2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

One-Shot Transfer of Long-Horizon Extrinsic Manipulation Through Contact Retargeting

Albert Wu[1], Ruocheng Wang[1], Sirui Chen[1], Clemens Eppner[2], and C. Karen Liu[1]

***Abstract*— Extrinsic manipulation, the use of environment contacts to achieve manipulation objectives, enables strategies that are otherwise impossible with a parallel jaw gripper. However, orchestrating a long-horizon sequence of contact interactions between the robot, object, and environment is notoriously challenging due to the scene diversity, large action space, and difficult contact dynamics. We observe that most extrinsic manipulation are combinations of short-horizon primitives, each of which depend strongly on initializing from a desirable contact configuration to succeed. Therefore, we propose to generalize one extrinsic manipulation trajectory to diverse objects and environments by retargeting contact requirements. We prepare a single library of robust short-horizon, goal-conditioned primitive policies, and design a framework to compose state constraints stemming from contacts specifications of each primitive. Given a test scene and a single demo prescribing the primitive sequence, our method enforces the state constraints on the test scene and find intermediate goal states using inverse kinematics. The goals are then tracked by the primitive policies. Using a 7+1 DoF robotic arm-gripper system, we achieved an overall success rate of 80.5% on hardware over 4 long-horizon extrinsic manipulation tasks, each with up to 4 primitives. Our experiments cover 10 objects and 6 environment configurations. We further show empirically that our method admits a wide range of demonstrations, and that contact retargeting is indeed the key to successfully combining primitives for long-horizon extrinsic manipulation. Code and additional details are available at `stanford-tml.github.io/extrinsic-manipulation`.**

I. Introduction

Extrinsic manipulation describes the usage of environment contact to aid manipulation [28] and is an emerging field in robotic manipulation research. Leveraging environment contacts allows simple parallel jaw grippers to achieve complex tasks that are otherwise impossible. For instance, objects in ungraspable initial poses can be picked up by first executing a sequence of pregrasp motions involving pushing, pivoting, and pulling [3, 38, 37].

Achieving extrinsic manipulation requires a holistic orchestration of contacts interactions between the robot, object, and environment. In particular, the robot must be able to address the diverse object and environment geometries present in the real world. Earlier works attempt to perform control synthesis by modeling contact dynamics explicitly [9, 12, 1]. However, due to the inherent difficulty with modeling contact dynamics, these works are restricted to manipulating known objects with simple geometries. Recent literature on extrinsic manipulations have sought to produce reinforcement learning(RL)-based extrinsic manipulation policies that generalizes to novel objects [38, 21, 39, 8, 37], but none has successfully generalized to novel environments to our best knowledge.

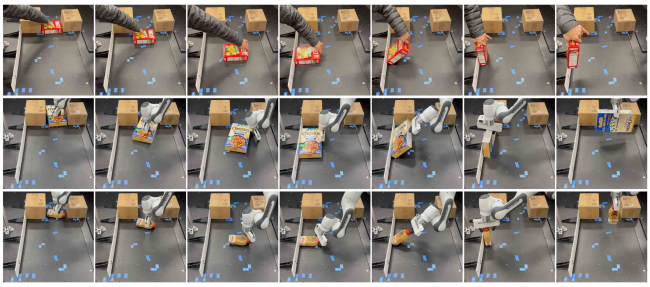


Fig. 1: Retargeting the *object retrieval* task from a human demo (top) to *oat* (middle) and *flapjack* (bottom). This 4-primitive task involves *pulling* the object from between the obstacles, *pushing* it to the wall, *pivoting* against the wall to expose a graspable edge, and finally *grasping* the object. Each row shows a trajectory in temporal order from left to right. Please refer to our supplementary video and website for animations.

Furthermore, many applications in manipulation, such as occluded grasping [38], are long-horizon in nature and require multiple contact switches. To overcome the difficulty in producing long-horizon plans, some works in literature seek to leverage demonstration [14, 6, 24]. However, to achieve achieve contact-rich manipulation, hundreds of demonstrations for a single task may be necessary [14, 6].

Other works seeking to achieve long-horizon manipulation leverage a hierarchical structure to abstract away repetitive, low-level motion "primitives" [26, 16, 11, 2]. While we observe that most long-horizon extrinsic manipulation is nothing more than a combination of short-horizon, fixed-contact configuration primitives, for instance "push" and "pivot", extrinsic manipulation primitives are significantly more challenging to abstract than simple primitives such as pick, place, or end effector movement [26, 16]. Extrinsic manipulation primitives leverage sophisticated contact interactions, for instance "pivot" requires an object to be in contact with an environment obstacle. This imposes scene-dependent contact preconditions and exacerbates the sim-to-real gap. Nevertheless, once the contact precondition is satisfied, an extrinsic manipulation primitive is much more likely to succeed.

We therefore propose to solve two subproblems instead of the full long-horizon extrinsic manipulation problem: obtain-

*This work was supported by NSF-FRR-2153854 and NSF-NRI-2024247.

[1]Computer Science Department, Stanford University, Stanford, CA 94305, USA. {amhwu, rcwang, ericcsr, karenliu}@cs.stanford.edu

[2]NVIDIA, Seattle, WA 98105, USA ceppner@nvidia.com

ing primitives for a given contact configuration that are robust under object and environment variations, and initializing each primitive in the desired contact configuration. Moreover, we observe that the sequence of primitives is typically fixed for a given manipulation objective. For instance, "occluded grasping by pushing object against an obstacle and pivoting to expose graspable edge" always has a sequence of "push-pivot-grasp." Hence, one may circumvent the combinatorially complex contact sequence planning by collecting a task demonstration that fixates the contact sequence.

In this paper, we describe a method to generalize long-horizon extrinsic manipulation plans from a single demonstration. Our contributions are:

- **Contact retargeting framework** to respect contact constraints while chaining extrinsic manipulation primitives. We formalize a process to merge semantic contact specifications from adjacent primitives, and leverage inverse kinematics (IK) to find states achieving such requirements.
- **One-shot transfer pipeline of extrinsic manipulation task demo** to diverse object and environment. We prepare one library of 4 extrinsic manipulation primitives robust to scene variations. By leveraging contact retargeting, our pipeline merely takes a single task demo of any primitive combination to achieve the same task in a distinct scene.
- **Extensive hardware validation over 4 long-horizon tasks** covering 10 objects and 4 environments using a wide range of demos. Our method achieved an overall success rate of 80.5% and outperformed [38] on occluded grasping. Ablation shows contact retargeting is indeed the key to successfully chaining extrinsic manipulation primitives.

