this paper, we propose **Retrieval Dexterity**, a new system for retrieving objects in clutter with one dexterous hand in a super-efficient manner. Our hardware system includes a Realman RM-75 robot equipped with an Inspired Hand (with 6 degrees of freedom) at the end, as shown in Figure 1. To solve this problem, we propose an approach leveraging Reinforcement Learning to train a policy in simulation and then perform Sim2Real transfer on real robots. Specifically, to solve the above challenges, we introduce two key technical contributions:

- We have developed a realistic scene generation pipeline for stacked environments. The system simulates the natural accumulation of common household objects through gravitational dropping, while strategically placing occluded target objects at specified positions. Physical parameters are carefully calibrated to ensure realistic stacking environment.
- We present a reinforcement learning framework for training object retrieval policies. The reward function is based on the pixel visibility of target objects captured by simulated cameras after a fixed time horizon. Using efficient reinforcement learning algorithms, we optimize the policy to

discover emergent retrieval strategies in complex stacked environments.

We conduct both simulation and real-world experiments. The results demonstrate that our method enables the hand to successfully retrieve objects and surpass baselines by a large margin in efficiency. To our knowledge, this is the first work that enables efficient object retrieval with dexterous multi-finger hands.

2. Related Work

2.1. Cluttered Objects Manipulation

Interacting with objects in a cluttered environment is of significant importance for real-world applications [1, 2, 3]. Prior research has extensively explored robotic manipulation in these environments, aiming to equip robots with the ability to master diverse and complex skills. For instance, Murali et al. [3], Pang et al. [4] have focused on improving robust object grasping techniques, while Li et al. [5], Zhao et al. [6] investigate retrieval tasks. Additionally, studies such as Goyal et al. [7], Tang and Sukhatme [8], Jia and Chen [9] address challenges in rearrangement, as well as grasping and throwing [10]. Visual-based approaches have also been widely adopted to enhance manipulation strategies in cluttered environments. For example, Huang et al. [11] leverage visual prediction and planning to forecast the future states of objects after pushing actions, thereby optimizing grasping paths. In a related effort, Kurenkov et al. [12] propose a continuous pushing strategy driven by real-time visual signals to improve object graspability.

2.2. Object Retrieval

Retrieving a target object from complex clutter is a fundamental robotic skill with broad applications, ranging from domestic services to manufacturing. To address this challenge, various studies have proposed solutions from multiple perspectives. For instance, some works focus on planning strategies, such as object search optimization [13], teacher-aided exploration [12], and human-guided planning [14], while others emphasize action-based methods, including push-grasping synergy policies [15] and learning pushing and grasping without visual foresight [16]. Additionally, approaches like analyzing support relations among cluttered objects have shown promise for improving retrieval efficiency [5]. In terms of perception, researchers have explored both tactile sensing [17, 6] and visual or language-based modalities [18, 8]. The choice of end-effector has also been a key focus, with methods employing rod-like pushers [19], parallel grippers [20, 8, 6, 21], and dexterous hands [22] to address the challenges posed by cluttered environments. Moreover, task scenarios vary widely, from

granular media [17] to confined spaces [6], requiring tailored approaches to accommodate environmental constraints. Our approach differs fundamentally from most previous methods by actively manipulating occluding objects to expose the target object, enabling efficient retrieval while introducing more challenging control requirements.

2.3. Reinforcement Learning for Dexterous Manipulation

Dexterous manipulation has remained a cornerstone of robotics research due to its critical role in replicating the sophisticated motor skills humans use to interact with diverse objects and achieve intelligent control [1, 2, 23]. While traditional methods employ analytical dynamic models for trajectory optimization, their simplified treatment of contact dynamics limits their effectiveness in complex tasks. Imitation learning (IL) has demonstrated impressive results in dexterous manipulation tasks [24, 25]. However, IL faces significant challenges due to its reliance on human expert demonstrations, making it resource-intensive and difficult to scale for contact-rich tasks [26, 27, 28]. In contrast, this work trains a generalizable policy using sim-to-real reinforcement learning without using any expert data. Reinforcement Learning (RL) has been widely adopted for robotic manipulation to master complex skills, particularly in unstructured and contact-rich scenarios. RL-based approaches offer two significant advantages: they simplify the controller design process and enable the acquisition of complex skills. For instance, Chen et al. [29] developed an efficient system for in-hand object re-orientation, while Lin et al. [28] proposed a sim-to-real framework for twisting lids of various bottle-like objects using two hands. Similarly, Huang et al. [30] designed a system for efficient bimanual handovers, and additional studies have explored tasks such as spinning pen-like objects [31], sequential block building [22], bimanual manipulation [32, 28] and diverse skills based on Vision-Language Models [33, 34] or exploration [35, 36]. On the other hand, several works [37, 38, 27, 28] demonstrate that RL-based methods can learn emergent dexterous behaviors without additional reward terms. Our work leverages this capability to discover emergent behaviors, where carefully designed reward functions and environmental setups enable autonomous learning of retrieval skills including pushing and poking.

3. Task Formulation

In this paper, we focus on the challenge of retrieving target objects in cluttered environments using a dexterous multi-fingered robot, as shown in Figure 1. The objective of this task is to efficiently expose the target object to the camera’s field of view,

enabling subsequent grasping operations. We then formulate the object retrieval task as a finite horizon Markov Decision Process (MDP), which contains a 5-tuple $(\mathcal{S}, \mathcal{A}, R, P, \gamma)$. \mathcal{S} and \mathcal{A} represent the state and action spaces. $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ represents the stochastic dynamics, which determines the probability of transferring to s' given state s and action a . $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is the reward function and $\gamma \in (0, 1)$ is the discount factor. The policy $\pi(a|s)$ is a mapping from state space to action space, which generates action distributions a conditioned on observations s to maximize the expected return $\mathbb{E}_\pi[\sum_{t=0}^{T-1} \gamma^t R]$ in an episode with T time steps. To achieve this, the system requires sophisticated manipulation skills, including searching through surrounding objects in a cluttered environment to determine the target object’s location and efficiently removing obstructions to expose it. This task is significant as it enables robots to efficiently search and retrieve objects in complex cluttered scenarios, even when targets are completely obscured.

4. Method

In this section, we introduce our system for efficient object retrieval. The overview of the system is shown in Figure 2. Our framework consists of three parts: Task Construction (Section 4.1), RL Problem Design (Section 4.2) and the Policy Training (Section 4.3). The details of our sim-to-real policy transfer are introduced in Section 4.4.

4.1. Task Construction

The key challenge in cluttered scenes arises from the diversity of object configurations (e.g., categories, geometries, locations, and poses) and their combinations. To simulate realistic scenarios, we place 18 household objects with varying masses, sizes, and geometries in a box to create diverse cluttered scenes. At task initialization, objects are dropped into the box with the target object placed at the bottom. For each trial, we vary both the choice of target object and its pose within the box boundaries, ensuring diverse testing scenarios.

