MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Sim-to-Real Contact-Rich Pivoting via Optimization-Guided RL with Vision and Touch

Shirai, Yuki; Ota, Kei; Jha, Devesh K.; Romeres, Diego

TR2025-169 December 10, 2025

Abstract

Non-prehensile manipulation is challenging due to complex contact interactions between objects, the environment, and robots. Model-based approaches can effi- ciently generate complex trajectories of robots and objects under contact constraints. However, they tend to be sensitive to model inaccuracies and require access to privileged information (e.g., object mass, size, pose), making them less suitable for novel objects. In contrast, learning-based approaches are typically more robust to modeling errors but require large amounts of data. In this paper, we bridge these two approaches to propose a framework for learning closed-loop pivoting manipulation. By leveraging computationally efficient Contact-Implicit Trajec- tory Optimization (CITO), we design demonstration-guided deep Reinforcement Learning (RL), leading to sample-efficient learning. We also present a sim-to-real transfer approach using a privileged training strategy, enabling the robot to perform pivoting manipulation using only proprioception, vision, and force sensing without access to privileged information. Our method is evaluated on several pivoting tasks, demonstrating that it can successfully perform sim-to-real transfer.

Embodied World Models for Decision Making, NeurIPS Workshop 2025

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Sim-to-Real Contact-Rich Pivoting via Optimization-Guided RL with Vision and Touch

Yuki Shirai
Mitsubishi Electric Research Laboratories
shirai@merl.com

Kei Ota
Mitsubishi Electric
ota.kei@ds.mitsubishielectric.co.jp

Devesh Jha
Mitsubishi Electric Research Laboratories
jha@merl.com

Diego Romeres
Mitsubishi Electric Research Laboratories
romeres@merl.com

Abstract

Non-prehensile manipulation is challenging due to complex contact interactions between objects, the environment, and robots. Model-based approaches can efficiently generate complex trajectories of robots and objects under contact constraints. However, they tend to be sensitive to model inaccuracies and require access to privileged information (e.g., object mass, size, pose), making them less suitable for novel objects. In contrast, learning-based approaches are typically more robust to modeling errors but require large amounts of data. In this paper, we bridge these two approaches to propose a framework for learning closed-loop pivoting manipulation. By leveraging computationally efficient Contact-Implicit Trajectory Optimization (CITO), we design demonstration-guided deep Reinforcement Learning (RL), leading to sample-efficient learning. We also present a sim-to-real transfer approach using a privileged training strategy, enabling the robot to perform pivoting manipulation using only proprioception, vision, and force sensing without access to privileged information. Our method is evaluated on several pivoting tasks, demonstrating that it can successfully perform sim-to-real transfer.

1 Introduction

Non-prehensile manipulation, such as pivoting, pushing, and sliding, plays an important role in enhancing the dexterity of robotic systems [11, 40, 51]. These skills allow robots to interact with the environment more flexibly, enabling them to adapt to a wide range of tasks without requiring secure grasps. However, achieving such skills is challenging due to the inherently complex contact interactions (e.g., making-breaking contact, sliding-sticking contact). These interactions introduce non-smooth dynamics that are difficult to model and control as the number of contacts increases.

Model-based optimization methods, such as CITO and Model Predictive Control (MPC) [60, 47, 36, 42, 67, 53], have demonstrated impressive performance, particularly in generating diverse trajectories at low computational cost. However, since these methods, in general, rely on simplified models of manipulation, they can be highly sensitive to uncertainties due to model inaccuracies. More critically, they often rely on offline system identification or online estimation of privileged information, such as object properties or contact states. This dependency limits the applicability of model-based controllers, particularly in real-world scenarios involving novel objects or partially observable environments.

Learning-based methods, such as RL, have also shown impressive performance, especially in their robustness against various sources of uncertainty [38, 50, 10, 37, 4, 27, 49]. These methods can

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Embodied World Models for Decision Making.

