Learning Long-Horizon Robot Manipulation Skills via Privileged Action

Xiaofeng Mao[1], Yucheng XU[1], Zhaole Sun[1], Elle Miller[1], Daniel Layeghi[1], Michael Mistry[1]
[1]School of Informatics, The University of Edinburgh, UK

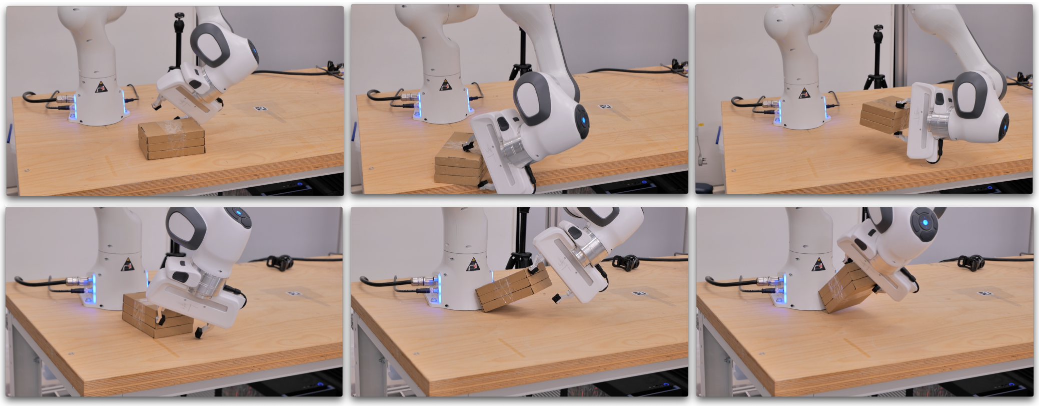


Figure 1: Long-horizon manipulation incorporates distinct non-prehensile skills: push-and-grasp (upper) and pivot grasp (blow). These skills are learned across diverse environmental settings with our proposed framework while employing the same reward function without any task-specific reward for non-prehensile manipulation. The videos are available at: https://youtu.be/4E74kLgeVas.

Abstract: Long-horizon contact-rich tasks are challenging to learn with reinforcement learning, due to ineffective exploration of high-dimensional state spaces with sparse rewards. The learning process often gets stuck in local optimum and demands task-specific reward fine-tuning for complex scenarios. In this work, we propose a structured framework that leverages privileged actions with curriculum learning, enabling the policy to efficiently acquire long-horizon skills without relying on extensive reward engineering or reference trajectories. Specifically, we use privileged actions in simulation with a general training procedure that would be infeasible to implement in real-world scenarios. These privileges include relaxed constraints and virtual forces that enhance interaction and exploration with objects. Our results successfully achieve complex multi-stage long-horizon tasks that naturally combine non-prehensile manipulation with grasping to lift objects from non-graspable poses. We demonstrate generality by maintaining a parsimonious reward structure and showing convergence to diverse and robust behaviors across various environments. Our approach outperforms state-of-the-art methods in these tasks, converging to solutions where others fail.

Keywords: Reinforcement Learning, Curriculum Learning, Robot Manipulation

1 Introduction

Training robots for long-horizon, contact-rich manipulation tasks from scratch using reinforcement learning (RL) remains a substantial challenge. While RL has excelled in learning various complex locomotion and manipulation tasks [1, 2, 3], most research focuses on optimizing specific short-horizon behaviors. The usual challenges associated with exploration compound dramatically with longer time horizons. If long-horizon sequences are considered, it is common to stitch multiple policies together [4, 5, 6]. Overall this approach is undesirable, as we want the robot to autonomously discover optimal behaviors without handcrafting and combining primitives.

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

Exploration is yet again more challenging in contact-rich tasks [7, 8]. Imagine learning to grasp a cube floating in mid-air versus on a surface. The set of state-action pairs that leads to grasping in mid-air is significantly larger compared to on a surface. Physical collision boundaries can greatly impede exploration in high-reward regions, potentially trapping the learning process in local optimum. Furthermore, most contact-rich tasks involve learning to manipulate and change the state of external objects. The additional non-linear dynamics in these scenarios poses a greater challenge for learning. To solve the exploration problem in long-horizon manipulation, recent work has explored using reference trajectories to provide a warm start and minimize unnecessary exploration [9, 10, 11]. However, training robots from scratch remains a valuable area of research, especially when robots must operate in diverse environments where behavior varies significantly. Collecting demonstrations for each unique environment is labor-intensive and impractical at scale. Furthermore, limitations in teleoperation devices and mapping accuracy reduce the effectiveness of human teleoperation for teaching manipulation skills. When environmental parameters vary significantly the optimal policy can change substantially [12, 13], and human-provided demonstrations may introduce biases that are misaligned with the optimal behavior of the robot due to differences in human and robotic capabilities. Training robots in simulation offers interesting opportunities beyond what is feasible in the real-world. For example, simulations can easily provide privileged information, which has been shown to significantly improve sample efficiency and policy performance by providing useful features to learn from [14, 15, 16]. Despite this, privileged information is not guaranteed to help the robot discover innovative behaviors. We propose the novel concept of *privileged actions*: actions that are infeasible in the real world, but enable efficient policy exploration. This includes relaxing constraints by disabling collisions and applying virtual forces to simplify interactions. We are motivated to solve long-horizon contact-rich tasks with RL, without requiring carefully tuned reward shaping. We introduce a framework that enhances exploration via privileged action, coupled with a learning curriculum to gradually align any physically infeasible training with real-world settings. The main contributions are summarised as follows:

- **Novel concept of privileged action**: We propose to leverage privileged actions that are not feasible in the real world to simplify the problem and improve training efficiency.
- **General framework combing privileged action with curriculum learning**: We build a general framework that enables the policy to efficiently solve long-horizon, contact-rich manipulation tasks while enforcing real-world constraints through a learning curriculum.
- **Robust Policy Adaptation**: Experiments demonstrate that the method adapts to environmental changes and converges to new behaviors without requiring reward modifications and with no reward indication on non-prehensile manipulation. Our approach outperforms state-of-the-art methods under identical setups, excelling across various tasks.