To avoid hand interference during scene initialization and enable reliable reward computation, we define two key static poses: the *prepare pose* and the *suspending pose*. Both poses position the hand above the box with a downward-facing palm. The *prepare pose* positions the hand directly above the box as the initial configuration. The *suspending pose* is specifically designed to keep the hand away from the box, ensuring it does not interfere with object dropping or reward computation. At initialization, we first move the hand to the *suspending pose* to allow object dropping and scene formation. The hand then returns

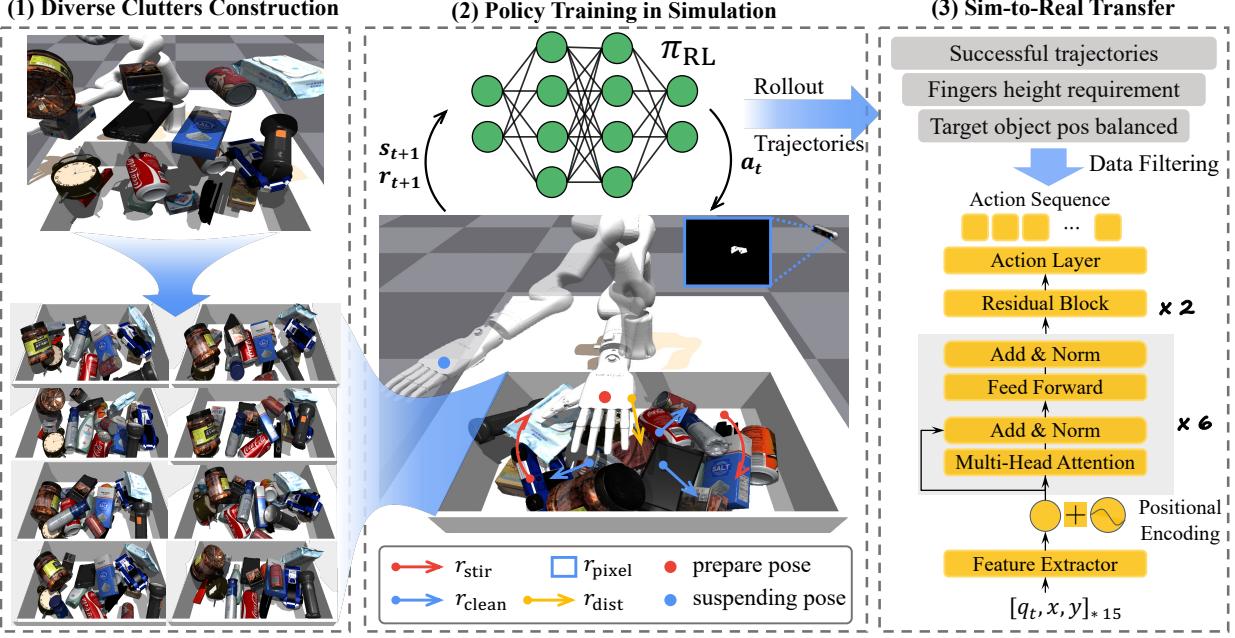


Figure 2 | Illustration of the Retrieval Skill System Design. (a) Constructs diverse cluttered scenes using a drop-from-above strategy. (b) Utilizes large-scale parallel RL with well-designed rewards to train policies. (c) Generates trajectories from the RL expert policy, selects useful ones based on our principle, and trains the distilled policy for deployment on a real robot.

to the *prepare pose* for policy training. During the training process, the robotic hand inevitably moves above the target object, interfering with our camera-based reward computation. Therefore, every 10 steps, we temporarily move the hand to the *suspending pose* for reward evaluation. We compute the reward by counting ground truth target object pixels from the top-down segmentation mask, then return the hand to its previous configuration. This strategy improves the stability of policy learning by assessing cumulative behavior over time rather than relying on noisy immediate feedback, providing a more stable training process. Notably, during policy evaluation, we no longer need to compute rewards and thus eliminate this periodic hand movement to the *suspending pose*. Detailed implementations of the Task Construction can be found in Appendix 7.1.

4.2. RL Problem Design

After establishing the cluttered scene, we address these challenging object retrieval tasks using model-free reinforcement learning. Below, we introduce the observation and action space of our policy, followed by the reward formulation.

Observation Space. At timestep t , the control policy observes a combination of proprioceptive and visual information. The proprioceptive inputs $q_t = (q_t^{\text{arm}}, q_t^{\text{hand}}) \in \mathbb{R}^{13}$ include the arm and hand joint positions $q_t^{\text{arm}} \in \mathbb{R}^7$ and $q_t^{\text{hand}} \in \mathbb{R}^6$, while the vi-

sual inputs consist of processed representations such as the bounding box coordinates of the target object’s segmentation $b_t = (x_t, y_t, w_t, h_t) \in \mathbb{R}^4$, its area $a_t = (w_t \cdot h_t) \in \mathbb{R}$, and the depth of the bounding box center pixel $d_t \in \mathbb{R}$. To facilitate more effective policy learning, we improve policy training by incorporating privileged information accessible in simulation. Specifically, the observation space is defined as:

$$s_t = \{q_t, b_t, a_t, \{T_t^{f,i}\}_{i=1}^5, T_t^{\text{obj}}, \dot{q}_t, v_t^{\text{obj}}, \{T_t^{\text{near}, i}\}_{i=1}^5\}. \quad (1)$$

These include the poses of five fingertips $\{T_t^{f,i}\}_{i=1}^5 \in \mathbb{R}^{35}$ and target object $T_t^{\text{obj}} \in \mathbb{R}^7$, the velocities of the current joints $\dot{q}_t \in \mathbb{R}^{13}$, and the target object’s linear and angular velocities $v_t^{\text{obj}} = (v_t^{\text{obj}}, \omega_t^{\text{obj}}) \in \mathbb{R}^6$. Additionally, the positions of the 5 nearest objects to the target $\{T_t^{\text{near}, i}\}_{i=1}^5 \in \mathbb{R}^{15}$ are included.

Action Space. The action space of our system is the target joint angles of our robot $a = (a_t^{\text{arm}}, a_t^{\text{hand}}) \in \mathbb{R}^{13}$. For better stable control, the policy generates a target joint position $a_t^{\text{hand}} \in \mathbb{R}^6$ for the hand and applies a linear smoothing update to blend it with the previous target, reducing abrupt movements. Specifically, the blending is computed as

$$a_t^{\text{hand}} = \lambda a_t^{\text{hand}} + (1 - \lambda) a_{t-1}^{\text{hand}}, \quad (2)$$

where λ is the smoothing factor. For the robotic arm, the action $a_t^{\text{arm}} \in \mathbb{R}^7$ represents relative joint position

changes, which are added to the current joint angles to obtain target positions for control.

Reward Function. We design a fine-grained reward function to optimize object retrieval skills and enable the hand to efficiently expose the target object. Specifically, the reward function comprises the following components: (1) *Distance Reward*. This reward encourages the hand to locate occluded areas by minimizing the distance between the hand’s palm and the target object. It is defined as $r_{\text{dist}} = \exp(-5 \cdot \min(d - e_0, 0))$, where d represents the distance between the hand’s palm and the target object, and $e_0 = 0.15$ is a predefined threshold. (2) *Stir Reward*. To encourage the hand to actively displace objects, especially in cases of complete occlusion. Let the positions of all objects in the clutters at timestep t be denoted as p_t^{all} . The stir reward is given by $r_{\text{stir}} = \alpha \|p_t^{\text{all}} - p_{t-1}^{\text{all}}\|_2$, where α is a scaling factor. (3) *Proximity Clearance Reward*. We define this reward to guide the agent in clearing occluding objects around the target object. Let the sum of distances between the target object and its k nearest objects be $\sum_{i=1}^k f_i$. The reward is formulated as $r_{\text{clean}} = \beta \cdot r_{\text{dist}} \cdot \sum_{i=1}^k f_i$, where β is a scaling hyperparameter. (4) *Pixel Emergency Reward*. We also design a vision-based holistic evaluation reward to encourage the hand to expose the target object. Denoting the count of pixels within the segmentation mask of the target object in the top-down camera frame as C , this reward is defined as $r_{\text{pixel}} = C/15$. (5) *Penalty*. To discourage undesirable behaviors, we introduce penalty terms as used in previous works [22, 30, 28], including action penalties, contact penalties, and penalties for displacing the target object.