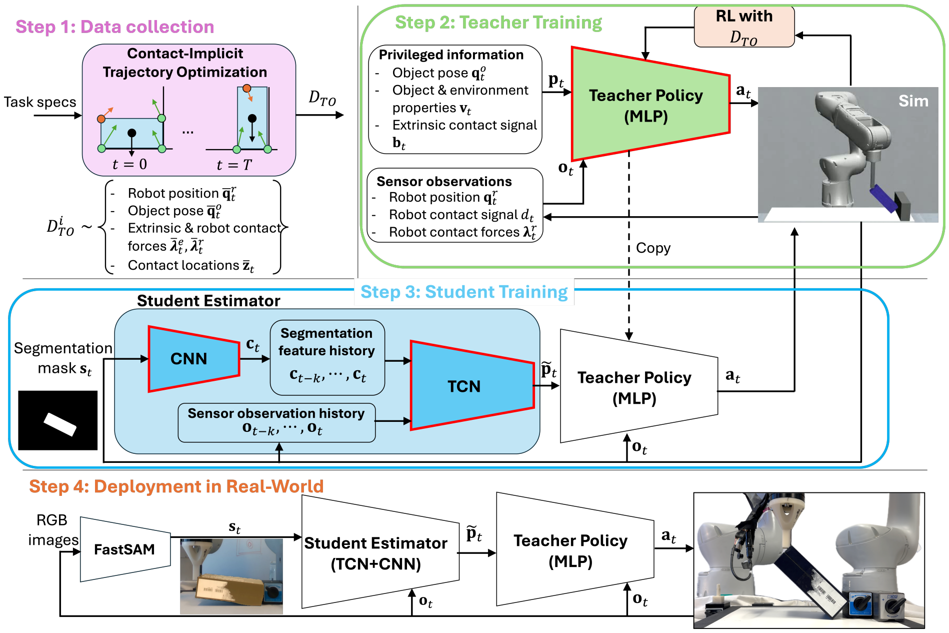

Step 1: Data collection
Contact-Implicit Trajectory Optimization
Task specs
$t = 0$
$t = T$
D_{TO}
$D_{TO}^i \sim$
Robot position $\bar{\mathbf{q}}_t^r$
Object pose $\bar{\mathbf{q}}_t^o$
Extrinsic & robot contact forces $\bar{\boldsymbol{\lambda}}_t^e, \bar{\boldsymbol{\lambda}}_t^r$
Contact locations $\bar{\mathbf{z}}_t$
Step 2: Teacher Training
RL with D_{TO}
Privileged information
Object pose $\mathbf{q}_t^o$
Object & environment properties $\mathbf{v}_t$
Extrinsic contact signal $\mathbf{b}_t$
$\mathbf{p}_t$
Teacher Policy (MLP)
$\mathbf{a}_t$
Sim
$\mathbf{o}_t$
Sensor observations
Robot position $\mathbf{q}_t^r$
Robot contact signal d_t
Robot contact forces $\boldsymbol{\lambda}_t^r$
Copy
Step 3: Student Training
Student Estimator
Segmentation mask $\mathbf{s}_t$
CNN
$\mathbf{c}_t$
Segmentation feature history
$\mathbf{c}_{t-k}, \cdots, \mathbf{c}_t$
Sensor observation history
$\mathbf{o}_{t-k}, \cdots, \mathbf{o}_t$
TCN
$\tilde{\mathbf{p}}_t$
Teacher Policy (MLP)
$\mathbf{a}_t$
$\mathbf{o}_t$
Step 4: Deployment in Real-World
RGB images
FastSAM
$\mathbf{s}_t$
Student Estimator (TCN+CNN)
$\tilde{\mathbf{p}}_t$
Teacher Policy (MLP)
$\mathbf{a}_t$
$\mathbf{o}_t$
$\mathbf{o}_t$

Figure 1: Overview of our proposed framework. Trainable modules have red edges. **Step 1**: We collect data using CITO given a user-specified task. **Step 2**: The teacher policy is trained using RL with privileged information and sensor observations, leveraging the demonstrations collected in Step 1. **Step 3**: The student estimator is trained to estimate the privileged information. The estimator consists of a CNN and a TCN to process temporal sensor observations, including segmentation and force measurements. **Step 4**: During deployment in real-world, the learned student estimator and teacher policy run in zero-shot sim-to-real transfer on physical hardware.