2 Related Works

Robot Learning of Long-horizon Tasks. Learning robotic policies for long-horizon manipulation tasks has been a longstanding and complex challenge. Many works focused on leveraging human prior knowledge to simplify this problem [17, 18, 19, 20]. Imitation learning is commonly used to simplify the complexities of long-horizon manipulation tasks. By breaking down long-horizon skills into sub-skills, imitation learning enables effective training of long-horizon policies, as shown in [4, 21]. To further enhance learned behaviors, these policies can be improved through RL [11] or offline parameter optimization techniques [22]. Training a long-horizon manipulation policy directly using RL often requires manual designed transitions between different primitives. To achieve robot grasping with external dexterity, [23] incorporate a pre-generated grasp pose in the observation. They expanded the reward function to include the difference from the desired grasp pose and a penalty for collisions, to ensure the feasibility of the learned policy. Similarly, [24] split long-horizon tasks into a series of interconnected subtasks. They introduce a transition feasibility function that incrementally refines sub-policies to improve the success rate of chaining subtasks. By splitting the non-prehensile manipulation into pre-contact and post-contact stages, [25] jointly train two policies where the pre-policy is used to determine the contact pose between the end-effector and the object, and the post-policy is used to apply action on the object. These two policies are jointly trained with a highly complex and fine-tuned reward function. In this work, rather than relying on human demonstrations or reward engineering to guide robots toward predefined optimal behaviors, we propose a framework that enables the policy to discover solutions autonomously. By simplifying the problem and expanding the state-action space through privileged actions, our approach addresses the inefficiency of RL in exploring sparse-reward environments.

Curriculum Learning. Curriculum strategies are widely used in RL to enable robots to master challenging tasks. These strategies naturally guide the learning process by starting with simpler tasks and gradually increasing complexity. For instance, [26] use a curriculum to enable a human hand model with 39 muscles to rotate two Baoding balls in its palm. The work [27] devise a gravity-based curriculum to enable in-hand manipulation. They first train with the gravity vector pointing upwards, and then gradually decrease until the normal value. Additionally, [9] combine human demonstrations with an auto-curriculum strategy for dexterous manipulation. Demonstrations provide initial guidance to reduce the search space and accelerate policy convergence, while the auto-curriculum identifies areas requiring improvement and enhances them through RL. Similarly, [28] employ an adaptive curriculum based on velocity commands to train a robot to run and turn quickly on natural terrains. Our work leverages curriculum learning to gradually reduce the availability of privileged actions and guide the policy to a physically realistic solution.

Privileged Information and Actions. Simulations offer access to rich information that is often difficult to obtain in the real world. Previous works have extensively leveraged simulation to provide privileged information to enable policies to acquire essential knowledge, resulting in robust performance [29, 30, 31, 32]. Learning from privileged information improves policy learning by reducing the complexity of the state-action mapping required to be learned. [33] employ a contact-invariant optimization (CIO) method to specify when and where contact should occur on an object, using hand movements to replace this auxiliary decision variable gradually. Likewise, [34] utilized Monte Carlo Tree Search (MCTS) to explore contact points on objects, enabling robot manipulation with exceptional dexterity. By relaxing collisions between the robot and obstacles, [35] combined a curriculum strategy with a specifically designed reward with penetration terms to train a vision-based parkour policy. In [23], predefined grasp poses were used to learn grasping with extrinsic dexterity. Relaxed collisions and penalties for penetration were incorporated into the training process. Despite the success of these methods, they often require either complex task specific formulation or finely crafted reward functions. To the best of our knowledge, we are the first to propose a structured framework for solving long-horizon manipulation tasks without introducing any delicately designed reward terms.

3 Method

Our method is a structured framework that provides a solution to solving a wide range of long-horizon manipulation tasks using privileged actions with curriculum learning. It does not rely on heavy reward shaping or human priors e.g. reference trajectories. As shown in Fig. 2, the method follows a three-stage process consisting of constraint relaxation, virtual forces, and the normal setting. In this section, we first present the problem formulation of our work, then we introduce the three stages depicted in Fig. 2 in depth. After that, we introduce the reward settings we used to train our policy and how we conduct sim-to-real transfer by applying domain randomization and improved control bandwidth. In this work we focus specifically on manipulation, however our framework can be naturally extended to other robotic control tasks.

3.1 Problem Formulation

Let us consider the model-free RL setting that corresponds to manipulating objects on a tabletop. In this case our state consists of the positions and velocities of a robot and an object that is to be manipulated $\mathbf{x} = [\mathbf{q}_{R,t}, \mathbf{q}_{O,t}, \dot{\mathbf{q}}_{R,t}, \dot{\mathbf{q}}_{O,t}]$. The transition of this state is typically formulated as a Markov Decision Process (MDP). At each time step $\mathbf{t}$, the policy π_θ predicts the action $\mathbf{u}$ based on the current observation $\mathbf{x}$. The objective of training the policy π_θ is to maximise the discounted return over the episode length T. To perform this maximisation, RL frameworks typically use a simulator which internally solves an optimisation problem to compute physically realistic motions, accelerations and forces. Below, we present these elements into a concise mathematical program which provides an overview of the general tabletop manipulation problem. The detailed definition of parameters is provided in Table 2, located in Appendix D.

Maximize:

$$J(\theta) = \mathbb{E}_{\mathbf{x}_0 \sim p_0,\, \mathbf{u}_t \sim \pi_\theta(\cdot|\mathbf{x}_t)} \Big[\sum_{t=0}^{\infty} \gamma^t \, r(\mathbf{x}_t, \mathbf{u}_t) \Big] \tag{1}$$

Subject to:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \Big[\mathbf{f}(\mathbf{x}_t) \; + \; \mathbf{g}(\mathbf{x}_t)\, \mathbf{u}_t \Big] \, \Delta t, \tag{2}$$

$$\mathbf{f}(\mathbf{x}_t) = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{M}(\mathbf{q})^{-1} \Big(\mathbf{J}(\mathbf{q})^\top \begin{bmatrix} \mathbf{F}_{R,t} \\ \mathbf{F}_{O,t} \end{bmatrix} \; - \; \mathbf{c}(\mathbf{x}_t) \Big) \end{bmatrix}, \tag{3}$$

$$\mathbf{g}(\mathbf{x}_t) = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}(\mathbf{q})^{-1} \begin{bmatrix} \mathbf{I}_{n_{v_r} \times m_R} \\ \mathbf{0}_{n_{v_o} \times m_R} \end{bmatrix} \end{bmatrix}, \tag{4}$$

$$\mathbf{u}_{R,t} \sim \pi_\theta(\cdot \mid \mathbf{x}_t), \quad t = 0, 1, 2, \ldots \tag{5}$$

Maximising the reward in this case means overcoming the challenges inherent to the environment. For example the state of the object can only be changed through contact forces applied by the robot. If the object is in a non-graspable pose, it becomes particularly difficult for the robot to determine how to interact with object from scratch under sparse rewards. Equation 4 formally describes this constraint showing that any direct forces on the object via the policy are nullified. This challenge arises due to the involvement of non-prehensile manipulation to change the object to a grasp pose, which is inherently contact-rich and requires extensive long-horizon exploration at this stage.

Privileged actions serve as an effective technique to simplify this problem, facilitating more efficient exploration during policy learning. By reducing collision complexity and minimizing the need for direct interaction with the object, the robot can accomplish the task more easily, guiding the robot state-action space toward a more feasible subset. This strategy mitigates the risk of the robot becoming trapped in local optimum and significantly improves its exploration capability. The following subsections present a detailed discussion of privileged actions, reward setting, and the auto-curriculum framework.