To facilitate more efficient learning while mitigating the risk of reward hacking [39], we adopt the reward shaping technique proposed by Ng et al. [40]. Specifically, we define a state function:

$$\Phi(s) = r_{\text{dist}}(s) + r_{\text{clean}}(s) + r_{\text{pixel}}(s), \quad (3)$$

and formulate a potential-based shaping function as the final reward function:

$$\mathcal{R}(s, a, s') = \Phi(s') - \Phi(s). \quad (4)$$

4.3. Policy Training

We use PPO [41] to train a closed-loop policy for object retrieval with a dexterous robotic hand in cluttered environments. Thanks to the high-parallel simulation capabilities of IsaacGym [42], we parallelize our policy training across 512 environments simultaneously. To enhance the generalization capabilities of our policy, we apply domain randomization techniques, including scenario diversification and object pose variation. At each episode initialization, we randomly select the

box position and generate cluttered scenes by randomly dropping objects into the box while ensuring the target object remains covered, creating a broad spectrum of challenging environments. Additionally, we randomize the initial pose of the target object to further improve policy robustness. Detailed training configurations are provided in the Appendix 7.2.

4.4. Sim-to-Real Transfer

When deploying the policy in the real world, some observations, such as joint velocity and object velocity, cannot be accurately estimated. To address this, we use rollout trajectories from the trained RL expert policy. To improve effectiveness, we design a data selection principle, selecting successful trajectories where the fingers maintain a minimum height of 2 cm above the box bottom and the target object is evenly distributed within the box. Using the collected data, we distill a student policy suitable for real-world deployment through Behavior Cloning [43]. During distillation, the student policy receives an observation $o_t = (q_t^{\text{arm}}, x_t, y_t)$, where q_t^{arm} represents the arm’s joint position, and (x_t, y_t) denotes the position of the target object. While q_t^{arm} can be easily accessed, our system uses a side top-down camera to track the target object for real-time position acquisition. First, we use the RGB image from the camera to obtain the coordinates of the target object in the camera frame. Specifically, we use SAM [44] to obtain an initial binary mask of the target object as the input for Cutie [45] to continuously track the mask of the target object over time. By lining out the bounding box of the tracked mask, we compute its center point which represents the pixel coordinate of the target point and convert it to the coordinate in the camera frame via the intrinsic parameters of the camera. Then we do the hand-eye calibration to transform the coordinates in the camera frame into the world frame which is just the real-time position (x_t, y_t) of the target objects. The entire tracking system runs at the speed of 30Hz. The student policy then generates an action $a \in \mathbb{R}^{13}$, which corresponds to the target joint angles of the arm and hand. As shown in Figure 2, we employ a transformer network as the policy, which takes a sequence of observations, including the current observation and the nine history observations, and outputs the action to be taken at the current timestep. The transformer network has a powerful ability to effectively handle temporal dependencies in the sequential observation data, which is crucial for learning the long-term strategies required for retrieval tasks. Implementation details, including hyperparameter settings and network architecture are provided in Appendix 7.5.

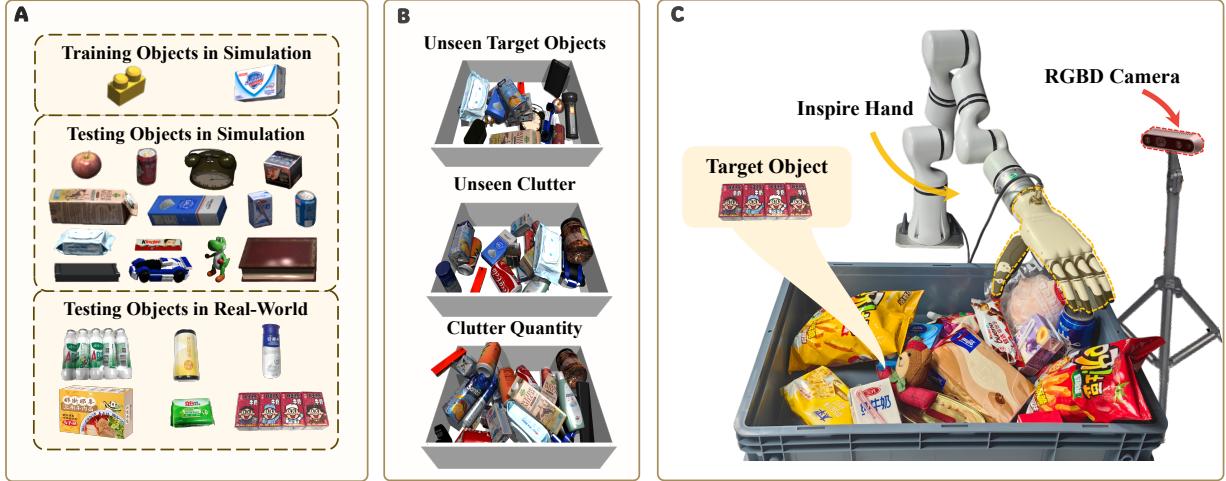


Figure 3 | Overview of the Experimental Setups. (A) Training object sets in simulation and testing object sets in both simulation and the real world. (B) Cluttered scenes in simulation. (C) Workspace of the real setup. We use an Inspired Hand mounted on a Realman RM-75 robot, equipped with a RealSense D435 camera.

5. Experiment

In this section, we evaluate our proposed framework through comprehensive experiments conducted in both simulation and real-world environments. Our investigation focuses on the following four research questions:

(1) How effective is our framework in performing object retrieval tasks? (2) How well does our framework generalize to objects with different geometries, masses, and cluttered patterns in object retrieval task? (3) How does our method achieve high efficiency in object retrieval tasks? (4) How well does our retrieval policy perform on real-world dexterous robotic systems?

Below, we first describe our experimental setup and dataset composition. We then present the evaluation metrics and baseline methods used for comparison. Finally, we systematically address each research question through detailed experimental results and analysis. All simulation results are averaged over 10 random seeds, while real-world performance metrics are derived from 10 independent trials per experiment.

5.1. Setups

Dataset. During policy optimization, we use a LEGO block and a soap box as target objects in the training dataset. At the beginning of each episode, one of these objects is randomly selected for policy training. To evaluate the generalizability of our policy to unseen objects, we supplement a test set comprising 8 small objects (e.g., apples) and 6 large objects (e.g., books), differentiated by shape and weight. Furthermore, we introduce several new objects to construct

novel stacked scenarios, testing the policy’s ability to generalize across various cluttered environments.

Evaluation Metrics. The primary goal of object retrieval is to locate the target object and maximize its visibility within the camera’s field of view to facilitate subsequent manipulations. We define *exposure* as the ratio of unobstructed pixels representing the target object on the imaging plane. Each episode is evaluated in two phases: (1) recording the target object’s visible pixel count, p_t^{curr} , and its 6D pose every 10 steps during retrieval; (2) removing all occluding objects in the simulation, resetting the target object to the recorded 6D pose, and recording its total visible pixel count, p_t^{all} . The exposure at time t is computed as $\text{exposure}_t = p_t^{\text{curr}} / p_t^{\text{all}}$. Detailed procedures are provided in Appendix 7.4. Retrieval is considered successful when the exposure exceeds 95%. To systematically evaluate both the performance and efficiency of the retrieval policy, we define the following metrics:

- **Success Rate (SR):** The percentage of trials where the target object achieves 95% exposure within 210 steps.
- **Retrieval Steps (RS):** The number of steps required to achieve successful retrieval.
- **Increase in Exposure Ratio (IER):** The absolute increase in the target object’s exposure from its initial to final state.

Baselines. We compare our method against the following baselines: (1) *Ours*. A policy is trained using PPO with a carefully designed reward function in diverse stacking environments. (2) *Ours w/o RS*. A policy is trained without reward shaping to assess the contribution of this technique to retrieval performance. (3)

Table 1 | Main Results for all methods.