II. Related work

A. Manipulation using environment contacts

Many works in literature have explored applications that require environment contacts, such as in-hand repositioning with a simple gripper [10, 33], grasping from an initially ungraspable pose [9, 38, 37, 34, 13, 27], and realignment for industrial assembly [28]. Typically, the manipulated object is pushed along or slid against a flat surface in the environment, such as a tabletop or a wall [9, 38, 21, 39, 12, 8, 37, 18, 13]. In some cases, gravity is also leveraged for in-hand reorientation [10, 33].

To produce these motions, earlier works often synthesize control policies through hand-designed strategies [28, 9, 13] or physical models [10, 9, 12, 1, 18, 5, 27]. However, the inherent difficulty of modeling contact-rich motion restricts these works to manipulating known objects with simple geometries. Recent papers instead uses RL to obtain a feedback policy [38, 21, 39, 8, 37, 34], which can be significantly more robust and generalize to geometries never seen during training [38, 39, 8, 34]. Attention to variable environments, meanwhile, is much more scarce. Among the papers reviewed in this section, only [5, 34] consider environment variability. The discussion in [5] is limited to simulation experiments, and [34] uses a fingerless, ball-shaped end effector with no grasping capability.

B. Composing primitives for long-horizon manipulation

Divide-and-conquer has been a popular way to tackle long-horizon manipulation in literature. The learning-based robotics community typically leverage the framework of Parameterized Action MDP [25] to break down tasks into shorter horizon primitives. The primitives can be manually specified [11] or learned [26]. A high-level policy, which may be based on RL [11, 26], Graph Neural Network [2], or imitation learning [16] is then used to determine the sequence and parameters of the primitives.

The main challenge with primitive-based approaches is ensuring the composability of the primitives. Many advanced primitives cannot succeed from arbitrary initial states and experience a large sim-to-real gap. [11] has no hardware experiments. [26, 16] only have hardware experiments on pick-and-place tasks and end effector movement. [2] can only perform a single stowing task. [4] uses dexterous manipulation primitives that must be fine-tuned for a specific execution sequence. Composing complex primitives arbitrarily is still a missing ability in literature.

We note that primitive sequence planning, which is considered in [11, 26, 2], is beyond the scope of this work. As such, this section excludes the task and motion planning (TAMP) literature, which has an emphasis on planning such sequences. The interested reader is referred to [17] for a thorough review on TAMP. Our contribution focuses on assuring the feasibility of the primitive sequence; choice of the discrete sequence is obtained directly from the single demonstration.

C. Demonstration for manipulation

Leveraging demonstration for robotic manipulation has gained significant traction in recent years. The robot learning community has explored many approaches, including variations of behavior cloning (e.g. [14, 36, 6, 24]) and fine-tuning a pretrained RL agent (e.g. [32, 31]). We refer the readers to [29] for a thorough review.

While these methods have achieved many sophisticated manipulation tasks, the scalability and generalizability of the policies produced by these methods is unclear. Despite novel systems such as [15, 7] lowering the barrier to collect demonstrations, the sheer amount of data needed means training a policy is still costly. Every task in [6, 14, 7] requires between 50 to over 500 demonstrations. Some works such as [36, 19, 32] emphasize the use of single demonstrations, but the tasks achieved by these approaches are restricted to end-effector pose movement with a firmly grasped object. Achieving complex manipulation tasks from few demonstrations remains an open problem to the best of our knowledge.

III. Notation and problem definition

We consider the quasi-static extrinsic manipulation of an object with geometry $\mathcal{O} \in \boldsymbol{\mathcal{O}}$ in environment $\mathcal{E} \in \boldsymbol{\mathcal{E}}$ using a 7-degree-of-freedom (DoF) robotic arm with a 1DoF parallel jaw gripper. The robot system has state $\boldsymbol{q} \in \mathcal{Q} \subseteq \mathbb{R}^8$, and the pose of the object is $\boldsymbol{x} \in \mathcal{X} \subseteq SE(3)$. We adopt the

shorthand $\boldsymbol{s}_t \triangleq (\boldsymbol{x}_t, \boldsymbol{q}_t) \in \mathcal{S} \triangleq \mathcal{X} \times \mathcal{Q}$ to denote the state of the system. Denote the control action as $\boldsymbol{u} \in \mathcal{U} \subset \mathbb{R}^8$. The system is governed by the environment and object dependent, discretized dynamics: $\boldsymbol{s}_{t+1} = {}^{\mathcal{E},\mathcal{O}}f(\boldsymbol{s}_t, \boldsymbol{u}_t)$. A $T+1$ step trajectory can thus be described as $\left\{{}^{\mathcal{E},\mathcal{O}}\boldsymbol{s}_t\right\}_{t=0,1,\dots,T}$.

We first make assumptions that are commonly satisfied in manipulation scenarios.

Assumption 1. *$\mathcal{E}$ is fixed throughout the task.*

Assumption 2. *The robotic manipulator is fully actuated. A motion planner is available such that the robot can reach anywhere within the workspace without collisions and does not obstruct any object-environment contacts.*

To concretely describe contact configurations in the system, we adopt Definitions 1 and 2.

Definition 1 (Contact configuration)**.** *A contact configuration describes a **minimal set** of semantic contact requirements, for instance "robot finger in contact with the top of an object; object in contact with the obstacle." We denote this as $\sigma \in \Sigma$. We use $\sigma^{\boldsymbol{x}}$ and $\sigma^{\boldsymbol{q}}$ to denote the subsets of σ concerning semantic environment-object and robot-object contact configurations. $\sigma^{\boldsymbol{x}} \cup \sigma^{\boldsymbol{q}} = \sigma$. Under Assumption 2, robot-environment contact is not considered. While our flexible framework allows each manipulation primitive to define its own contact requirements, we define a small set of general contact configurations used to implement all the manipulation primitives in this paper (Section IV-B).*

Definition 2 (Contact configuration in $\mathcal{E}, \mathcal{O}$)**.** *For a specific $\mathcal{E}, \mathcal{O}$, ${}^{\mathcal{E},\mathcal{O}}\sigma \subset \mathcal{S}$ concretely defines a set of robot and object states that satisfies such semantic contact requirements. By assumption 2, environment-object contact is independent of the robot configuration, thus ${}^{\mathcal{E},\mathcal{O}}\sigma^{\boldsymbol{x}} \subset \mathcal{X}$ defines a set of object states that satisfy $\sigma^{\boldsymbol{x}}$. On the other hand, robot-object contact is dependent on both the object state and robot state. Thus we denote ${}^{\mathcal{E},\mathcal{O}}\sigma^{\boldsymbol{q}|\boldsymbol{x}} \subset \mathcal{Q}$ as the set of robot states that satisfy $\sigma^{\boldsymbol{q}}$ given a particular $\boldsymbol{x}$.*