3.2 Constraint Relaxation with Collision Management

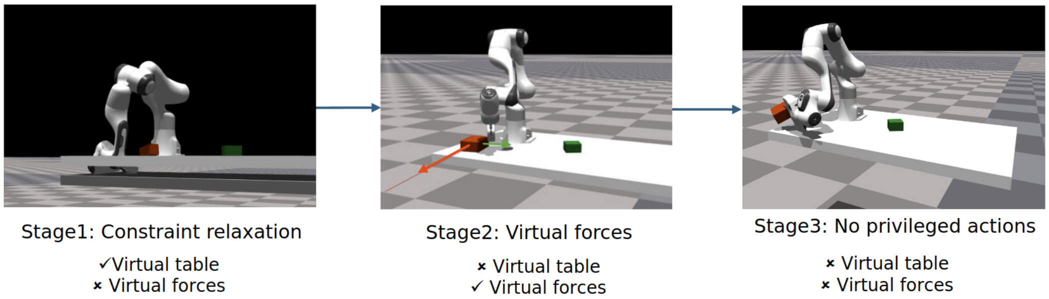
When grasping an object from an initially ungraspable pose, the table surface may be considered as an obstacle that impedes the robot from achieving a successful grasp. We first train the policy with constraint relaxation, by cancelling the collision between the robot and the table. This allows the robot to learn the manipulation skill more effectively. Concretely, we make the contact constraint forces between the robot and the table $\mathbf{F}_{R,t}$ less restrictive by increasing the distance at which contact is triggered, $\phi_R(\mathbf{x}_t)$ by Δ_R.

$$\mathbf{F}_{R,t} \geq \mathbf{0}, \phi_R(\mathbf{x}_t) + \Delta_R \; \geq 0, \big(\phi_R(\mathbf{x}_t) + \Delta_R\big)\, \mathbf{F}_{R,t} = 0, \tag{6}$$

However, it is important to note that constraint relaxation expands the robot's state-action space, potentially causing significant deviations in the action distribution. For instance, the robot may learn to lift the object using its arm rather than the gripper, leading to incorrect behaviors that create challenges in later training stages. To mitigate this issue, we introduce a virtual table that interacts with the robot and gradually increase its height until it aligns with the actual table surface. This process is illustrated in the left figure of Fig. 2, where the white table does not collide with the robot, while the grey table represents the virtual surface that enforces collision constraints. It is noticeable that the virtual table setting used in our experiments arises from simulation limitations (which restrict flexible control of object penetration).

3.3 Virtual Force

When restoring the collision relationship between the robot and table, the robot follows the previously learned policy, often attempting penetration actions. It is important to highlight that the


Stage1: Constraint relaxation

✓Virtual table
× Virtual forces

Stage2: Virtual forces

× Virtual table
✓ Virtual forces

Stage3: No privileged actions

× Virtual table
× Virtual forces

Figure 2: A structured framework utilizes privileged actions with curriculum learning. In stage 1, the robot penetrates the white table while a grey virtual table limits this penetration and is gradually lifted during training; stage 2 applies virtual forces on the object, indicated by blue and green arrows, and by stage 3, no privileged actions is used.

constraint relaxation employed in [35, 23] is effective primarily due to dense rewards that penalise penetration.When transitioning back to real collision dynamics, the dense rewards ensures that the robot receives timely reward feedback for state changes, allowing it to quickly adapt to the updated scenario.

Learning the interaction between the robot and the object presents a significant challenge, as the subset of the state-action space capable of inducing meaningful object state changes is much narrower than the robot's original state-action space. The robot must establish contact with the object to induce state transitions and maintain this contact to achieve effective manipulation. Consequently, training an RL policy for non-prehensile manipulation without reward guidance is highly challenging. To overcome this difficulty, we introduce a virtual force to promote the interaction between the robot hand and the object. Specifically, we design the trained policy to predict the force applied to the object while enforcing a constraint that ensures the robot's end-effector actions align with the virtual force. Formally, the control is sampled from $[\mathbf{u}_{R,t}, \mathbf{u}_{O,t}] \sim \pi_\theta(\cdot \mid \mathbf{x}_t)$ additionally the space in which the control can act on is modified to

$$\mathbf{g}(\mathbf{x}_t) = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}(\mathbf{q})^{-1} \begin{bmatrix} \mathbf{I}_{n_{v_r} \times m_R} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}(\mathbf{x}_t) \end{bmatrix} \end{bmatrix}, \tag{7}$$

$$\mathbf{B}(\mathbf{x}_t) = \begin{cases} \mathbf{I}_{n_{v_o} \times m_O}, & \text{if } \left\|\mathbf{q}_{O,t} - \mathbf{q}_{EE,t}\right\| < \delta_p \cdot \alpha \\ & \wedge \left\|\dot{\mathbf{q}}_{O,t} - \dot{\mathbf{q}}_{EE,t}\right\| < \delta_v \cdot \alpha, \\ \mathbf{0}_{n_{v_o} \times m_O}, & \text{otherwise.} \end{cases} \tag{8}$$

The gating matrix $\mathbf{B}(\mathbf{x_t})$ enables control action forces on the object. Specifically, $\mathbf{B}(\mathbf{x}_t)$ activates $\mathbf{I}$ only if $\left\|\mathbf{q}_{O,t} - \mathbf{q}_{EE,t}\right\| < \delta_p\, \alpha$ and $\left\|\dot{\mathbf{q}}_{O,t} - \dot{\mathbf{q}}_{EE,t}\right\| < \delta_v\, \alpha$, ensuring that the policy's force $\mathbf{u}_{O,t}$ affects the object only when the end-effector is sufficiently close in position and velocity. Otherwise, $\mathbf{B}(\mathbf{x}_t)$ is zero, leaving the object unactuated. Overall, the curriculum learning is conducted to encourage the robot movement to gradually replace and approximate the virtual force.

3.4 Reward Setting

Previous works utilising relaxed collision constraints added extra reward terms to guide the behavior of the robot [35, 23]. However, such methods require careful fine-tuning of the reward function; otherwise, the local optimum of the reward function may shift, negatively impacting learning. In this work, we do not incorporate any additional rewards to guide the robot's behavior when using privileged actions. Instead, we employ an auto-curriculum framework that allows the policy to learn autonomously and efficiently, gradually transitioning from privileged actions to the standard setting through a curriculum strategy. Details of the reward design can be found in the Appendix A.