Method	Seen			Unseen (Small)			Unseen (Large)		
	RSR	RS	IER	RSR	RS	IER	RSR	RS	IER
VMP	25.31 ± 5.85	192.39 ± 1.71	61.22 ± 1.36	32.29 ± 3.90	179.30 ± 6.38	60.45 ± 2.56	8.33 ± 3.21	204.55 ± 2.16	51.06 ± 0.90
Ours w/o RS	55.45 ± 1.61	149.45 ± 2.83	73.80 ± 2.99	46.71 ± 2.66	158.72 ± 10.81	69.32 ± 3.73	26.23 ± 4.83	179.51 ± 1.74	67.47 ± 1.00
Ours w/o r_{stir}	73.89 ± 1.45	132.48 ± 9.40	84.62 ± 1.32	69.27 ± 6.03	140.20 ± 4.87	79.72 ± 3.63	34.90 ± 7.25	174.37 ± 5.69	76.54 ± 1.06
Ours w/o r_{clean}	69.27 ± 3.32	126.23 ± 2.21	79.43 ± 1.27	63.56 ± 3.85	134.79 ± 4.60	72.89 ± 1.32	39.06 ± 5.85	167.91 ± 1.47	73.99 ± 1.76
Ours	84.23 ± 3.23	105.26 ± 3.79	89.85 ± 2.39	77.60 ± 1.47	127.85 ± 7.58	84.17 ± 2.19	62.25 ± 2.55	157.56 ± 6.60	84.08 ± 1.55

Ours w/o r_{stir} . A policy is trained without the stir reward to examine its role in policy learning. (4) *Ours w/o r_{clean}* . A policy is trained without the proximity clearance reward to analyze its significance in the overall reward structure. (5) *Visual-based Motion Planning Search (VMP)*. A heuristic motion planning baseline that uses segmentation masks of the target object to guide the robotic hand. Predefined rules are employed for retrieval manipulation.

Further details for baselines are provided in Appendix 7.3.

5.2. Results and Analysis

Main Results. We evaluate our method against various baselines in simulation with target objects of differing sizes. As shown in Table 1, our system consistently achieves higher retrieval success rates (RSR) and requires fewer retrieval steps (RS) compared to all baselines on both seen and unseen objects. Specifically, our method improves the success rate by 22.6% and 59.4% on small and large objects, demonstrating superior generalization capabilities. Across all methods, retrieving small target objects (e.g., a small LEGO block) is generally easier and more efficient than retrieving larger ones.

Ablation studies reveal the critical role of individual reward components. Removing the stir reward (r_{stir}) significantly reduces success rates for larger objects by 43.9%, indicating its importance in encouraging the robotic hand to stir objects within the clutter to effectively clear obstructions above large targets. In contrast, the proximity clearance reward (r_{clean}), which incentivizes the removal of objects near the target, proves more effective for smaller objects, increasing their retrieval efficiency by 22.1%. Furthermore, the potential-based reward function plays a pivotal role in enhancing both the performance and efficiency of our method, reducing retrieval steps by 19.4% and enhancing the success rate by 39.8%.

In comparison, the VMP relies on segmentation masks and predefined open-loop rules, suffers from lower success rates and efficiency. This highlights the advantages of our learned policy, which adapts dynamically to varying clutter configurations and object properties.

Impact of Occlusion Rate. We also explore the impact of clutter occlusion rate (i.e., 1 – exposure) on target objects. Figure 4 presents the relationship between occlusion rates, retrieval success rate (RSR), and retrieval steps (RS) in cluttered environments, comparing smaller and larger target objects. As the occlusion rate rises from 30% to 100%, RSR decreases for both object sizes, with smaller objects consistently achieving higher RSR than larger ones. This indicates that larger objects, although generally easier to detect, are more prone to being significantly obstructed by substantial clutter. In contrast, retrieval steps increase as occlusion rates rise, signifying reduced efficiency in highly cluttered settings. Smaller objects generally require fewer retrieval steps than larger ones across mostly occlusion levels, likely because retrieving larger objects involves removing more obstructing clutter. These findings reveal that smaller objects are retrieved with higher success and efficiency, while larger objects face greater challenges posed by occlusion.

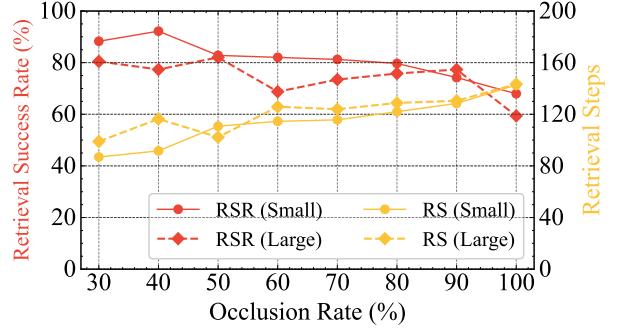


Figure 4 | Impact of Occlusion Rate on Performance and Efficiency. We evaluate the retrieval success rate and retrieval steps of our policy for small and large target objects under varying occlusion levels.

5.3. Generalization Capability

We investigate both in-distribution performance and generalization capabilities by comparing all methods across various generalization tasks. We consider three levels of skill generalization: unseen target object generalization (L1), unseen cluttered environment generalization (L2), and cluttered object quantity generalization (L3), as illustrated in Figure 3.B. For unseen

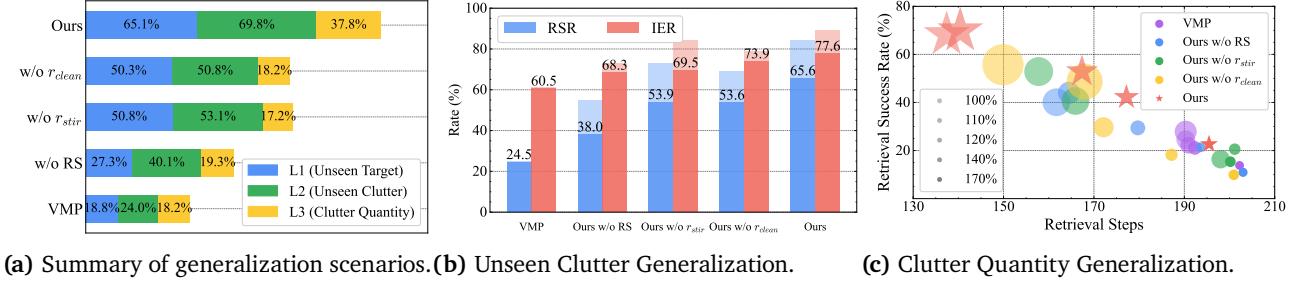


Figure 5 | Performance on Task Generalization. (a) depicts the average success rate across three levels of generalization. (b) illustrates performance on unseen clutter. (c) presents the impact of clutter quantity. Darker colors indicate a higher object count in the clutter, while larger shapes represent a greater average exposure increase (i.e., higher IER) during retrieval.

target object generalization (L1), only the target object is changed while keeping the clutter consistent. For unseen cluttered environment generalization (L2), 80% of the clutter objects are replaced, but the target object remains the same. For cluttered object quantity generalization (L3), 20% to 70% more clutter objects are added while keeping the target object unchanged. The success rate for these three generalization scenarios is summarized in Figure 5a, which highlights the superior generalization performance of our method. Notably, the reward shaping technique plays a critical role in our approach, achieving an average improvement of 100% across all generalization scenarios and an impressive 157% improvement in the unseen target object scenario.

As shown earlier in Table 1, we achieve exceptional target object generalization, both for small and large target objects. Furthermore, we observe that the number of objects in the clutter significantly influences retrieval manipulation performance, motivating the design of L3 generalization. As shown in Figure 5c, there is a noticeable performance gap between the different methods as the number of retrieval steps increases. Our system (denoted by red stars) demonstrates more stable performance, maintaining a higher success rate even with more retrieval steps, which highlights its robustness.

The comparison between different variants of Ours reveals that excluding certain components leads to decreased performance. Among these, excluding the RS technique causes the most substantial drop in success rate, underscoring the importance of this component for improving retrieval efficiency.

Regarding the data point size, it reflects the average increase in target object exposure during retrieval. We find that although some data points have similar sizes, there is a large gap in success rate and retrieval steps (e.g., the red star and yellow circle in the 100% scenario). This suggests that while exposure is increased

in many cases, it is only through consistently exposing the target object during the retrieval process that efficiency and success rate can be significantly improved.