To allow abstracting the robot-object interaction away and admitting demonstrations performed by arbitrary end effectors, we define the following:

Definition 3 (Freestanding object states)**.** *For a given $\mathcal{E}, \mathcal{O}$, the "freestanding object states" are objects states that stay unchanged indefinitely unless there is robot-object contact. We denote this as ${}^{\mathcal{E},\mathcal{O}}\mathcal{X}_s$.*

Problem 1 (Retargeting extrinsic manipulation)**.** *Consider an extrinsic manipulation trajectory $\left\{{}^{\bar{\mathcal{E}},\bar{\mathcal{O}}}\bar{\boldsymbol{s}}_t\right\}_{t=0,1,\dots,T}$ collected in environment $\bar{\mathcal{E}}$ on object $\bar{\mathcal{O}}$ under $\bar{\boldsymbol{s}}_{t+1} = {}^{\bar{\mathcal{E}},\bar{\mathcal{O}}}f(\bar{\boldsymbol{s}}_t, \bar{\boldsymbol{u}}_t)$. The trajectory satisfies a manipulation objective $\mathcal{G} \subseteq \mathcal{X}$, $\bar{\boldsymbol{x}}_T \in \mathcal{G}$. Additionally, the trajectory progresses through N contact configurations $\sigma_1, \dots, \sigma_N$, and has $n-1$ contact switches at $0 < t_1 < \cdots < t_{N-1} < T$ at freestanding states, i.e.*

$$\bar{\boldsymbol{s}}_t \in {}^{\bar{\mathcal{E}},\bar{\mathcal{O}}}\sigma_i,\ \forall t_i < t < t_{i+1}, \tag{1}$$

$$\bar{\boldsymbol{x}}_{t_i} \in {}^{\bar{\mathcal{E}},\bar{\mathcal{O}}}\sigma_i^{\bar{\boldsymbol{x}}} \cap {}^{\bar{\mathcal{E}},\bar{\mathcal{O}}}\sigma_{i+1}^{\bar{\boldsymbol{x}}} \cap {}^{\bar{\mathcal{E}},\bar{\mathcal{O}}}\mathcal{X}_s \tag{2}$$

Given a test environment $\mathcal{E}$ and a test object $\mathcal{O}$, find a policy $\boldsymbol{\pi}(\cdot)$ such that, under a strictly-increasing time-remapping function $\eta : \mathbb{R} \mapsto \mathbb{R}$, $\tau = \eta(t)$, $\eta(0) \equiv 0$, $d\eta/dt > 0$, $\mathcal{O}$ achieves $\mathcal{G}$ in $\mathcal{E}$

$$\boldsymbol{s}_{\tau+1} = {}^{\mathcal{E},\mathcal{O}}f\left(\boldsymbol{s}_\tau, \boldsymbol{\pi}(\boldsymbol{s}_\tau)\right),\ \boldsymbol{x}_{\eta(T)} \in \mathcal{G}, \tag{3}$$

using the same contact sequence

$$\boldsymbol{s}_{\eta(t)} \in {}^{\mathcal{E},\mathcal{O}}\sigma_i,\ \forall t_i < t < t_{i+1} \tag{4}$$

$$\boldsymbol{x}_{\eta(t_i)} \in {}^{\mathcal{E},\mathcal{O}}\sigma_i^{\boldsymbol{x}} \cap {}^{\mathcal{E},\mathcal{O}}\sigma_{i+1}^{\boldsymbol{x}} \cap {}^{\mathcal{E},\mathcal{O}}\mathcal{X}_s. \tag{5}$$

Lastly, we make Assumption 3 to ensure the problem can be solved with the same manipulation strategy as the demo.

Assumption 3. *Problem 1 can be solved using contact switches at freestanding states, i.e.*

$${}^{\mathcal{E},\mathcal{O}}\mathcal{X}_s \cap {}^{\mathcal{E},\mathcal{O}}\sigma_i^{\boldsymbol{x}} \cap {}^{\mathcal{E},\mathcal{O}}\sigma_{i+1}^{\boldsymbol{x}} \neq \emptyset, \forall i = 1, \dots, N-1.$$

IV. Generalizing Extrinsic Manipulation Demos with Contact Retargeting

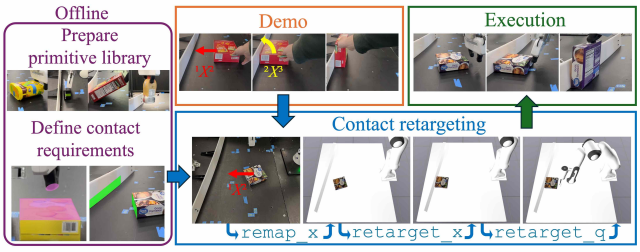


Fig. 2: Approach overview. We prepare a primitive library and define each primitive's contact requirements online. Given a demo task trajectory and a test scene, we retarget the demo to the test scene by enforcing contact requirements. The demo's primitive sequence is then used to perform the task in the test scene.

Our approach is grounded in the following observation:

1) Long-horizon extrinsic manipulation can be decomposed into a sequence of primitives based on contact switches.
2) A small set of primitives is sufficient to capture most extrinsic manipulation.
3) The success of each primitive is highly dependent on satisfying the desired σ.
4) Under Assumption 2 and 3, satisfying σ only entails selecting an environment-dependent $\boldsymbol{x}$.

Observations 1 and 2 motivates building a primitive library to handle different scenarios. Observations 3 and 4 suggests that the key to generalizing extrinsic manipulation from $\bar{\mathcal{E}}, \bar{\mathcal{O}}$ to $\mathcal{E}, \mathcal{O}$ is to find object states that all σs are maintained. We thus break down Problem 1 into the following steps:

1) Prepare a library of short-horizon primitives for any $\mathcal{E}, \mathcal{O}$ (Sec. IV-A, IV-B)
2) Identify the primitive sequence in the demonstration (Sec. IV-D)
3) Remap the object states from $\bar{\mathcal{E}}, \bar{\mathcal{O}}$ to $\mathcal{E}, \mathcal{O}$ (Sec. IV-C)
4) Combine the sequence of remapped primitive sequence to achieve $\mathcal{G}$ (Sec. IV-E)

We summarize our approach in Fig. 2. The rest of this section describes each of the steps in detail. In Section VI, we show empirically that this is a much more tractable formulation for generalizing long-horizon extrinsic manipulation.