3.5 Structured Privilege Actions with Curriculum Learning

The curriculum learning strategy is an effective approach for training robots to master complex tasks by initially focusing on simpler ones. In this work, we employ a curriculum strategy to guide the policy in progressively learning applicable behaviors by gradually reducing privileged actions. In this work, we initially set the parameter to a relatively large value, which means that there are no restrictions during the early stage of training. Once the environment achieves success, these thresholds are progressively reduced by multiplying them with a factor. The detailed process is presented in Algorithm 1 in the Appendix B.

4 Experiment Result

In this work, we focus on long-horizon, contact-rich tasks involving non-prehensile manipulation combined with grasping to lift objects from non-graspable poses. All tasks are set up and trained using the IsaacGym simulator [36].

We conduct two tasks to validate the performance of our proposed approach. The first task involves grasp and lift object from non-graspable pose using a Franka robot arm and a Franka gripper. The second task, performed in simulation, involves grasping and in-hand manipulation of several YCB objects [37], which is a more challenging task due to the use of a dexterous hand with higher degree of freedom. These experiment demonstrates the generality of our method across different setups.

Additionally, we conduct real-world experiments to show that our approach not only enables efficient policy convergence to robust solutions in simulation but is also adaptable across various platforms and tasks. More importantly, when the environment changes, with the same reward and did not include any further guidance on the non-prehensile manipulation behavior, our method still allows the policy to adapt and converge to physically acceptable and task-specific behaviors.

4.1 Grasp and Lift Object from Non-graspable Pose

Experiment setup. We evaluate our method using the Franka robot to grasp an object in an ungraspable pose. The Franka robot is set up on a table. The object used in the experiment measures 15cm × 10cm × 6cm, while the maximum opening distance of the Franka gripper is 8cm. When the object is lying flat on the table, the Franka gripper cannot directly grasp the object due to the limited opening distance of its gripper. Instead, the robot must learn how to push the object to the edge of the table and grasp it from the object's side. We further evaluated our method by placing the Franka robot in a more constrained environment with small walls surrounding the table. This setup prevented the robot from using its previous strategy of pushing the object to the edge for side grasping. Despite these constraints, our method enabled the robot to develop a stable and effective behavior, leveraging its own frame as support to do the pivot grasp. Traditionally, such behaviors require carefully designed rewards or human guidance, but our approach eliminates the need for manual intervention. Additionally, we tested both tasks with the Franka robot in real-world scenarios, demonstrating that the behaviors emerging from our method are not only robust but also physically valid and applicable in real-world environments.

Results in simulation. With our method, the Franka robot successfully demonstrated long-horizon behavior by pushing the object to the edge of the table, grasping it from the side, and lifting it. The policy also can adapt to various object pose, when the pose of the object is not suitable for directly push and grasp from the side, it will reorient the object first, and then grasp it. In the more constrained environment, where the original solution was blocked by walls, the policy also can adapt to it and leveraging the robot base as support to achieve a pivot grasp, all without human guidance in either the observation or reward function. In comparison, the vanilla PPO failed on both tasks. The robot end-effector remains at the center of the object and fails to lift, as this behavior is the local optimum of the reward function that minimize object to end-effector distance.

Results in real-world. Our focus is on grasping objects from non-graspable poses in a tabletop setting, where occlusions frequently occur, making it challenging to obtain pre-

PushGrasp				PivotGrasp			
Original	ShapeDiff	WeightDiff	PoseDiff	Original	ShapeDiff	WeightDiff	PoseDiff
8/10	8/10	7/10	7/10	9/10	7/10	8/10	7/10

Table 1: Success rates for PushGrasp and PivotGrasp tasks.

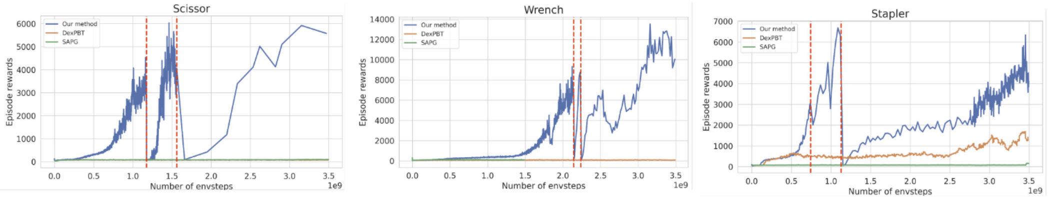


Figure 3: Reward curves comparing our method with DexPBT and SAPG on three challenging objects indicate that our framework, represented by the blue lane, performs well on all objects. The red line indicate when the stage changes.

cise pose information. Therefore, we distilled the robot's movement trajectories from simulation and transferred them to the real environment by replicating the recorded behavior. As shown in Fig. 1, we conducted two real-world experiments. The learned behavior of pushing objects to the edge of the table adapts to various object poses. When the object's pose is unsuitable for a direct push-and-grasp action, the robot first reorients the object before grasping it from the side. The most challenging task is the pivot grasp, where the robot learns to use the its base as support to perform the pivot grasp task. This behavior, learned in a constrained environment, enables the robot to stably maintain the pivot grasp pose and gradually adjust the object pose for a successful grasp.

Our real-world experiments demonstrated robust sim-to-real transfer, achieving successful task execution across variations in object shape (originally 6 cm in height compared to 4 cm), mass (originally 72 g compared to 203 g), and initial pose (randomly sampled within a 0.2 m square), with success rates shown in Table 1. Our framework aims to provide a general and broadly applicable method, and thus we did not alter the environment-provided policy observations.

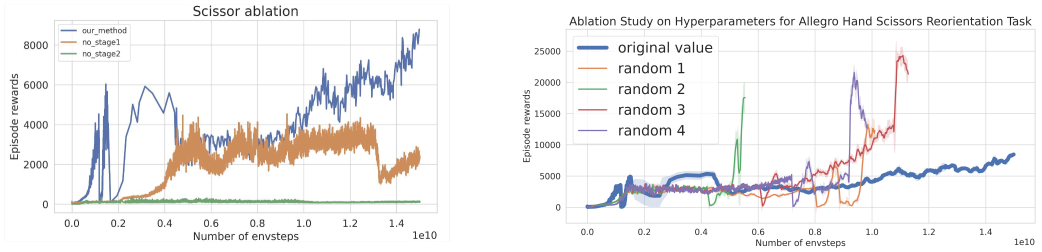
4.2 Dexterous Manipulation of Challenge YCB Objects

Experiment setup. To evaluate the generalization capability of our proposed method across different platforms, we conducted experiments involving the grasping of thin objects using a Kuka robot arm with an AllegroHand (e.g. scissors, stapler and wrench). These tasks involve grasping objects from a tabletop, lifting them, and reorienting them to achieve a specific target pose.

The work [38] focuses on functional grasping of various objects with Shadow hand using a carefully designed reward function, a novel learning framework, and predefined grasp poses. However, their results indicate that grasping thin objects remains a significant challenge. Therefore, to further validate our method, we selected three objects: scissors, stapler, and wrench, which this approach failed to grasp successfully.