5.4. Retrieval Efficiency

In this section, we evaluate the retrieval efficiency of our system through experiments that compare it to baseline methods. Retrieval efficiency is measured by the number of manipulations required for successful object retrieval.

Table 2 | The number of steps for successful retrieval.

Method	Small	Large	Complex
Ours	101.19 ± 2.06	125.61 ± 5.37	134.11 ± 5.56
VMP	163.43 ± 1.73	171.14 ± 5.13	191.93 ± 2.53
Grasp	1049.43 ± 9.73	1286.68 ± 15.73	1478.75 ± 12.60

We consider three scenarios: small target objects, large target objects, and complex clutter (defined as scenarios with 40% more clutter than typical cases). Each scenario involves retrieving a target object that is 90% occluded, a challenging level of occlusion chosen to test the limits of retrieval strategies. The goal is to assess how efficiently our dexterous multi-finger hand, capable of performing skill-based actions such as pushing, stirring, and poking, can retrieve the target object compared to traditional object removal methods.

This experiment includes two baseline methods: (1) *Visual-based Motion Planning*. The system uses the object’s segmentation mask to compute its position and employs pre-defined finger actions to displace objects in the clutter to expose the target object. (2) *Grasp-and-Pick*. This method assumes the stacking relationships of occluding objects are known and attempts to grasp and lift these objects one by one to reveal the target. For this experiment, we assume the successful grasp and removal of each occluding object.

The results, summarized in Table 2 demonstrate that



Figure 6 | Retrieval Sequence in Real-World Clutters. We present four everyday objects as target items, varying in shape and size.

our method consistently outperforms the baseline approaches in terms of retrieval efficiency. Compared to VMP, our method reduces the number of steps by an average of 38% across all scenarios. When compared to Grasp-and-Pick, our method shows an even greater reduction in steps, averaging a 90% reduction. This efficiency is primarily attributed to the multi-finger hand’s ability to directly interact with and displace occluding objects, rather than removing them sequentially as in the baseline methods.

5.5. Real-World Experiment

We conduct sim-to-real experiments (Figure 6) to evaluate the performance of our method and two baseline methods on a real-robot platform shown in Figure 3.C. Our objective is to address the following key questions regarding the performance of our system:

- Can the policies learned in simulation zero-shot transfer to a real-world dexterous multi-fingered robotic system?
- Can the distilled policy successfully retrieve target objects that generalize to different positions?
- Can our system achieve more efficiency in multi-object stacked environments compared to existing methods?

Retrieve Various Target Objects. We present quantitative results comparing our policy to baseline policies in Table 3. Specifically, we evaluate small and large daily objects as target objects across three shapes: cuboids, cylinders, and spheres. Our method demonstrates consistent and stable retrieval performance across all target objects. Notably, it achieves higher success rates for smaller, cylindrical, and spherical objects. In contrast, the visual-based motion planning method suffers from limited flexibility due to its predefined hand manipulations, often causing previously removed occluding objects to re-block the target, thereby reducing retrieval success rates.

Table 3 | Performance in Real-World Settings.

Settings	Cuboid				Cylinder		Sphere
	Milk1	Milk2	Noodles	Soap	Yogurt	Trash Bag	Ball
Ours	6/10	8/10	7/10	9/10	7/10	9/10	8/10
VMP	2/10	3/10	0/10	1/10	1/10	4/10	4/10

Target Object Position Generalization. We also investigate the impact of target object positions on retrieval performance. The target objects are placed in five distinct regions of the box: center, top-left, bottom-left, top-right, and bottom-right. The experimental

Table 4 | Success rates for various object positions.

Position	Center	Top Left	Bottom Left	Top Right	Bottom Right
Ours	8/10	6/10	7/10	5/10	6/10
VMP	2/10	1/10	3/10	0/10	2/10

results are summarized in Table 4. We observe that positions near the center and bottom (closer to the robotic arm) achieve better generalization. In contrast, positions farther away such as top-left and top-right are constrained by the robot’s workspace and reduce manipulation flexibility.

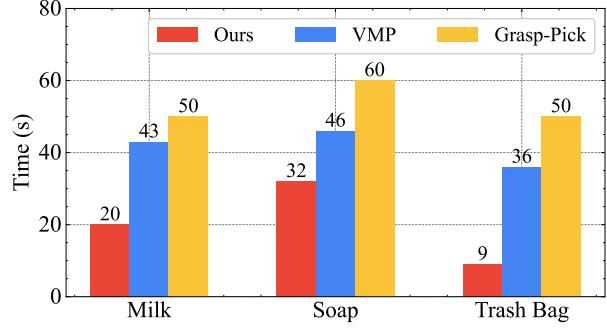
Retrieval Efficiency. Retrieval efficiency is the most important thing which is measured by the time required to retrieve the target object. We compare our system against baseline methods, including VMP and sequential grasping and removal. As shown in Figure 7, our policy effectively clears occluding objects, facilitating the exposure of the target. Specifically, our approach reduces retrieval time by 51.2% compared to VMP and by 61.9% compared to sequential grasping and removal on average. These results highlight the advantage of leveraging skill-based interactions, such as pushing and stirring, over sequential removal strategies.

6. Limitation

The primary limitation of our work lies in the requirement for object mask inputs to the policy in real-world scenarios, necessitating human intervention. Future work will explore leveraging pretrained vision-language models and foundational vision models to automatically generate the required masks.

7. Conclusion

In this work, we have presented a novel approach to efficient object retrieval in cluttered environments using dexterous multi-finger hands. Our system demonstrates the ability to manipulate occluding objects strategically, exposing target objects for retrieval—a capability that significantly improves upon traditional sequential removal methods. Through careful design of our simulation environment and reinforcement learning framework, we have addressed key challenges including time efficiency, object diversity, and the complexity of high-dimensional control in contact-rich environments. Our experimental results, both in simulation and real-world settings, validate the effectiveness of our approach. The system successfully generalizes across diverse objects and achieves zero-shot transfer to real-world robots, demonstrating robust performance without additional training. This work represents a step toward more efficient and ca-

**Figure 7 |** Retrieval time for various target objects.

pable robotic manipulation in cluttered environments, though opportunities remain for future exploration, particularly in achieving fully autonomous operation through integration with advanced perception systems.

References

- [1] Yunfei Bai and C. Karen Liu. Dexterous manipulation using both palm and fingers. In *International Conference on Robotics and Automation (ICRA)*, pages 1560–1565, 2014.
- [2] S. Gruber. Robot hands and the mechanics of manipulation. *IEEE Journal on Robotics and Automation*, 2(1):59–59, 1986.
- [3] Adithyavairavan Murali, Arsalan Mousavian, Clemens Eppner, Chris Paxton, and Dieter Fox. 6-dof grasping for target-driven object manipulation in clutter. In *International Conference on Robotics and Automation (ICRA)*, pages 6232–6238, 2020.
- [4] Jing-Cheng Pang, Si-Hang Yang, Xiong-Hui Chen, Xinyu Yang, Yang Yu, Mas Ma, Ziqi Guo, Howard Yang, and Bill Huang. Object-oriented option framework for robotics manipulation in clutter. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1230–1237, 2023.
- [5] Yitong Li, Ruihai Wu, Haoran Lu, Chuanruo Ning, Yan Shen, Guanqi Zhan, and Hao Dong. Broadcasting support relations recursively from local dynamics for object retrieval in clutters. In *Robotics: Science and Systems (RSS)*, 2024.
- [6] Xinyuan Zhao, Wenyu Liang, Xiaoshi Zhang, Chee Meng Chew, and Yan Wu. Unknown object retrieval in confined space through reinforcement learning with tactile exploration. In *International Conference on Robotics and Automation (ICRA)*, pages 10881–10887, 2024.