operate without privileged information by directly learning policies from raw observations. However, they typically require a large number of training samples, resulting in long training times, which poses a significant challenge for practical deployment. This is especially problematic in non-prehensile manipulation, where the policy must reason object pose, contact locations, contact forces, and feasible action spaces from indirect and partial observations. Unlike prehensile manipulation (e.g., grasping [40]), where grasping provides stable control, non-prehensile tasks often involve underactuated dynamics and complex contact constraints that make the learning problem significantly harder. As a result, RL may often fail to discover viable solutions within a reasonable training time.

In this paper, we propose a framework that integrates the strengths of model-based planning with learning-based policy execution for non-prehensile pivoting manipulation. As illustrated in Fig. 1, we first employ CITO to collect a large number of task demonstrations across a range of privileged information parameters. Second, a teacher policy is trained in a simulator using RL, leveraging the demonstrations (e.g., robot, object, & contact trajectories) generated by CITO. As a result, the teacher policy achieves significantly higher sample efficiency. Third, we train a student estimator to predict the privileged information required by the teacher policy from observations. Finally, we evaluate the trained policy over various baselines in both simulation and hardware experiments, achieving zero-shot sim-to-real transfer. Our contributions are as follows.

- A framework for learning contact-rich non-prehensile manipulation controllers and estimators by leveraging demonstrations generated by CITO.
- A sim-to-real transfer approach based on a student-teacher architecture, where the student estimates privileged information from partial observations using a temporal history of visual and force sensing.

2 Related Work

Model-Based Optimization for Contact-Rich Manipulation. Model-based optimization methods have successfully achieved various non-prehensile manipulation skills, such as pushing [42, 47, 7], pivoting [3, 57, 54], and pulling [29, 32]. These methods design manipulation skills computationally efficiently by leveraging techniques such as contact smoothing [47, 59], mixed-integer convex optimization [3, 28], and distributed optimization [7, 55]. However, these methods typically require privileged information (i.e., full-state feedback). For example, [6] assumes that contact forces in extrinsic contacts between the object and the environment are directly measurable, which becomes increasingly impractical as task complexity grows. In this paper, we relax the full-state feedback assumptions by adopting an RL approach, while still leveraging CITO to generate a large number of demonstrations. This strategy enables the agent to learn manipulation skills significantly more efficiently than standard RL methods that rely solely on sparse rewards.

Learning-Based Methods for Contact-Rich Manipulation. Learning-based methods, such as RL, Imitation Learning (IL), and foundation model-based methods, have demonstrated remarkable success in robotic manipulation [24, 16, 17, 20, 12, 63, 69, 39, 43, 52], enabling complex tasks such as bimanual cable manipulation and folding laundry. However, all of these methods require a large number of training samples, resulting in prohibitively long training times.

To improve sample efficiency, demonstration-guided RL has been studied [66, 50, 5], where the demonstrations are used to guide exploration of RL agent to learn the policy and improve sample efficiency. For example, [46] uses Rapidly-exploring Random Trees (RRT) and [68] uses human videos for generating kinematically feasible demonstrations for manipulation. However, these works [46, 68, 72] only consider kinematically feasible demonstrations. Incorporating contact force information into demonstrations could be critical to learn fine manipulation due to very thin margins of error imposed by contact constraints. Although some works have explored dynamically feasible demonstrations in locomotion tasks [21, 61, 13], there has been relatively little work on applying such demonstrations to manipulation tasks. This is due to the lack of a reliable module for generating dynamically feasible demonstrations considering extrinsic contact states in manipulation. Some works [30, 14] leverage human demonstrations to capture contact forces, but collecting such data at scale is challenging and often requires significant manual effort. In contrast, we use CITO to automatically generate robot, object, and contact force trajectories, providing richer supervision and greater scalability than human demonstrations.