For this experiment, we utilized the same environment setup provided by DexPBT [39]. DexPBT employs a population-based training method to enhance the exploration capabilities of deep reinforcement learning. Additionally, SAPG [40], another approach utilizing the same environment as DexPBT, proposes an efficient way to leverage large-scale environments by partitioning them into smaller chunks and recombining them via importance sampling. These methods are the current SOTA for this environment setup and they were chosen as baselines for comparison with our approach. To ensure a fair comparison, all experimental setups, as well as the policy's observations and reward, were kept identical across methods.

Results in simulation. As shown in Appendix E, Fig. 8, our approach enables the robot to successfully grasp the scissors by maneuvering it to the edge of the table. Although DexPBT and SAPG demonstrate strong capabilities in achieving rapid convergence and efficient exploration for randomly sized cubes, they struggle when applied to the more challenging YCB objects. The training progress for all three YCB objects is shown in Fig.3. The red line indicate the stage switch. Among these objects, the stapler is relatively thick and can be grasped directly; however, the rewards for DexPBT only converged at approximately 1500. This is due to the necessity of long-horizon exploration, where the robot must establish a stable grasp with an appropriate pose while simultaneously lifting and accurately adjusting orientation of the object to achieve the desired goal.



(a) Effectiveness of the three-stage framework on a challenging object.

(b) Robustness of our method with respect to hyperparameter variations.

Figure 4: Ablation studies evaluating (a) the contribution of the three-stage framework, and (b) the robustness of the method under different hyperparameter settings.

4.3 Ablation Study

Experiment setup. We also performed an ablation study to evaluate the importance of the three-stage privileged action curriculum learning framework. Based on the results presented in Fig. 4.2, we selected the most challenging object, scissors, for this analysis.

As shown in Fig. 4a, we compared the training performance by removing specific stages from the curriculum. Specifically, we trained the policy without Stage 1 and without Stage 2 to assess their individual contributions to the overall learning process. This comparison highlights the impact of each stage on improving the policy's ability to explore and learn effectively.

Results and analysis. It is obvious from Fig. 4a that without stage 2, the policy gets stuck in a local optimum. The constraint relaxation through collision management aids the robot in discovering a stable grasp strategy, which is crucial for successfully executing the subsequent in-hand orientation task. However, without the virtual force stage, the policy lacks sufficient exploration, leading to failure in finding a feasible solution for grasping the scissors under realistic collision constraints.

When training without the stage 1, the policy can still converge to a successful grasping behavior, but it requires significantly more training time. Notably, a policy trained from the virtual force stage directly can eventually learn a workable behavior after an extended period of exploration, rather than getting trapped in local optimum. This is because virtual force reduces the complexity of object-robot interactions, making it easier for the policy to explore effective interaction strategies. However, without stage 1, the policy struggles to select an optimal grasp pose, which is essential for the subsequent reorientation task, ultimately leading to a lack of long-horizon planning capability.

4.4 Robustness to Curriculum Hyperparameter

In our framework, the curriculum parameters, which include penetration thresholds, as well as distance and velocity thresholds, are set initially at high values to effectively minimize environmental constraints, and then automatically tightened through curriculum learning based on the task success rate. The default hyperparameter values used in this work are detailed in Appendix B. We validate the robustness of our framework by applying the same value of hyperparameters across two distinct tasks. To further confirm this robustness, we conducted an ablation study by randomly selecting parameters (penetration threshold: 0.03-0.3, distance and velocity thresholds: 0.5-2, curriculum factor: 0.5-0.9). As shown in Fig. 4b, policies trained with these randomly selected hyperparameters (four cases: orange, green, red, and purple) rapidly converge to high reward values, demonstrating the stability of our approach with respect to hyperparameter variation.

5 Conclusion

In this work, we propose a structured framework that integrates privileged actions with curriculum learning for tackling long-horizon, contact-rich manipulation tasks. Through extensive evaluations in both simulation and real-world experiments, we demonstrate that our framework can significantly enhances the policy's exploration efficiency. Our method not only facilitates robust non-prehensile manipulation under sparse reward conditions but also enables the robot to learn diverse behaviors within the same reward setting. Additionally, it outperforms SOTA methods on challenging dexterous manipulation tasks.

6 Limitation

While our framework significantly simplifies traditional reward engineering, it still requires the selection of curriculum parameters. Nonetheless, this tuning process is considerably less task-specific and reduces the overall engineering burden. The use of privileged actions enables more effective exploration, allowing the policy to discover alternative behaviors. For example, it can adopt pivot grasping strategies when conventional push-and-grasp motions are obstructed, which in turn support long-horizon planning and enhance the robustness of grasp execution. In contrast, baseline methods typically only enable simple grasp-and-lift behaviors and often fail in complex reorientation tasks. Despite introducing curriculum parameters, our method substantially improves the generality and robustness of learned behaviors while lowering the overall system design complexity. Additionally, to maintain broad applicability, we retain the original policy observations provided by the environment without additional modification. However, reliable closed-loop execution could be further enhanced by incorporating advanced perception techniques. For instance, leveraging Teacher-Student Networks [30] to distill policies from point cloud observations could help address challenges such as object tracking and occlusions, thereby improving the robustness of policy deployment in real-world scenarios. Finally, while our experiments focus on tabletop manipulation, the privileged action framework has the potential to generalize to a wider range of robot learning problems, which we leave for future work.

References

[1] O. M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. doi:10.1177/0278364919887447. URL https://doi.org/10.1177/0278364919887447.

[2] D. Kalashnikov, J. Varley, Y. Chebotar, B. Swanson, R. Jonschkowski, C. Finn, S. Levine, and K. Hausman. Scaling up multi-task robotic reinforcement learning. In *Conference on Robot Learning*, pages 557–575. PMLR, 2022.

[3] T. Haarnoja, B. Moran, G. Lever, S. H. Huang, D. Tirumala, J. Humplik, M. Wulfmeier, S. Tunyasuvunakool, N. Y. Siegel, R. Hafner, et al. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *Science Robotics*, 9(89):eadi8022, 2024.

[4] J. Luo, C. Xu, X. Geng, G. Feng, K. Fang, L. Tan, S. Schaal, and S. Levine. Multi-stage cable routing through hierarchical imitation learning. *IEEE Transactions on Robotics*, 2024.

[5] U. A. Mishra, S. Xue, Y. Chen, and D. Xu. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Conference on Robot Learning*, pages 2905–2925. PMLR, 2023.

[6] H. Ha, P. Florence, and S. Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *Conference on Robot Learning*, pages 3766–3777. PMLR, 2023.