- [7] Ankit Goyal, Arsalan Mousavian, Chris Paxton, Yu-Wei Chao, Brian Okorn, Jia Deng, and Dieter Fox. Ifor: Iterative flow minimization for robotic object rearrangement. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14787–14797, 2022.
- [8] Bingjie Tang and Gaurav S. Sukhatme. Selective object rearrangement in clutter. In *Conference on Robot Learning (CoRL)*, volume 205, pages 1001–1010, 2023.
- [9] Yinsen Jia and Boyuan Chen. Cluttergen: A cluttered scene generator for robot learning. In *Conference on Robot Learning (CoRL)*, 2024.
- [10] Hamidreza Kasaei and Mohammadreza Kasaei. Harnessing the synergy between pushing, grasping, and throwing to enhance object manipulation in cluttered scenarios. *arXiv preprint arXiv:2402.16045*, 2024.
- [11] Baichuan Huang, Shuai D. Han, Jingjin Yu, and Abdeslam Boularias. Visual foresight trees for object retrieval from clutter with nonprehensile rearrangement. *IEEE Robotics and Automation Letters*, 7(1):231–238, 2022.
- [12] Andrey Kurenkov, Joseph Taglic, Rohun Kulakarni, Marcus Dominguez-Kuhne, Animesh Garg, Roberto Martín-Martín, and Silvio Savarese. Visuomotor mechanical search: Learning to retrieve target objects in clutter. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 8408–8414, 2020.
- [13] Yuchen Xiao, Sammie Katt, Andreas ten Pas, Shengjian Chen, and Christopher Amato. Online planning for target object search in clutter under partial observability. In *International Conference on Robotics and Automation (ICRA)*, pages 8241–8247, 2019.
- [14] Rafael Papallas and Mehmet R. Dogar. Non-prehensile manipulation in clutter with human-in-the-loop. In *International Conference on Robotics and Automation (ICRA)*, pages 6723–6729, 2020.
- [15] Kechun Xu, Hongxiang Yu, Qianen Lai, Yue Wang, and Rong Xiong. Efficient learning of goal-oriented push-grasping synergy in clutter. *IEEE Robotics and Automation Letters*, 6(4):6337–6344, 2021.
- [16] Michael Danielczuk, Andrey Kurenkov, Ashwin Balakrishna, Matthew Matl, David Wang, Roberto Martín-Martín, Animesh Garg, Silvio Savarese, and Ken Goldberg. Mechanical search: Multi-step retrieval of a target object occluded by clutter. In *International Conference on Robotics and Automation (ICRA)*, pages 1614–1621, 2019.
- [17] Jingxi Xu, Yinsen Jia, Dongxiao Yang, Patrick Meng, Xinyue Zhu, Zihan Guo, Shuran Song, and Matei Ciocarlie. Tactile-based object retrieval from granular media. *arXiv preprint arXiv:2402.04536*, 2024.
- [18] Oliver Lemke, Zuria Bauer, René Zurbrügg, Marc Pollefeys, Francis Engelmann, and Hermann Blum. Spot-compose: A framework for open-vocabulary object retrieval and drawer manipulation in point clouds. In *Workshop on Mobile Manipulation and Embodied Intelligence at ICRA 2024*, 2024.
- [19] Kechun Xu, Shuqi Zhao, Zhongxiang Zhou, Zizhang Li, Huajin Pi, Yifeng Zhu, Yue Wang, and Rong Xiong. A joint modeling of vision-language-action for target-oriented grasping in clutter. In *International Conference on Robotics and Automation (ICRA)*, pages 11597–11604, 2023.
- [20] Wissam Bejjani, Wisdom C. Agboh, Mehmet R. Dogar, and Matteo Leonetti. Occlusion-aware search for object retrieval in clutter. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 4678–4685, 2021.
- [21] Baichuan Huang, Teng Guo, Abdeslam Boularias, and Jingjin Yu. Interleaving monte carlo tree search and self-supervised learning for object retrieval in clutter. In *International Conference on Robotics and Automation (ICRA)*, pages 625–632, 2022.
- [22] Yuanpei Chen, Chen Wang, Li Fei-Fei, and Karen Liu. Sequential dexterity: Chaining dexterous policies for long-horizon manipulation. In *Conference on Robot Learning (CoRL)*, volume 229, pages 3809–3829, 2023.
- [23] Vikash Kumar, Yuval Tassa, Tom Erez, and Emanuel Todorov. Real-time behaviour synthesis for dynamic hand-manipulation. In *International Conference on Robotics and Automation (ICRA)*, pages 6808–6815, 2014.
- [24] Priyanka Mandikal and Kristen Grauman. Learning dexterous grasping with object-centric visual affordances. In *International Conference on Robotics and Automation (ICRA)*, pages 6169–6176, 2021.

- [25] Qiuyu Chen, Karl Van Wyk, Yu-Wei Chao, Wei Yang, Arsalan Mousavian, Abhishek Gupta, and Dieter Fox. Learning robust real-world dexterous grasping policies via implicit shape augmentation. In *Conference on Robot Learning (CoRL)*, volume 205, pages 1222–1232, 2023.
- [26] Yuanpei Chen, Chen Wang, Yaodong Yang, and Karen Liu. Object-centric dexterous manipulation from human motion data. In *8th Annual Conference on Robot Learning*, 2024.
- [27] Max Yang, chenghua lu, Alex Church, Yijiong Lin, Christopher J. Ford, Haoran Li, Efi Psomopoulou, David A.W. Barton, and Nathan F. Lepora. Anyrotate: Gravity-invariant in-hand object rotation with sim-to-real touch. In *Conference on Robot Learning (CoRL)*, 2024.
- [28] Toru Lin, Zhao-Heng Yin, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Twisting lids off with two hands. In *Conference on Robot Learning (CoRL)*, 2024.
- [29] Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. In *Conference on Robot Learning (CoRL)*, volume 164, pages 297–307. PMLR, 08–11 Nov 2022.
- [30] Binghao Huang, Yuanpei Chen, Tianyu Wang, Yuzhe Qin, Yaodong Yang, Nikolay Atanasov, and Xiaolong Wang. Dynamic handover: Throw and catch with bimanual hands. In *Conference on Robot Learning (CoRL)*, volume 229, pages 1887–1902, 06–09 Nov 2023.
- [31] 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.
- [32] Yuanpei Chen, Yiran Geng, Fangwei Zhong, Jiaming Ji, Jiechuang Jiang, Zongqing Lu, Hao Dong, and Yaodong Yang. Bi-dexhands: Towards human-level bimanual dexterous manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):2804–2818, 2024.
- [33] Runze Liu, Chenjia Bai, Jiafei Lyu, Shengjie Sun, Yali Du, and Xiu Li. Vlp: Vision-language preference learning for embodied manipulation. *arXiv preprint arXiv:2502.11918*, 2025.
- [34] Shengjie Sun, Runze Liu, Jiafei Lyu, Jing-Wen Yang, Liangpeng Zhang, and Xiu Li. A large language model-driven reward design framework via dynamic feedback for reinforcement learning. *arXiv preprint arXiv:2410.14660*, 2024.
- [35] Fengshuo Bai, Hongming Zhang, Tianyang Tao, Zhiheng Wu, Yanna Wang, and Bo Xu. Picor: Multi-task deep reinforcement learning with policy correction. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 37(6):6728–6736, Jun. 2023.
- [36] Hongming Zhang, Fengshuo Bai, Chenjun Xiao, Chao Gao, Bo Xu, and Martin Müller. β -dqn: Improving deep q-learning by evolving the behavior. *arXiv preprint arXiv:2501.00913*, 2025.
- [37] Wenzuan Zhou and David Held. Learning to grasp the ungraspable with emergent extrinsic dexterity. In *Conference on Robot Learning (CoRL)*, 2022.
- [38] Ananye Agarwal, Shagun Uppal, Kenneth Shaw, and Deepak Pathak. Dexterous functional grasping. In *Conference on Robot Learning (CoRL)*, volume 229, pages 3453–3467, 06–09 Nov 2023.
- [39] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.
- [40] Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning (ICML)*, page 278–287, 1999.
- [41] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [42] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu based physics simulation for robot learning. In *Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS)*, volume 1, 2021.
- [43] Dean A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 1. Morgan-Kaufmann, 1988.
- [44] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023.