Bridging the sim-to-real gap is another key challenge. Privileged information used during training is often unavailable during real-world deployment. Some prior works reconstruct privileged states using external sensors [61, 13], such as AprilTags [44, 19]. The recent advances in the student-teacher framework [15, 35, 16, 41, 33, 62, 9, 31] enable zero-shot sim-to-real transfer by learning to predict privileged information. Although some works have applied the student-teacher framework to manipulation, they often rely on restrictive assumptions (e.g., assuming that object size remains constant [22, 19]). In contrast, although we also adopt a student-teacher framework, we do not rely on such assumptions. By using a temporal history of force measurements and segmentation images, our student estimator is more broadly applicable to real-world scenarios involving novel objects.

3 Method

In this section, we present our proposed framework, as shown in Fig. 1. The objective is to learn pivoting manipulation using only proprioceptive, visual, and force sensing. The proposed framework consists of three steps. In Step 1, task demonstrations are generated using CITO. In Step 2, a teacher policy which has access to the privileged information is trained using RL with the sampled demonstrations collected in Step 1. In Step 3, a student estimator is trained to estimate the privileged information, which serves as input to the teacher policy. The teacher policy with the predictions of the trained student estimator is ultimately deployed on physical hardware for real-world validation.

In this work, we make the following assumptions: (1) both the objects and the robots are rigid and the center of mass is located at the geometric centers, (2) manipulation occurs under quasi-static condition in SE(2), and (3) the robot end-effector pose, camera sensing, and robot contact force measurements are consistently available throughout manipulation.

3.1 Step 1: Collecting Demonstrations Using Contact-Implicit Trajectory Optimization

We collect a large set of datasets using CITO in [58]. For N_r robots, we consider the following CITO:

$$\begin{aligned} &\min_{\bar{\mathbf{q}}_t,\dot{\bar{\mathbf{q}}}_t,\bar{\mathbf{y}}_t} \sum_{t=0}^{T} \|\bar{\mathbf{q}}_t - \bar{\mathbf{q}}_t^{\text{ref}}\|_Q^2 \\ &\text{s. t. }, f_{\text{dyn}}\left(\bar{\mathbf{q}}_t, \bar{\mathbf{q}}_{t+1}, \dot{\bar{\mathbf{q}}}_t, \bar{\mathbf{y}}_t\right) = \mathbf{0}, g\left(\bar{\mathbf{q}}_t, \dot{\bar{\mathbf{q}}}_t, \bar{\mathbf{y}}_t\right) = \mathbf{0}, \end{aligned} \tag{1}$$

where $\bar{\mathbf{q}}_t := [\bar{\mathbf{q}}_t^o, \bar{\mathbf{q}}_t^r]$ and $\bar{\mathbf{y}}_t := [\bar{\boldsymbol{\lambda}}_t^e, \bar{\boldsymbol{\lambda}}_t^r, \bar{\mathbf{z}}_t]$. $\bar{\mathbf{q}}_t^o \in \mathbb{R}^3$ represent an object pose in SE(2) and $\bar{\mathbf{q}}_t^r \in \mathbb{R}^{2\times N_r}$ represent robot end-effector positions in SE(2), respectively. The end-effector orientation is kept fixed throughout the task. $\bar{\boldsymbol{\lambda}}_t^e \in \mathbb{R}^{2\times N_e}$ and $\bar{\boldsymbol{\lambda}}_t^r \in \mathbb{R}^{2\times N_r}$ represent contact forces between an object and the environment, and between an object and the robots, respectively. N_e represents the potential extrinsic number of contacts between the object and the environment. We denote $\bar{\mathbf{z}}_t \in \mathbb{R}^{2\times N_e}$ as the extrinsic contact location between the object and the environment. $\bar{\mathbf{q}}_t^{\text{ref}} \in \mathbb{R}^3$ represents the linear interpolation between the start and goal object pose with T steps. We use the subscript t to represent the timestep t. We denote f_{dyn} as non-smooth dynamics of the non-prehensile manipulation, including nonsmooth contact switching, force and moment balance, and friction cone constraints. We denote g as non-dynamics related constraints, such as bounds of decision variables and collision-avoidance. We emphasize that the generation of trajectories that satisfy kinematic feasibility alone and not dynamic feasibility are simple to obtain by removing some of the f_{dyn} constraints, such as force and moment balance constraints. Thus, we denote kinematically feasible dynamics as f_{kin}. The problem (1) is solved using solvers such as Gurobi [25] and SNOPT [23]. See [58] and the appendix for more details. Solving (1) generates N demonstrations $D_{\text{TO}} := \{D_{\text{TO}}^i\}_{i=1}^N$, where $D_{\text{TO}}^i := \{\{\bar{\mathbf{q}}_t\}_{t=0}^T, \{\bar{\mathbf{y}}_t\}_{t=0}^T\}^i$. While previous works (e.g., [46, 68, 72]) only consider $\bar{\mathbf{q}}_t$ with f_{kin}, this work explicitly considers $\bar{\mathbf{q}}_t$ and $\bar{\mathbf{y}}_t$ with f_{dyn}. In particular, $\bar{\boldsymbol{\lambda}}_t^r$ guides for agents to learn robot motion direction, while $\bar{\boldsymbol{\lambda}}_t^e$ and $\bar{\mathbf{z}}_t$ offer insights into preferred extrinsic contacts.