A. Primitive library

We build a primitive library Π to handle short extrinsic manipulation tasks that start from a single contact configuration σ_i. In particular, we seek to develop a goal-conditioned policy $\boldsymbol{\pi}_{\sigma_i}$ where, given any $\mathcal{E} \in \boldsymbol{\mathcal{E}}, \mathcal{O} \in \boldsymbol{\mathcal{O}}$, initial state $\boldsymbol{s}_0 = (\boldsymbol{x}_0, \boldsymbol{q}_0) \in {}^{\mathcal{E},\mathcal{O}}\sigma_i$, and intermediate manipulation objective $\mathcal{G}_i \subseteq {}^{\mathcal{E},\mathcal{O}}\sigma_i^{\boldsymbol{x}} \subset \mathcal{X}$, the state evolution

$$\boldsymbol{s}_{t+1} = {}^{\mathcal{E},\mathcal{O}}f\left(\boldsymbol{s}_t, \boldsymbol{\pi}_{\sigma_i}(\boldsymbol{s}_t; \mathcal{E}, \mathcal{O}, \mathcal{G}_i)\right), \tag{6}$$

eventually leads to a state in $\mathcal{G}_i$. Table I summarizes our 4 primitives, *push, pivot, pull, grasp*. The primitives may be obtained from any technique, including model-based and learning-based, and leverage different control method. Our primitives use joint impedance control and operational space control (OSC) [20]. The restriction of initializing in σ_i and the short task horizon makes producing π_{σ_i} for arbitrary $\mathcal{E}, \mathcal{O}$ significantly easier. Our primitive library is scalable: to add an additional primitive, one only need to provide π_{σ_i} and implement σ_i (Section IV-B).

We note that a robustly designed π_{σ_i} may relax the requirements specified by σ_i. Enforcing contact configurations often requires real-time feedback control. To increase hardware performance, π_{σ_i} may be designed such that states in the vicinity of ${}^{\mathcal{E},\mathcal{O}}\sigma$ can still successfully execute. Our pivot and pull primitives leverage compliance and feedback control to ensure the robot-object and environment-object contacts are maintained. We also discovered the pushing policy performs better without specifying the robot-object contact; the RL policy is able to choose the contact implicitly.

B. Implementing σ

We implement σ as state constraints in the inverse kinematics (IK) problem detailed in Section IV-C. States satisfying the constraints here are states in the set ${}^{\mathcal{E},\mathcal{O}}\sigma$, though the set is not computed explicitly in practice. The following environment-object constraint types are implemented for $\sigma^{\boldsymbol{x}}$.

1) **Ground**: the object is in contact with the ground.
2) **Wall**: a lower edge of the object is in contact and orthogonal to the wall normal. This is implemented using the bounding box of the object. Of the 4 vertices that are closest to the wall, the 2 lowest ones must be on the wall, and the distance between the wall and the object is 0.

The following robot-object constraint types are defined for $\sigma^{\boldsymbol{q}|\boldsymbol{x}}$. As $\boldsymbol{x}$ is given, constraints specified relative to the world frame and environment are well defined.

1) **Top**: the gripper fingertips are approximately in contact with the "top" of the object. This is implemented by finding the 4 vertices with the highest world z coordinate of the object's bounding box, then finding the geometric center of the 4 points. The robot-object contact point is found by moving the end effector from geometric center along world $-z$ until contact established.
2) **Antipodal**: the gripper fingertips are in contact with the object on the opposite side of the wall-object contact. This is implemented as an intersection of two constraints: the distances between the object and the fingertips are zero; the fingertips are within a cone centered at the object's geometric center, has the wall's normal as its axis, and has a half-angle of $\pi/6$.
3) **Grasp**: the end effector is placed such that closing the gripper results in a top-down object grasp.

The usage of these constraints in our extrinsic manipulation primitives is summarized in Table I.

C. Contact retargeting

We propose a *contact retargeting* procedure to ensure each primitive is started from a state in its contact configuration regardless of $\mathcal{E}, \mathcal{O}$. First, we observed from Assumption 3 that the robot $\boldsymbol{q}$ can change arbitrarily during each primitive switch:

Proposition 1 (Arbitrary robot-object contact switch at a freestanding state)**.** *$\forall \boldsymbol{x} \in {}^{\mathcal{E},\mathcal{O}}\mathcal{X}_s \cap {}^{\mathcal{E},\mathcal{O}}\sigma_i^{\boldsymbol{x}} \cap {}^{\mathcal{E},\mathcal{O}}\sigma_{i+1}^{\boldsymbol{x}}$, there exists $\boldsymbol{q}_1 \in {}^{\mathcal{E},\mathcal{O}}\sigma_i^{\boldsymbol{q}|\boldsymbol{x}}$ and $\boldsymbol{q}_2 \in {}^{\mathcal{E},\mathcal{O}}\sigma_{i+1}^{\boldsymbol{q}|\boldsymbol{x}}$ such that $(\boldsymbol{x}, \boldsymbol{q}_1) \in {}^{\mathcal{E},\mathcal{O}}\sigma_i$ and $(\boldsymbol{x}, \boldsymbol{q}_2) \in {}^{\mathcal{E},\mathcal{O}}\sigma_{i+1}$.*

By Proposition 1, in order to start the next primitive $\pi_{\sigma_{i+1}}$ from a state in ${}^{\mathcal{E},\mathcal{O}}\sigma_{i+1}$, we only need the previous primitive to end at an object state in ${}^{\mathcal{E},\mathcal{O}}\sigma_{i+1}^{\boldsymbol{x}}$. The robot can be moved subsequently to satisfy ${}^{\mathcal{E},\mathcal{O}}\sigma_{i+1}^{\boldsymbol{q}}$. This leads to a two-stage contact retargeting subroutine: `retarget_x` and `retarget_q`. Both are implemented as IK problems using Drake [35]. Additionally, we map the demo object states to the test object using `remap_x` to serve as the initial guess for `retarget_x`.