- [45] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3151–3161, 2024.
- [46] John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*, 2016.
- [47] Martin Skrodzki. The kd tree data structure and a proof for neighborhood computation in expected logarithmic time. *arXiv preprint arXiv:1903.04936*, 2019.

Supplementary Material

Implementation Details

7.1. Task Construction

Domain Randomization. To enhance the robustness and generalization capability of our system, we implement comprehensive domain randomization strategies during the environment reset phase. The randomization encompasses multiple aspects of the environment:

- Object Mass Randomization: At the beginning of each episode, object masses are randomized by scaling each object’s default mass with a random factor sampled from a uniform distribution $U(1, 1.5)$ (units: kg):

$$m_{\text{curr}} = m_{\text{default}} \cdot \alpha, \quad \alpha \sim U(1, 1.5)$$

- Object Position Randomization: Small perturbations are applied to the initial positions of objects to introduce variability (units: meters):

$$\begin{aligned} \Delta x &\sim U(-0.02, 0.02) \\ \Delta y &\sim U(-0.02, 0.02) \end{aligned}$$

- Target Position Randomization: The initial position of the target object within the box is randomized. The random displacements are sampled from (units: meters):

$$\begin{aligned} \Delta x &\sim U(-0.15, 0.15) \\ \Delta y &\sim U(-0.2, 0.2) \end{aligned}$$

This ensures that the target object can be placed within 70% of the box’s area.

- Camera Mount Randomization: During data collection, the camera’s mounting position is perturbed with small random displacements (units: meters):

$$\mathbf{p}_{\text{camera}} = \mathbf{p}_{\text{default}} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim U(-0.01, 0.01)^3$$

7.2. RL Training

We employ the Proximal Policy Optimization (PPO) algorithm [41] to train a continuous control policy using an actor-critic architecture. Detailed hyperparameters are provided in Table 5. The policy network is parameterized as a multi-layer perceptron (MLP) with three layers of sizes [1024, 512, 256], utilizing the ELU activation function for improved gradient flow and non-linearity. The standard deviation of the policy distribution is learned via a log-std representation, enabling dynamic adjustment of exploration during training.

To ensure stable and efficient learning, we adopt adaptive learning rate scheduling, starting at 3×10^{-4} . The

advantage function is normalized to reduce variance in policy gradient updates, while the Generalized Advantage Estimation (GAE) [46] parameter τ is set to 0.95, striking a balance between bias and variance. Gradient clipping with a norm threshold of 1 is applied to prevent exploding gradients. To constrain policy updates, we employ a PPO clipping range of 0.1, which limits large deviations from the current policy, and enforce a KL divergence threshold of 0.02 to promote conservative updates and prevent policy collapse.

The training runs for a maximum of 50,000 epochs, with model checkpoints saved every 1,000 epochs. The best-performing model is selected based on validation returns and retained after 200 epochs to prevent overfitting. A separate centralized value function is used for advantage estimation, parameterized as an MLP with the same architecture as the policy network. The critic network employs a higher learning rate of 1×10^{-3} to facilitate faster convergence in value estimation, a choice informed by preliminary experiments indicating more stable critic updates with this configuration.

7.3. Baseline Implementation

In our simulation experiments, we compare our method against five baseline approaches. Three of these baselines—*Ours w/o RS*, *Ours w/o r_{stir}* , and *Ours w/o r_{clean}* —are derived by removing specific components from our proposed method. The other two baselines are Visual-based Motion Planning Search (VMP) and Grasp-Pick. VMP is a heuristic motion planning approach that uses target object segmentation masks to guide the robotic hand toward the target object and employs predefined rules for retrieval manipulation. Grasp-Pick involves sequentially grasping and placing objects based on the support relationships within the cluttered scene.

VMP. The VMP system implements a vision-guided manipulation framework for dexterous robotic retrieval tasks in cluttered environments. It integrates visual perception, motion planning, and control execution through a state machine architecture to ensure reliable object manipulation.

The vision module employs a top-down camera with a resolution of 1024×512 , capturing RGB, depth, and segmentation maps of the workspace. Target objects are identified using segmentation masks obtained from the segmentation map, with their IDs corresponding to known object labels. The center of the target mask’s bounding box is extracted as the 2D image coordinate, which is projected into 3D space using depth data to obtain precise object localization. For motion planning, the robotic arm moves its end effector to

Table 5 | Hyperparameters for PPO Training.

Category	Parameter	Value	Description
<i>Model Architecture</i>	MLP Layers	[1024, 512, 256]	Number of neurons per layer
	Activation Function	ELU	Non-linearity used in the network
<i>Training Parameters</i>	Learning Rate	3×10^{-4}	Step size for policy update
	Discount Factor (γ)	0.99	Reward discounting factor
	GAE Parameter (τ)	0.95	Smoothing factor for GAE
	Entropy Coefficient	0	Weight of entropy regularization
	Gradient Clipping	Norm 1	Prevents gradient explosion
	Clip Range (ϵ)	0.1	PPO clipping threshold
	KL Threshold	0.02	KL divergence threshold for stopping training
	Minibatch Size	512	Batch size for optimization
	Mini Epochs	5	Number of updates per batch
	Horizon Length	8	Number of steps before update
	Max Training Epochs	50,000	Maximum number of training iterations
	Value Learning Rate	1×10^{-3}	Learning rate for value function

the computed 3D coordinate and performs a scrape action to retrieve the object.

When the target object is completely occluded and its segmentation mask cannot be detected, the system employs an exploration strategy by randomly sampling four 3D coordinates within the cluttered bin area. The arm sequentially moves to these coordinates, performing scrape actions to uncover the target object.

Specifically, the entire motion planning and scrape action process employs a four-stage approach to ensure reliable object retrieval.

1. Pre-approach stage: The end-effector moves to a predefined position ($h = 0.5$ m) above the target object. This configuration facilitates subsequent control of the hand to reach any position within the bounding box.
2. Final approach stage: Precise positioning is achieved using visual feedback combined with damped least squares inverse kinematics:

$$\tau = J^T (JJ^T + \lambda I)^{-1} \Delta x$$

where $\lambda = 0.05$ is the damping parameter, J is the Jacobian matrix, and Δx represents the positional error.

3. Scraping stage: The system executes a periodic motion pattern defined by:

$$x(t) = A \sin(2\pi ft + \phi) + O$$

where the amplitude $A = 2.0$, frequency $f = 20$ Hz, phase shift $\phi = \pi/4$, and offset $O = 0.5$.

4. Reset stage: The target object has been retrieved, so the robot arm will return the end effector to its initial position.

The control execution module utilizes position-based control for both arm and finger joints. Adaptive damping parameters are applied to ensure stable motion, while joint limits are strictly enforced throughout the execution process: $q_{\min} \leq q \leq q_{\max}$.

Grasp-Pick. This method relies on support relationships among cluttered objects to guide the grasping sequence. To ensure these assumptions hold, we designed tailored setups for both simulation and real-world experiments.

In simulation, object positions are directly accessible, enabling precise calculation of support relationships. We employ a KD-Tree [47] to organize the coordinates of objects near the target. Based on Euclidean distance, we select the three to five nearest objects, depending on the scenario, and manipulate them sequentially to clear access to the target. While this approach offers computational simplicity, it assumes ideal sensing conditions and may not generalize to more complex spatial arrangements.