[7] D. Wang, C. Liu, F. Chang, H. Huan, and K. Cheng. Multi-stage reinforcement learning for non-prehensile manipulation. *IEEE Robotics and Automation Letters*, 2024.

[8] J. D. A. Ferrandis, J. Moura, and S. Vijayakumar. Learning visuotactile estimation and control for non-prehensile manipulation under occlusions. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=oSU7M7MK6B.

[9] M. Bauza, J. E. Chen, V. Dalibard, N. Gileadi, R. Hafner, M. F. Martins, J. Moore, R. Pevceviciute, A. Laurens, D. Rao, et al. Demostart: Demonstration-led auto-curriculum applied to sim-to-real with multi-fingered robots. *arXiv preprint arXiv:2409.06613*, 2024.

[10] Y. Chen, C. Wang, Y. Yang, and C. K. Liu. Object-centric dexterous manipulation from human motion data. *arXiv preprint arXiv:2411.04005*, 2024.

[11] E. Triantafyllidis, F. Acero, Z. Liu, and Z. Li. Hybrid hierarchical learning for solving complex sequential tasks using the robotic manipulation network roman. *Nature Machine Intelligence*, 5 (9):991–1005, 2023.

[12] X. Yu, M. Dunion, X. Li, and S. V. Albrecht. Skill-aware mutual information optimisation for generalisation in reinforcement learning. *arXiv preprint arXiv:2406.04815*, 2024.

[13] V. Atanassov, W. Yu, A. L. Mitchell, M. N. Finean, and I. Havoutis. Constrained skill discovery: Quadruped locomotion with unsupervised reinforcement learning. *arXiv preprint arXiv:2410.07877*, 2024.

[14] H. Jiang, T. Chen, J. Cao, J. Bi, G. Lu, G. Zhang, X. Rong, and Y. Li. Stable skill improvement of quadruped robot based on privileged information and curriculum guidance. *Robotics and Autonomous Systems*, 170:104550, 2023.

[15] 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*, 7(62):eabk2822, 2022.

[16] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza. Learning high-speed flight in the wild. *Science Robotics*, 6(59):eabg5810, 2021.

[17] S. Cheng and D. Xu. League: Guided skill learning and abstraction for long-horizon manipulation. *IEEE Robotics and Automation Letters*, 2023.

[18] Z. Zhou, A. Garg, D. Fox, C. R. Garrett, and A. Mandlekar. SPIRE: Synergistic planning, imitation, and reinforcement learning for long-horizon manipulation. In *8th Annual Conference on Robot Learning*, 2024. URL `https://openreview.net/forum?id=cvUXoou8iz`.

[19] J. O. von Hartz, T. Welschehold, A. Valada, and J. Boedecker. The art of imitation: Learning long-horizon manipulation tasks from few demonstrations. *IEEE Robotics and Automation Letters*, 2024.

[20] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. In *Conference on Robot Learning*, pages 201–221. PMLR, 2023.

[21] X. Mao, G. Giudici, C. Coppola, K. Althoefer, I. Farkhatdinov, Z. Li, and L. Jamone. Dexskills: Skill segmentation using haptic data for learning autonomous long-horizon robotic manipulation tasks. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5104–5111, 2024. doi:10.1109/IROS58592.2024.10802807.

[22] N. Kumar, T. Silver, W. McClinton, L. Zhao, S. Proulx, T. Lozano-Pérez, L. P. Kaelbling, and J. Barry. Practice makes perfect: Planning to learn skill parameter policies. In *Robotics: Science and Systems (RSS)*, 2024.

[23] W. Zhou and D. Held. Learning to grasp the ungraspable with emergent extrinsic dexterity. In *Conference on Robot Learning*, pages 150–160. PMLR, 2023.

[24] Y. Chen, C. Wang, L. Fei-Fei, and K. Liu. Sequential dexterity: Chaining dexterous policies for long-horizon manipulation. In J. Tan, M. Toussaint, and K. Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 3809–3829. PMLR, 06–09 Nov 2023. URL `https://proceedings.mlr.press/v229/chen23e.html`.

[25] M. Kim, J. Han, J. Kim, and B. Kim. Pre-and post-contact policy decomposition for non-prehensile manipulation with zero-shot sim-to-real transfer. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10644–10651. IEEE, 2023.

[26] A. S. Chiappa, P. Tano, N. Patel, A. Ingster, A. Pouget, and A. Mathis. Acquiring musculoskeletal skills with curriculum-based reinforcement learning. *Neuron*, 112(23):3969–3983.e5, 2024. ISSN 0896-6273. doi:https://doi.org/10.1016/j.neuron.2024.09.002. URL `https://www.sciencedirect.com/science/article/pii/S0896627324006500`.

[27] T. Chen, J. Xu, and P. Agrawal. A system for general in-hand object re-orientation. *Conference on Robot Learning*, 2021.

[28] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid locomotion via reinforcement learning. *The International Journal of Robotics Research*, 43(4):572–587, 2024.

[29] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[30] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84):eadc9244, 2023.

[31] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik. General in-hand object rotation with vision and touch. In *Conference on Robot Learning*, pages 2549–2564. PMLR, 2023.

[32] M. Yang, C. Lu, A. Church, Y. Lin, C. Ford, H. Li, E. Psomopoulou, D. A. Barton, and N. F. Lepora. Anyrotate: Gravity-invariant in-hand object rotation with sim-to-real touch. *arXiv preprint arXiv:2405.07391*, 2024.

[33] I. Mordatch, Z. Popović, and E. Todorov. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, pages 137–144, 2012.

[34] X. Cheng, S. Patil, Z. Temel, O. Kroemer, and M. T. Mason. Enhancing dexterity in robotic manipulation via hierarchical contact exploration. *IEEE Robotics and Automation Letters*, 9(1): 390–397, 2023.

[35] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot parkour learning. *arXiv preprint arXiv:2309.05665*, 2023.

[36] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[37] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar. Yale-cmu-berkeley dataset for robotic manipulation research. *The International Journal of Robotics Research*, 36(3):261–268, 2017.

[38] T. Wu, Y. Gan, M. Wu, J. Cheng, Y. Yang, Y. Zhu, and H. Dong. Unidexfpm: Universal dexterous functional pre-grasp manipulation via diffusion policy. *arXiv preprint arXiv:2403.12421*, 2024.

[39] A. Petrenko, A. Allshire, G. State, A. Handa, and V. Makoviychuk. Dexpbt: Scaling up dexterous manipulation for hand-arm systems with population based training. In *RSS*, 2023.

[40] J. Singla, A. Agarwal, and D. Pathak. Sapg: Split and aggregate policy gradients. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, Proceedings of Machine Learning Research, Vienna, Austria, July 2024. PMLR.