`remap_x` extracts the relative transforms $\left\{{}^{i}X^{i+1}\right\}$ in the demo object states $\{\bar{\boldsymbol{x}}_{t_i}\}$ and apply them to the test object's initial state $\boldsymbol{x}_0$ to obtain a sequence of initial guess object states $\{\tilde{\boldsymbol{x}}_i\}$, i.e.

$$\bar{\boldsymbol{x}}_{t_{i+1}} = {}^{i}X^{i+1}(\bar{\boldsymbol{x}}_{t_i}), \;\; \tilde{\boldsymbol{x}}_{i+1} \triangleq {}^{i}X^{i+1}(\boldsymbol{x}_i). \tag{7}$$

`retarget_x` finds $\mathcal{G}_i \subseteq {}^{\mathcal{E},\mathcal{O}}\sigma_i^{\boldsymbol{x}} \cap {}^{\mathcal{E},\mathcal{O}}\sigma_{i+1}^{\boldsymbol{x}}$ given $\sigma_i^{\boldsymbol{x}}, \sigma_{i+1}^{\boldsymbol{x}}, \mathcal{E}, \mathcal{O}, \tilde{\boldsymbol{x}}_{i+1}$ by solving the following IK problem:

$$\min_{\boldsymbol{x}} \; d(\boldsymbol{x}, \tilde{\boldsymbol{x}}_{i+1}), \;\; \text{s.t. } \boldsymbol{x} \in {}^{\mathcal{E},\mathcal{O}}\sigma_i^{\boldsymbol{x}} \cap {}^{\mathcal{E},\mathcal{O}}\sigma_{i+1}^{\boldsymbol{x}} \tag{8}$$

$d(\cdot, \cdot)$ is implemented as the L_2 norm of the position difference. This formulation encourages the retargeted object

TABLE I: Extrinsic manipulation primitives.

	Push	Pull	Pivot	Grasp
Description	Push from side	Pull from top	Pivot against environment	Top-down grasp
Type	RL trained in Isaac Gym [23]	Designed	Designed	Designed
Control type	Joint impedance	OSC	OSC	OSC
State feedback	$\boldsymbol{x}, \boldsymbol{q}$	Open loop	$\boldsymbol{q}$	$\boldsymbol{x}$
Environment contact requirements $\sigma^{\boldsymbol{x}}$	Ground	Ground	Ground, wall	Ground
Robot contact requirements $\sigma^{\boldsymbol{q}}$	$\emptyset$ (decided by $\boldsymbol{\pi}$)	Top	Antipodal	Grasp

state to stay close to the initial guess from demonstration if possible. $\mathcal{G}_i$ is defined as the ϵ-ball around the solution $\boldsymbol{x}$.

`retarget_q` find $\boldsymbol{q}$ given $\boldsymbol{x} \in {}^{\mathcal{E},\mathcal{O}}\sigma_i^{\boldsymbol{x}} \cap {}^{\mathcal{E},\mathcal{O}}\sigma_{i+1}^{\boldsymbol{x}}, \mathcal{E}, \mathcal{O}$, such that $(\boldsymbol{x}, \boldsymbol{q}) \in {}^{\mathcal{E},\mathcal{O}}\sigma_{i+1}$. This is summarized by the following IK feasibility problem:

$$\min_{\boldsymbol{q}} \ 0, \ \text{ s.t. } \boldsymbol{q} \in {}^{\mathcal{E},\mathcal{O}}\sigma_{i+1}^{\boldsymbol{q}|\boldsymbol{x}}. \tag{9}$$

The usage of these subroutines is summarized in Algorithm 1. Implementation of the state constraints in Equations 8 and 9 is described in Section IV-B.

D. Demonstration collection

The key information to be extracted to the demonstration are the object state trajectory $\{\bar{\boldsymbol{x}}_t\}_{t=0,\dots,T}$, the associated contact sequence $\{\sigma_t\}_{t=0,\dots,T}$, and the contact transitions $\{t_i\}_{i=1,\dots,N-1}, \bar{\boldsymbol{x}}_{t_i} \subset {}^{\bar{\mathcal{E}},\bar{\mathcal{O}}}\mathcal{X}_s$. Not needing $\bar{\boldsymbol{q}}$ explicitly allows great flexibility in how the demo is obtained. In this paper, we simply have a human manipulate the object. The top row of Fig. 1 shows an example of demonstrating *retrieval* on *cracker*. The human also provides the primitive label. Each demo takes fewer than 30 seconds to complete.

E. Policy composition and execution

Algorithm 1 summarizes an N-primitive demo is retargeted to the test time $\mathcal{E}, \mathcal{O}$ and object initial state $\boldsymbol{x}_0$. Each primitive is given a concrete objective using `retarget_x`. Upon completion of the current primitive and prior to executing the next primitive, we leverage `retarget_q` at each (freestanding) contact switch state to compute the robot-object contact for the next primitive. We assume the availability of an additional subroutine `move_robot_to`, which relocates the robot to a new state without collision. In our experiments, `move_robot_to` is implemented as a joint position controller, and the robot is always returned to a default configuration prior to moving to the next target q.

Algorithm 1 `compose_policy`

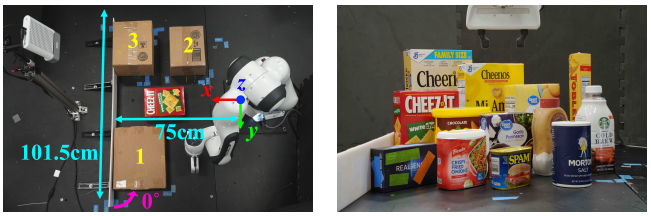
Input: $\mathcal{E}, \mathcal{O}, \mathcal{G}, \Pi, \Psi, \{\bar{\boldsymbol{x}}_t, \sigma_t\}, \{t_i\}, \boldsymbol{x}_0$

$\boldsymbol{G} \leftarrow \{\}$
$\{\tilde{\boldsymbol{x}}_i\} \leftarrow$ `remap_x`$(\{\bar{\boldsymbol{x}}_{t_i}\}, \boldsymbol{x}_0, \mathcal{E}, \mathcal{O})$
for $i = 0, \dots, N-1$ **do**
  $\boldsymbol{G}$.append(`retarget_x`$(\sigma_i, \sigma_{i+1}, \mathcal{E}, \mathcal{O}, \tilde{\boldsymbol{x}}_{i+1})$)
$\boldsymbol{G}$.append($\mathcal{G}$)
$\boldsymbol{x} \leftarrow \boldsymbol{x}_0$
for $i = 1, \dots, N$ **do**
  $\boldsymbol{q} \leftarrow$ `relocate_q`$(\boldsymbol{x}, \mathcal{O}, \mathcal{E})$
  $\boldsymbol{s} \triangleq (\boldsymbol{x}, \boldsymbol{q})$
  `move_robot_to`$(\boldsymbol{q})$
  while $\boldsymbol{x} \notin \boldsymbol{G}[i]$ **do**
    $\boldsymbol{s} \leftarrow {}^{\mathcal{E},\mathcal{O}}f((\boldsymbol{s}, \boldsymbol{\pi}_{\sigma_i}(\boldsymbol{s}_t; \mathcal{E}, \mathcal{O}, \mathcal{G}_i))$

V. System Setup

Our hardware setup is shown in Fig. 3. More details are available on our project website.