For robotic control, we implement damped least squares inverse kinematics:

$$\dot{\mathbf{q}} = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T + \lambda \mathbf{I})^{-1} \dot{\mathbf{x}},$$

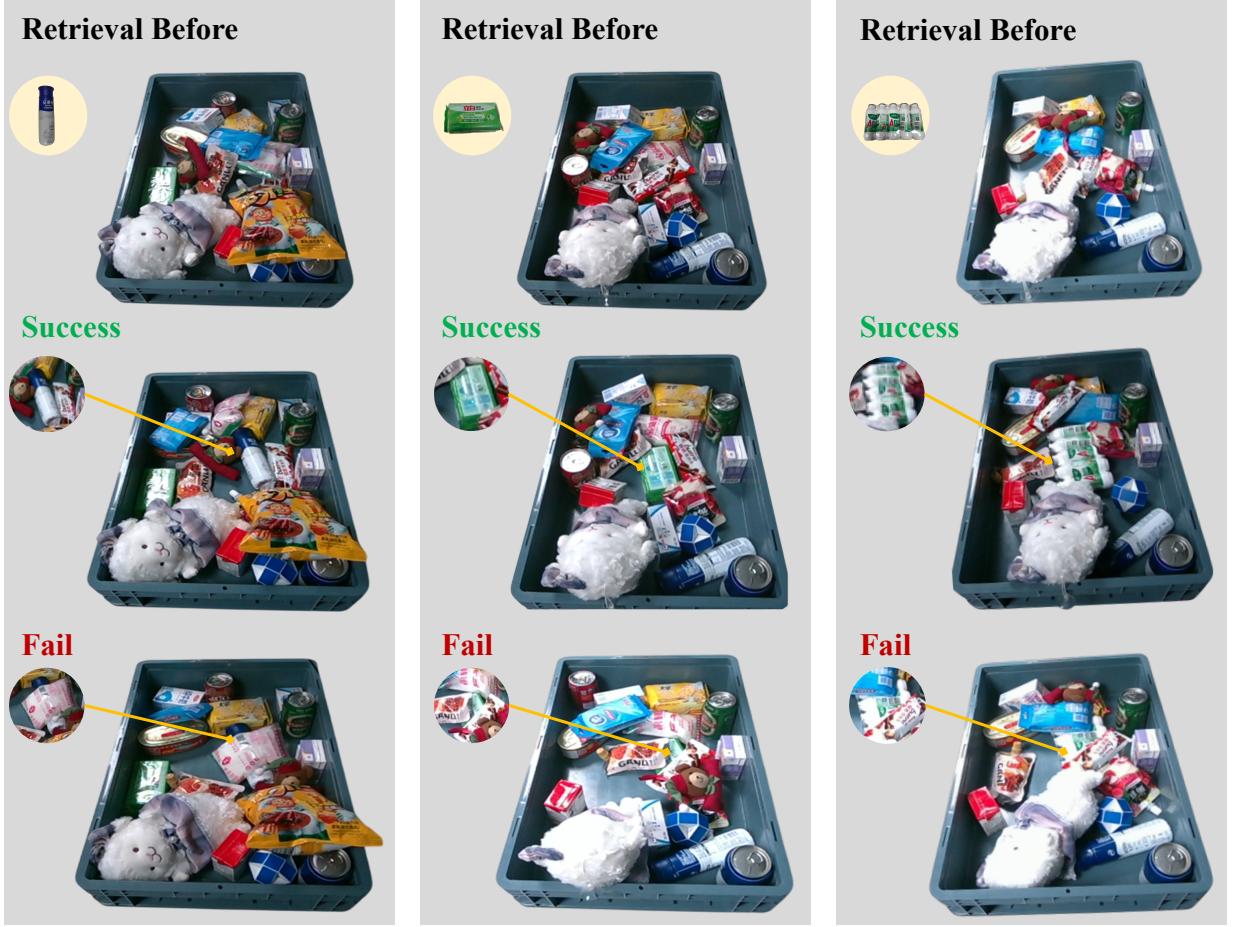


Figure 8 | Examples of successful and failed object retrievals on the real robot.

where λ is the damping coefficient, \mathbf{J} is the Jacobian matrix, and $\dot{\mathbf{x}}$ is the desired end-effector velocity. This formulation offers stable solutions near singularities but may limit the dexterity needed in cluttered environments.

In real-world experiments, sequentially grasping and removing multiple objects in stacked scenes with a dexterous hand remains challenging due to perception and control limitations. To address this, we employ predefined trajectories for each trial, simulating an idealized execution scenario. While this implementation provides an upper-bound estimate of this method's efficiency, it does not reflect the challenges of autonomous execution in unstructured environments.

7.4. Evaluation metric

Exposure Calculation The primary goal of object retrieval is to locate the target object and enhance its visibility within the camera's field of view, facilitating subsequent manipulations. We define *exposure* as the proportion of unobstructed pixels of the target object in the imaging plane. Considering that changes in the object's pose can affect the number of visible pixels,

we proceed as follows:

At timestep t , we record the target object's visible pixels p_t^{curr} and its 6D pose. Subsequently, all objects except the target are removed, the target object's recorded 6D pose is reset, and its total visible pixels are recorded as p_t^{all} . The exposure at time t is then computed as:

$$\text{exposure}_t = \frac{p_t^{\text{curr}}}{p_t^{\text{all}}}. \quad (5)$$

Success in Real-World Experiments. To systematically evaluate the success rate of object retrieval on the real robot, we capture images before and after each task using a side-mounted RealSense D435 camera. The success of the retrieval is determined by comparing the exposure of the target object in these images. As illustrated in Figure 8, we present examples of both successful and failed retrieval attempts.

7.5. Sim-to-Real transfer

For real-world deployment, we collect a set of trajectories generated by our RL expert policy. To ensure

Table 6 | Hyperparameters of Distilled Policy.

Category	Parameter	Value	Description
Model Architecture	Input State Dimension	9	Size of input state vector
	Action Dimension	13	Number of output actions
	History Frames	15	Past frames used as input
	Future Action Frames	5	Future actions predicted
	Transformer Hidden Size (d_{model})	384	Hidden layer size
	Number of Attention Heads	6	Transformer attention heads
	Number of Transformer Layers	6	Transformer depth
Training Parameters	Feed-forward Dimension	2048	FFN hidden size
	Dropout Rate	0.15	Dropout probability
	Batch Size	512	Training batch size
	Total Iterations	10,000	Training iterations
	Learning Rate	1e-4	Initial learning rate
	Optimizer	Adam	Optimization algorithm
	Loss Function	Negative Log Product	Loss function used
	Gradient Clip Norm	1.0	Gradient clipping threshold

data quality and consistency, we first select successful trajectories. We then filter out trajectories where the finger's z -coordinate lower 2,cm above the box, a threshold empirically chosen to prevent unstable behavior and reduce the risk of collision with the box during manipulation. To promote generalization, we balance the dataset across various target object positions within the box, ensuring uniform coverage of spatial configurations. This prevents the model from overfitting to specific object placements and enhances its adaptability to unseen scenarios.

The model architecture consists of a state encoder followed by a multi-head self-attention mechanism with six transformer layers, each containing six attention heads. This design captures complex temporal dependencies across historical state sequences of length 15, enabling accurate prediction of future actions over a five-step horizon. The hidden dimension of 384, paired with a feed-forward expansion ratio of 5.33 (2048/384), strikes a balance between model expressiveness and computational efficiency.

To effectively manage the continuous action space inherent in robotic control, we introduce a custom Negative Log Product Loss function, which penalizes trajectory deviations more sensitively than traditional mean squared error. This loss function emphasizes multi-step consistency, enhancing the model's predictive stability. Training is performed using the Adam optimizer with a learning rate of 1×10^{-4} over 10,000 iterations and a batch size of 512. Mixed-precision

training accelerates computation without compromising accuracy, while gradient clipping at 1.0 maintains stable learning dynamics. Hyperparameter selection was guided by cross-validation on a held-out dataset to optimize both performance and robustness. Detailed architectural specifications and hyperparameters are provided in Table 6. Despite strong simulation performance, real-world deployment introduces challenges such as sensor noise, domain discrepancies, and dynamic environmental conditions. Our transformer model mitigates these issues by leveraging temporal patterns to predict smooth and consistent actions.