Appendix

A. Details of Reward Setting

Dense rewards are often used to guide learning. However, relying on heavily shaped reward functions offers limited generalisability and can demand substantial human effort. Thus, we adopt the default reward for grasping and lifting objects from the table, as provided by IsaacGym [36], without introducing any additional modifications.. Specifically, the task we focus on in this work involves non-prehensile manipulation with multi-stage behavior. The reward for non-prehensile manipulation is sparse, as no explicit rewards are provided to guide this behavior. The reward we used in this task is defined as follows:

$$r_{total} = r_f + r_l + r_k + r_p + r_b \tag{9}$$

Where r_f is the distance reward between the robot end-effector and the object, r_l is the lifting reward, and r_k is the distance reward between the object and the goal. r_p is the penalty term, which reduces jerk motion by penalizing sudden changes in movement. Additionally, r_b is the bonus reward for successfully reaching the goal.

B. Algorithms for Three-stage Curriculum Learning

Algorithm 1 Three-Stage Curriculum Training

```
Initialize: $\Delta_R \leftarrow 0.3$, $\alpha \leftarrow 0.85$, epoch $\leftarrow 0$
Set: $\Delta_i \leftarrow \Delta_R/3$, $\alpha_d \leftarrow 0.9$, $\alpha_{min} \leftarrow 0.06$
while not end of training do
    if $\Delta_R > 0.0$ then                                  ▷ Stage 1: Virtual surface
        Train policy $\pi_\theta$ with constraint relaxation
        if success rate $> 70\%$ then
            $\Delta_R \leftarrow \Delta_R - \Delta_i$
        end if
    else if $\alpha > \alpha_{min}$ then                      ▷ Stage 2: Virtual force
        Train policy $\pi_\theta$ with virtual force
        if success rate $> 70\%$ then
            $\alpha \leftarrow \max(\alpha \cdot \alpha_d, \alpha_{min})$
        end if
    else                                                      ▷ Stage 3: No privileged actions
        Train policy $\pi_\theta$ without privileged actions
    end if
end while
```

In this work, We initialize the penetration offset as $\Delta_R = 0.3$, positioning the virtual table 30 cm below the actual surface, and progressively raise it as the policy achieves success. The initial distance and velocity thresholds are set to $\delta_p = 1$, $\delta_v = 0.5$, and the curriculum factor $\alpha = 0.85$, introducing minimal constraints during early training. Additionally, the virtual force applied to the object is restricted to the $\mathbf{x}$ and $\mathbf{y}$ directions. As the agent meets the success condition, these thresholds are scaled by the factor α, which is updated using $\alpha = \text{clamp}(\alpha \cdot 0.9, 0.06, 0.85)$; otherwise, α remains unchanged. This curriculum learning strategy gradually tightens constraints, enabling a smooth transition from relaxed training conditions to realistic interactions.

C. Additional Experiments

Our framework is designed with generality by first relaxing environmental constraints, allowing rapid adaptation by the actor and accurate future reward estimation by the critic. The second stage further encourages diverse robot-object interactions, promoting effective exploration, which is particularly beneficial in sparse reward settings. The benefits of our framework may be less useful in simple manipulation tasks that do not require significant exploration (e.g., in-hand reorientation where the object is already set within the palm), our approach substantially enhances performance for more complex, long-horizon manipulation tasks. The Allegro Hand reorientation task demonstrates that our framework can effectively address long-horizon tasks involving push-to-grasp with reorientation. To further showcase the capabilities and generality of our approach, we conducted several additional experiments.

Object grasping and throwing. We evaluated our method on a task requiring the Allegro Hand to grasp various YCB objects and throw them into a bucket. Figure 5 compares our method against the SOTA methods, demonstrating its superior performance on these challenging tasks.

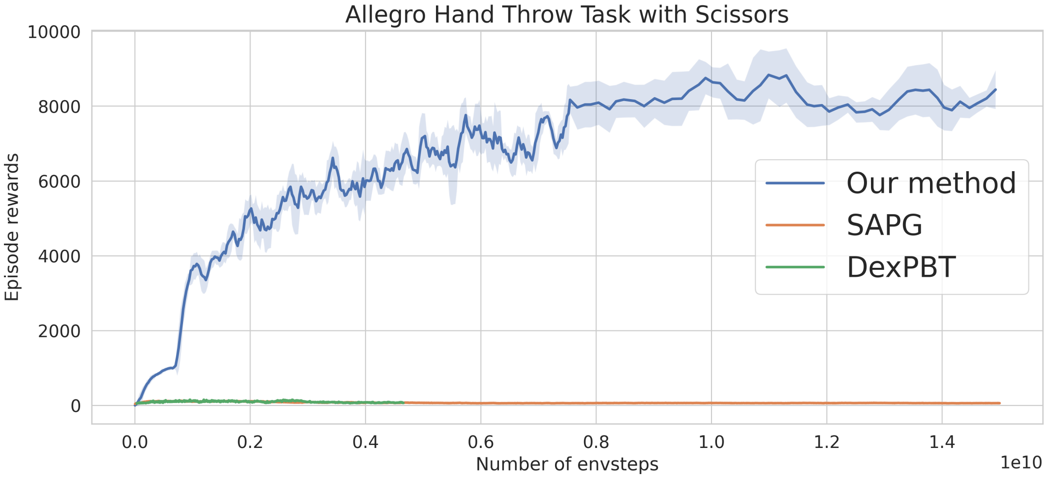


Figure 5: Performance comparison on the challenging YCB object in the throw task.

Push-to-insert task. To highlight the generality of our framework, we further applied it directly to a complex push-to-insert task, as shown in Figure 6. By relaxing object-hole collision constraints and leveraging virtual forces to guide the policy to adapt within the highly constrained goal region, our method successfully solves this task where a standard PPO baseline fails.

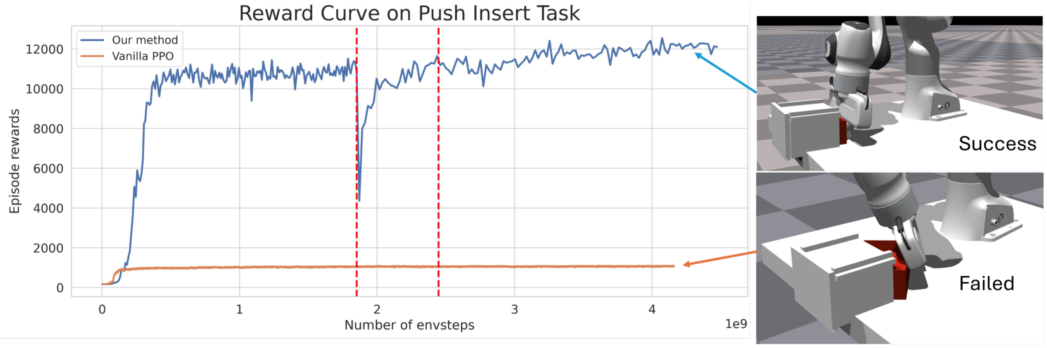


Figure 6: Robot push and insert task.