3.2 Step 2: Learning Privileged Teacher-Policy

In this step, a teacher policy is trained to achieve the desired pivoting manipulation in a simulation where privileged information is accessible. We formulate the problem as a Markov Decision Process (MDP), with each component defined as follows.

States. States consist of the privileged and non-privileged information. The privileged information $\mathbf{p}_t$ includes the object pose $\mathbf{q}_t^o \in \mathbb{R}^3$, the object and environment properties $\mathbf{v}_t \in \mathbb{R}^{N_p}$, and the extrinsic contact signal $\mathbf{b}_t \in \mathbb{Z}^{N_e}$. The object pose $\mathbf{q}_t^o$ lies in the SE(2) and consists of two positional components and one orientation. $\mathbf{v}_t$ encodes physical properties, which are the mass and size of the object, and the friction constants of both the object and the surrounding environment geometry. The extrinsic contact signal $\mathbf{b}_t$ is a binary vector where each element indicates whether a specific face of the object is in contact with a predefined environment surface (e.g., wall, table).

The non-privileged information $\mathbf{o}_t$ consists of the robot positions $\mathbf{q}_t^r \in \mathbb{R}^{2\times N_r}$, the binary robot contact signal $d_t \in \mathbb{Z}^1$, and the 2D contact forces $\boldsymbol{\lambda}_t^r \in \mathbb{R}^{2\times N_r}$ measured by force sensors mounted on the robots' wrists. All observations are approximately normalized to lie in the range $[-1, 1]$.

Actions. We consider linear translational actions in SE(2) for each robot, denoted as $\mathbf{a}_t^{2\times N_r}$. Specifically, each action represents a relative position command for the robots' end-effector. These action commands are converted into joint torques using Operational Space Control (OSC) [34].

Rewards. Based on how demonstrations are used, we consider three distinct reward formulations. We denote three different RL polices using different demonstrations (i.e., using different reward formulation) as (1) *Vanilla RL*, which does not use any demonstrations, (2) *Kinematics-conditioned RL*, and (3) *Dynamics-conditioned RL*. These policies are obtained by 3 different rewards defined as:

$$\begin{aligned} \text{(1) Vanilla RL:} \quad & r = r_p + r_s + r_a \\ \text{(2) Kinematics-conditioned RL:} \quad & r = r_p + r_s + r_a + r_{\text{kin}} \\ \text{(3) Dynamics-conditioned RL:} \quad & r = r_p + r_s + r_a + r_{\text{dyn}} \end{aligned} \tag{2}$$

First, the progress reward is $r_p = \alpha_1\left(\frac{\pi}{2} - \theta_e\right) + \alpha_2\left(\theta_e^2\right)$, where $\theta_e = \arccos\left(\frac{1}{2}\left(Tr\left(R^{\text{G}}R\right) - 1\right)\right)$. $Tr(\cdot)$ denotes the matrix trace, and R and R^{G} are the goal and current rotation matrices, respectively.