- **Robot.** A 2-finger Franka Hand mounted to a 7-DoF Franka Research 3 robot. The gripper has 1 DoF, leading to a 8-DoF action space. Deoxys [40] is used to control the robot system. We choose the Franka's base frame as the world frame and use it to define the environment configuration.
- **Environment.** A 10 cm tall, 101.5 cm long acrylic "wall" is erected at variable distances and orientations from the robot. The wall's pose is expressed with the world x position of its center and its yaw angle about the world z, where 0° is when the wall is parallel to y. Up to 3 obstacles may be mounted on the ground in the orientation in Fig. 3a. yuanpeThe dimensions of obstacles $1, 2, 3$ in cm are $(25.8, 30.8, 7.7), (18.5, 23.5, 14.0), (21.0, 25.5, 16.3)$. We express the position of an obstacle with the world (x, y) coordinate of its geometric center.
- **Objects.** 7 *standard* objects weighing $120g - 400g$ are tested on all tasks. 3 *short* objects are tested on the occluded grasping task. 3 *impossible* objects difficult to manipulate with the gripper are used solely to collect demonstrations in Section VI-B. A textured mesh of each object is captured with Kiri Engine for pose tracking, IK, and training the pushing primitive.



(a) Robot setup **(b)** Object set

Fig. 3: Hardware setup. Fig. 3a shows the world frame, the 3 obstacles, and the wall at $(75cm, 0^\circ)$. Fig. 3b show the 13 objects, from left to right beginning with the frontmost row: camera*, onion*, meat*, salt†; cracker, cocoa, seasoning, flapjack, coffee†; oat, cereal, wafer, chocolate†. *=*short* objects(3). †=*impossible* objects(3). The rest are *standard* objects(7).

• **Compute.** A desktop computer with Intel i9-13900K CPU and NVIDIA GeForce RTX 4090 GPU is used to run the pose estimation pipeline. A laptop computer with Intel i7-11800H CPU and NVIDIA GeForce RTX 3080 is used to run the primitives.
• **Perception and pose estimation.** A calibrated Microsoft Azure Kinect RGB-D camera is used to capture the scene. Textured meshes of the objects are used to get 6D pose estimations with Megapose [22].

VI. Experiments

A. Extrinsic manipulation tasks

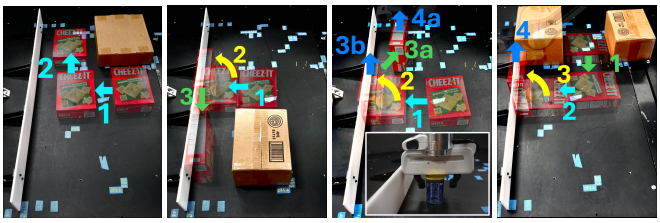
We evaluate our framework on 4 real-world extrinsic manipulation tasks, which are illustrated in Fig. 4.
• **Obstacle *avoidance***. *Push* the object forward, switch contact and *push* again to avoid the obstacle.
• **Object *storage***. *Push* an object toward the wall, *pivot* to align with an opening between the wall and the object, then *pull* it into the opening for storage.
• **Occluded *grasping***. *Push* the object in an ungraspable pose toward the wall, *pivot* it to expose a graspable edge, and *grasp* it. On *short* objects only, an additional *pull* is performed after *pivot* to create space between the wall and the object for inserting the gripper. To compare with [38], a simplified version with no *push* and $\boldsymbol{x}_0$ by the wall is also performed on *standard* objects.
• **Object *retrieval***. *Pull* the object from between two obstacles, *push* toward the wall, *pivot* it to expose a graspable edge, and *grasp* it.

Various environments are used for the demonstrations and tests to showcase our method's robustness against environment changes. All demonstrations are collected on *cracker*. Every task is evaluated on the 7 *standard* objects, each with 5 trials. Additionally, *occluded grasping* is evaluated on the 3 *short* with an extra *pull* step.

Numerical results and task parameters are summarized in Table II for *standard* object tasks. *Short* object *grasping* is summarized in Table III. Our method achieved an overall success rate of **80.5%** (**81.7%** for *standard* objects) in Section VI-A experiments. Despite not being tailored to *occluded grasping*, we outperformed the equivalent experiments in [38], both when the initial object state is against (**88.6%** vs. 78%) and away from (**77.1%** vs. 56%) the wall.

B. Retargeting from different demonstrations

To show that our method is agnostic to the specific demonstration, we collect demos for *grasping* on *oat* and the 3 *impossible* objects, whose shapes are unlikely to be graspable with a parallel jaw gripper. We then retarget all demos onto *cracker* from 5 different initial poses. We achieved **100%** success rate across 20 trials (Table III), showing that our method is capable of retargeting from a wide range of demos.



(a) Avoidance **(b)** Storage **(c)** Grasping **(d)** Retrieval

Fig. 4: Extrinsic manipulation tasks. The numbers and colors denote the primitive sequence. Push: cyan. Pull: green. Pivot: yellow. Grasp: blue. An additional "pull" is necessary for *short* objects, as the *a* and *b* branches illustrate in Fig. 4c.

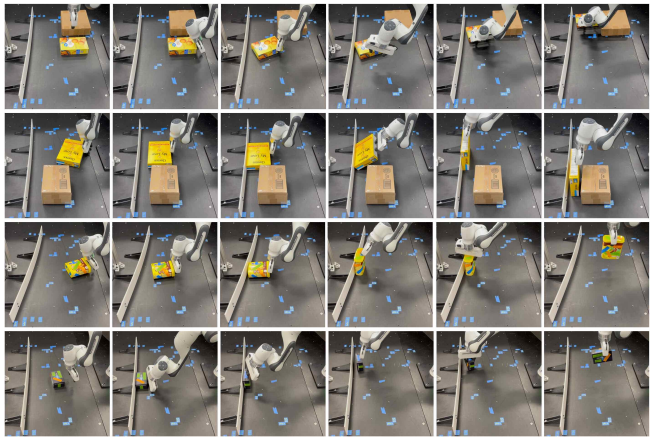


Fig. 5: Executing extrinsic manipulation tasks, in temporal order from left to right. From top row to bottom: *avoidance* on *wafer*, *storage* on *cereal*, *grasping* on *cocoa* and *camera*(*short* object). An extra *pull* (5th frame) is necessary to create clearance between the wall and gripper prior to grasping *camera*. Video is available in our supplementary material and on our website. Please refer to Table II for detailed task setups and Fig. 1 for the *retrieval* task.