Robustness to random seeds. We validated the stability of our framework by running our most challenging task, scissors reorientation, with five different random seeds. As shown in Figure 7, all runs consistently converge to a high reward, confirming that our method is robust and not sensitive to random initialization.

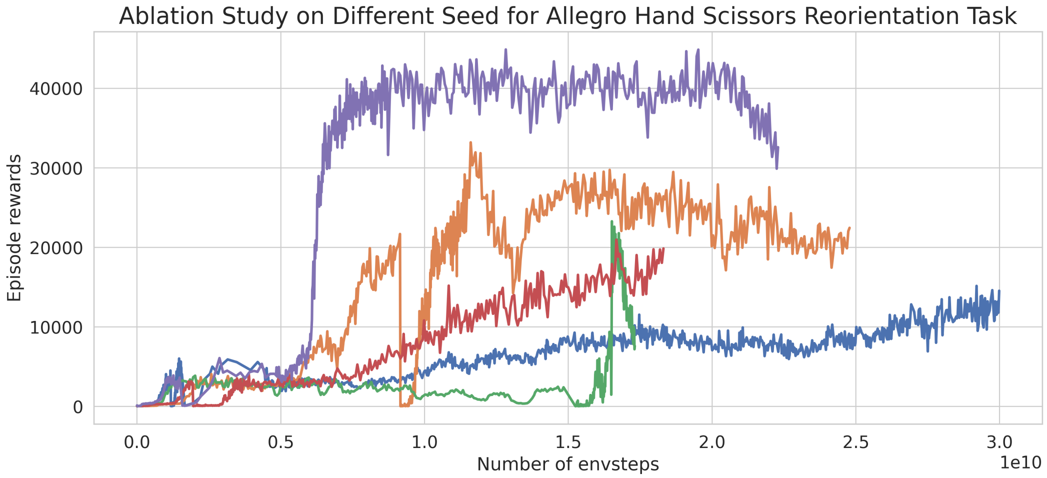


Figure 7: Robustness on different seeds

D. Symbol Definition

Symbol	Definition
$\mathbf{q}_t = \left[\mathbf{q}_{R,t}, \mathbf{q}_{O,t}\right] \in \mathbb{R}^{n_p=n_{p_r}+n_{p_o}}$	**System state**, representing the robot's and object's positions.
$\dot{\mathbf{q}}_t = \left[\dot{\mathbf{q}}_{R,t}, \dot{\mathbf{q}}_{O,t}\right] \mathbb{R}^{n_v=n_{v_r}+n_{v_o}}$	**System state**, representing the robot's and object's velocities.
$\mathbf{x}_t = \left[\mathbf{q}_{R,t}, \mathbf{q}_{O,t}, \dot{\mathbf{q}}_{R,t}, \dot{\mathbf{q}}_{O,t}\right] \in \mathbb{R}^{n_p+n_v}$	**System state**, representing the positions and velocities of the robot and object.
$\mathbf{u}_t = \left[\mathbf{u}_{R,t}, \mathbf{u}_{O,t}\right] \in \mathbb{R}^m$	**Control input**, partitioned into: $\mathbf{u}_{R,t} \in \mathbb{R}^{m_R}$ (robot joint torques/controls) and $\mathbf{u}_{O,t} \in \mathbb{R}^{m_O}$ (linear force applied to the object).
$\mathbf{f}(\mathbf{x}_t)$	**Passive dynamics**, including inertial and external contact forces.
$\mathbf{g}(\mathbf{x}_t)$	**Control influence** on state evolution.
$\mathbf{B}(\mathbf{x}_t) \in \mathbb{R}^{n_{v_o} \times m_O}$	**Gating matrix** that regulates the force applied to the object.
$\mathbf{M}(\mathbf{q}_t) \in \mathbb{R}^{n_v \times n_v}$	**Mass/inertia matrix** for the robot and object in generalized coordinates.
$\mathbf{c}(\mathbf{x}_t)$	**Bias term** accounting for Coriolis, gravity, and frictional forces.
$\mathbf{J}(\mathbf{q}_t)$	**Jacobian** mapping contact-space forces to joint torques.
$r(\mathbf{x}_t, \mathbf{u}_t)$	**Reward function** at each timestep.
$\gamma \in (0, 1]$	**Discount factor** for long-horizon returns.
$\delta_p, \delta_v > 0$	**Distance and velocity thresholds** for gating object forces.
α	**Curriculum factor** controlling δ_p and δ_v thresholds.
$\phi_R(\mathbf{x}_t), \phi_O(\mathbf{x}_t)$	**Signed distances** to the table for the robot's end-effector and the object.
$\mathbf{F}_{R,t}, \mathbf{F}_{O,t}$	**Normal contact forces**, e.g., forces between the robot, object, and table.
Δ_R	**Penetration offset** regulating robot-table collision relaxation.
$\mathbf{x}_0 \sim p_0$	**Initial state distribution**.

Table 2: Symbol definitions used in our framework.

E. AllegroHand Grasp Challenging YCB Object

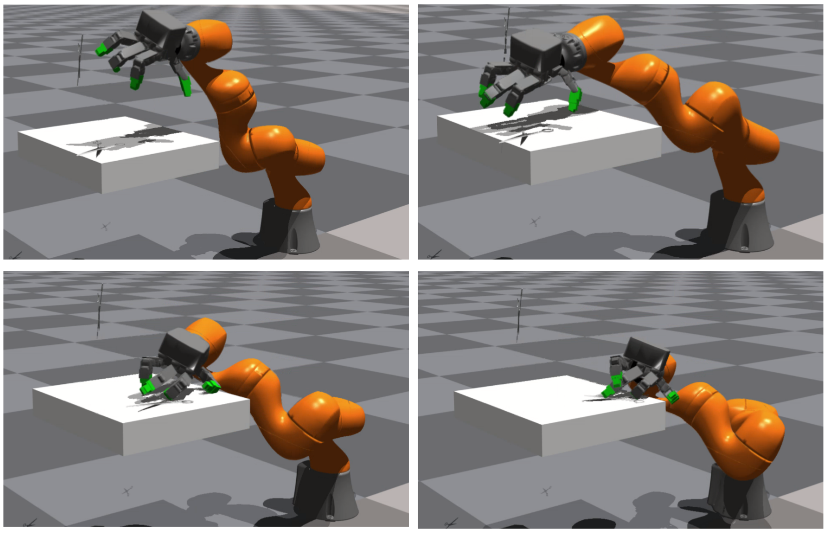


Figure 8: Using our framework and despite the absence of a specific reward indication for non-prehensile manipulation skills, the robot learns to grasp and lift the scissors by first pushing them to the edge of the table.