TABLE III: Additional *grasping* experiments.

Short objects		Retargeting to *cracker*	
Primitives	Push-pivot-pull-grasp	Primitives	Push-pivot-grasp
Demo wall	80cm, 0°	Demo wall	75cm, 0°
Test wall	75cm, 0°	Test wall	77.5cm, 8.5°
Camera	3/5	Chocolate	5/5
Meat	4/5	Coffee	5/5
Onion	3/5	Oat	5/5
-	-	Salt	5/5
Overall	**66.7%**	Overall	**100%**

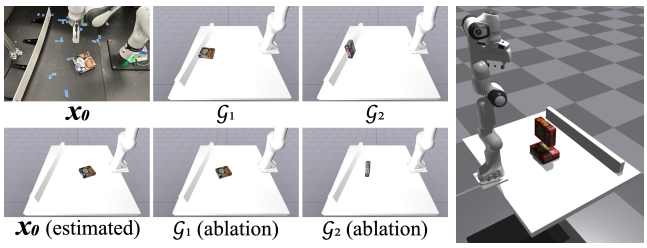
C. Ablation and comparison

The *occluded grasping* task (*push-pivot-grasp*) is chosen for our ablation study and comparison.
• **Ablation of `retarget_x`** shows the merit of contact retargeting. Here, $\tilde{\boldsymbol{x}}_i$ are directly set as $\mathcal{G}_i$. This drops the success rate to **37.1%** (Table II). Without `retarget_x`, the intermediate goals $\mathcal{G}_i$ seldom satisfy the contact requirements of the subsequent primitive. This is illustrated in Fig. 6a.
• **End-to-end reinforcement learning (RL)** serves as a comparison where contact information is not used and the task is treated as one long-horizon task. An RL agent is trained with proximal policy optimization [30] to move

TABLE II: Summary of experiments on 7 *standard* objects.

	Avoidance	Storage	Grasping		Retrieval	Grasping (Ablation)
Primitives	Push-push	Push-pivot-pull	Pivot-grasp	Push-pivot-grasp	Pull-push-pivot-grasp	Push-pivot-grasp
Demo wall	80cm, 0°	80cm, 0°	75cm, 0°	75cm, 0°	75cm, 0°	75cm, 0°
Test wall	80cm, 0°	75cm, 0°	75cm, 0°	77.5cm, $-8.5°$	80cm, 0°	77.5cm, $-8.5°$
Demo obstacles	$1:(-7.1, 14.6)$	$2:(10.8, 31.2)$	None	None	$2:(-19, 49), 3:(23.8, 33)$	None
Test obstacles	$1:(-7.1, 14.6)$	$2:(10.8, 31.2)$	None	None	$2:(-19, 54), 3:(23.8, 35.3)$	None
Cracker	4/5	5/5	4/5	4/5	4/5	2/5
Cereal	4/5	5/5	5/5	5/5	5/5	3/5
Cocoa	4/5	4/5	4/5	3/5	2/5	1/5
Flapjack	4/5	3/5	4/5	3/5	3/5	1/5
Oat	5/5	4/5	5/5	3/5	5/5	2/5
Seasoning	5/5	3/5	4/5	5/5	3/5	1/5
Wafer	4/5	5/5	5/5	4/5	4/5	3/5
Overall	**85.7%**	**82.9%**	**88.6%**	**77.1%**	**74.3%**	**37.1%**

cracker from one initial pose to a goal pose corresponding to a successful *grasping* execution on hardware (Fig. 6b). RL failed to learn meaningful actions using a reward function similar to the one we trained the *push* primitive with.



(a) Effect of `retarget_x` on $\mathcal{G}_i$ **(b)** RL scene

Fig. 6: Ablation and RL comparison. Fig. 6a shows the effects of ablating `retarget_x` away on the *grasping* task. The initial state $\boldsymbol{x}_0$, *push* and *pivot* intermediate goals $\mathcal{G}_1$, $\mathcal{G}_2$ are shown. $\mathcal{G}_1$, which is just $\tilde{\boldsymbol{x}}_1$ as computed by `remap_x` without `retarget_x`, is too far from the wall for *pivot*. Fig. 6b shows the scene, initial (on ground), and goal (floating) poses used to train RL.

D. Failure analysis

Despite not being a part of our contribution, the perception system and IK solver significantly affect our pipeline's success rate. Poor state estimation and $\mathcal{G}_i$ almost always result in failed execution. A better engineered solution can directly replace our implementation to boost performance. The failure rate of each primitive across experiments in Section VI-A are reported below. *Standard* (bold) and *short* objects are reported separately. *Push*: $\mathbf{7.5}\%, 0\%$. *Pull*: $\mathbf{4.5}\%, 18.2\%$. *Pivot*: $\mathbf{9.6}\%, 15.4\%$. *Grasp*: $\mathbf{6.6}\%, 10\%$. We attribute the higher failure rate on *short* objects to perception challenges and the small sizes. Occlusion from the wall is more significant when the object is shorter, and the pose estimation error may be larger. Furthermore, while action noises in our pipeline are of similar magnitude across objects, they also represent a relatively larger impact when the object is small.

VII. Conclusion and Future Work

This work presents a framework for generalizing long-horizon extrinsic manipulation from a single demonstration. Our method retargets the demonstration trajectory to the test scene by enforcing contact constraints with IK at every contact switches. The retargeted trajectory is then tracked with a sequence of short-horizon policies for each contact configuration. Our method achieved an overall success rate of 81.7% on real-world objects over 4 challenging long-horizon extrinsic manipulation tasks. Additional experiments show that contact retargeting is crucial to successfully retargeting such long-horizon plans, and a wide range of demonstration can be successfully retargeted with our pipeline. Future directions of this work include admitting language-based or simulation-based demonstrations, and generalizing the contact retargeting formulation to remove Assumption 3.

Acknowledgment

The authors would like to thank Stanford Robotics Lab for hardware support, Chen Wang for assistance with the perception pipeline, and NVIDIA Seattle Robotics Lab for insightful discussions.

References

[1] Bernardo Aceituno-Cabezas and Alberto Rodriguez. "A global quasi-dynamic model for contact-trajectory optimization in manipulation". In: *Robotics: Science and Systems*. 2020.

[2] Haonan Chen et al. "Predicting Object Interactions with Behavior Primitives: An Application in Stowing Tasks". In: *Conference on Robot Learning*. 2023.

[3] Sirui Chen, Albert Wu, and C. Karen Liu. "Synthesizing Dexterous Nonprehensile Pregrasp for Ungraspable Objects". In: *ACM SIGGRAPH 2023 Conf. Proc.*

[4] Yuanpei Chen et al. "Sequential Dexterity: Chaining Dexterous Policies for Long-Horizon Manipulation". In: *Conference on Robot Learning*. 2023.

[5] Xianyi Cheng et al. "Contact mode guided motion planning for quasidynamic dexterous manipulation in 3d". In: *2022 Int. Conf. Robot. Automat. (ICRA)*.

[6] Cheng Chi et al. "Diffusion Policy: Visuomotor Policy Learning via Action Diffusion". In: *The International Journal of Robotics Research* (2024).

[7] Cheng Chi et al. "Universal Manipulation Interface: In-The-Wild Robot Teaching Without In-The-Wild Robots". In: *Robotics: Science and Systems*. 2024.

[8] Yoonyoung Cho et al. “CORN: Contact-based Object Representation for Nonprehensile Manipulation of General Unseen Objects”. In: *The 12th International Conference on Learning Representations*. 2024.

[9] Marco Costanzo et al. “Grasp control for enhancing dexterity of parallel grippers”. In: *2020 IEEE Int. Conf. Robot. Automat. (ICRA)*.

[10] Nikhil Chavan Dafle et al. “Extrinsic dexterity: In-hand manipulation with external forces”. In: *2014 IEEE Int. Conf. Robot. Automat. (ICRA)*.

[11] Murtaza Dalal, Deepak Pathak, and Russ R Salakhutdinov. “Accelerating robotic reinforcement learning via parameterized action primitives”. In: *Advances in Neural Information Processing Systems* (2021).

[12] Neel Doshi, Orion Taylor, and Alberto Rodriguez. “Manipulation of unknown objects via contact configuration regulation”. In: *2022 Int. Conf. Robot. Automat. (ICRA)*.

[13] Clemens Eppner et al. “Exploitation of environmental constraints in human and robotic grasping”. In: *The International Journal of Robotics Research* (2015).

[14] Pete Florence et al. “Implicit behavioral cloning”. In: *Conference on Robot Learning*. 2022.

[15] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. “Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation”. In: *arXiv preprint arXiv:2401.02117* (2024).

[16] Tian Gao et al. “PRIME: Scaffolding Manipulation Tasks With Behavior Primitives for Data-Efficient Imitation Learning”. In: *IEEE Robotics and Automation Letters* (Oct. 2024), pp. 1–8.

[17] Caelan Reed Garrett et al. “Integrated task and motion planning”. In: *Annual review of control, robotics, and autonomous systems* (2021).

[18] Yifan Hou and Matthew T Mason. “Robust execution of contact-rich motion plans by hybrid force-velocity control”. In: *2019 Int. Conf. Robot. Automat. (ICRA)*.

[19] Edward Johns. “Coarse-to-fine imitation learning: Robot manipulation from a single demonstration”. In: *2021 IEEE Int. Conf. Robot. Automat. (ICRA)*.

[20] Oussama Khatib. “A unified approach for motion and force control of robot manipulators: The operational space formulation”. In: *IEEE Journal on Robotics and Automation* 3.1 (1987), pp. 43–53.

[21] Minchan Kim et al. “Pre-and post-contact policy decomposition for non-prehensile manipulation with zero-shot sim-to-real transfer”. In: *2023 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*.

[22] Yann Labbé et al. “MegaPose: 6D Pose Estimation of Novel Objects via Render & Compare”. In: *Conference on Robot Learning*. 2022.

[23] Viktor Makoviychuk et al. “Isaac Gym: High Performance GPU Based Physics Simulation For Robot Learning”. In: *35th Conf. Neural Inf. Process. Syst. Datasets and Benchmarks Track (Round 2)*.

[24] Ajay Mandlekar et al. “Learning to generalize across long-horizon tasks from human demonstrations”. In: *Robotics: Science and Systems*. 2020.

[25] Warwick Masson, Pravesh Ranchod, and George Konidaris. “Reinforcement learning with parameterized actions”. In: *Proceedings of the AAAI conference on artificial intelligence*. 2016.

[26] Soroush Nasiriany, Huihan Liu, and Yuke Zhu. “Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks”. In: *2022 Int. Conf. Robot. Automat. (ICRA)*.

[27] George Jose Pollayil et al. “Planning robotic manipulation with tight environment constraints”. In: *2021 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*.

[28] Mohamed Raessa et al. “Visually guided extrinsic manipulation for assembly tasks”. In: *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*.

[29] Harish Ravichandar et al. “Recent advances in robot learning from demonstration”. In: *Annual review of control, robotics, and autonomous systems* (2020).

[30] John Schulman et al. “Proximal policy optimization algorithms”. In: *arXiv preprint arXiv:1707.06347* (2017).

[31] Avi Singh et al. “Parrot: Data-Driven Behavioral Priors for Reinforcement Learning”. In: *International Conference on Learning Representations*. 2020.

[32] Sumedh Sontakke et al. “Roboclip: One demonstration is enough to learn robot policies”. In: *Advances in Neural Information Processing Systems* (2024).

[33] Simon Stepputtis, Yezhou Yang, and Heni Ben Amor. “Extrinsic dexterity through active slip control using deep predictive models”. In: *2018 IEEE Int. Conf. Robot. Automat. (ICRA)*.

[34] Zhaole Sun et al. “Learning pregrasp manipulation of objects from ungraspable poses”. In: *2020 IEEE Int. Conf. Robot. Automat. (ICRA)*.

[35] Russ Tedrake and the Drake Development Team. *Drake: Model-based design and verification for robotics*. 2019. URL: `https://drake.mit.edu`.

[36] Bowen Wen et al. “You Only Demonstrate Once: Category-Level Manipulation from Single Visual Demonstration”. In: *Robotics: Science and Systems*. 2022.

[37] Shih-Min Yang et al. “Learning Extrinsic Dexterity with Parameterized Manipulation Primitives”. In: *2024 IEEE Int. Conf. Robot. Automat. (ICRA)*.

[38] Wenxuan Zhou and David Held. “Learning to grasp the ungraspable with emergent extrinsic dexterity”. In: *Conference on Robot Learning*. 2023.

[39] Wenxuan Zhou et al. “HACMan: Learning hybrid actor-critic maps for 6D non-prehensile manipulation”. In: *Conference on Robot Learning*. 2023.

[40] Yifeng Zhu et al. “Viola: Imitation learning for vision-based manipulation with object proposal priors”. In: *Conference on Robot Learning*. 2023.