θ_e measures the angular deviation between the current and goal orientations, and $\frac{\pi}{2}$ is added as the offset. While the linear term in r_p is used in [73, 74], our experiments reveal that the inclusion of the quadratic term is necessary to achieve higher success rates under domain randomization (DR) [64] over the size of the objects, which was not discussed in [74]. Second, the sparse success reward is defined as $r_s = \alpha_3 \mathbb{I}_G \left(\mathbf{q}_t^o\right)$, where $\mathbb{I}_G$ is the indicator function over the goal set $G := \left\{\mathbf{q}_t^o \in \mathbb{R}^3 \mid \|\mathbf{q}_t^o - \mathbf{q}_{\text{goal}}^o\| \leq \epsilon_s\right\}$, where $\mathbf{q}_{\text{goal}}^o$ is the desired goal state of the object and ϵ_s is the user-specified positive constant. Third, the action smoothness reward is given by $r_a = \alpha_4 \|\mathbf{a}_{t-1} - \mathbf{a}_t\|^2$, for avoiding non-smooth actions.

Next, we define the reward based on demonstrations generated by CITO. For the kinematic reward r_{kin}, we use object and robot poses $\bar{\mathbf{q}}_t$ and extrinsic contact locations $\bar{\mathbf{z}}_t$ obtained by solving (1) with f_{kin}. Note that contact force demonstrations are not available in this setting, as f_{kin} does not have dynamics constraints. Thus, we compute r_{kin} as:

$$r_{\text{kin}} = \alpha_5 ||\mathbf{q}_t^r - \phi(\mathbf{q}_t^o)||^2 \tag{3}$$

where ϕ retrieves the closest reference robot configuration $\bar{\mathbf{q}}_t^r$ corresponding to the current object observation $\mathbf{q}_t^o$. Since both the object and environment parameters are sampled from a known dataset D_{TO} during simulation, the corresponding object reference trajectory $\bar{\mathbf{q}}_t^o$ is known. Using the current observation, we identify the closest object configuration within this trajectory, and consequently retrieve the closest robot configuration. This reward term encourages the robot to follow the kinematically feasible behaviors.

Similarly, we define the dynamics reward r_{dyn} by utilizing the demonstration $\bar{\mathbf{q}}_t$ and $\bar{\mathbf{y}}_t$ obtained by solving (1) with the dynamics model f_{dyn}:

$$r_{\text{dyn}} = \alpha_6 ||\mathbf{q}_t^r - \phi(\mathbf{q}_t^o)||^2 + \alpha_7 \arccos\left(\frac{\boldsymbol{\lambda}_t^r \cdot \psi(\mathbf{q}_t^o)}{||\boldsymbol{\lambda}_t^r|| ||\psi(\mathbf{q}_t^o)||}\right) + \alpha_8 \mathbf{b}_t \tag{4}$$

where ψ retrieves the closest reference robot contact forces $\bar{\boldsymbol{\lambda}}_t^r$ corresponding to $\mathbf{q}_t^o$, following the same logic as ϕ. This reward encourages the robot to follow the dynamically feasible behaviors. In particular, the arccosine term in r_{dyn} encourages the robot to perform a similar contact force direction as the demonstration shows. Importantly, we do not enforce matching the magnitude of the contact force, as we observe significant discrepancies between the dynamics model by f_{dyn} and those in simulators (e.g., MuJoCo), leading to a potential sim2sim gap in contact force magnitudes. Hence, this work focuses on the direction of contact forces. The term $\mathbf{b}_t$ is used to count if the desired extrinsic contact states occur. The constants $\alpha_{i=1,2,3,8}$ are positive and the others are negative.

3.3 Step 3: Learning Student-Estimator

The objective of this step is to train the student estimator only using sensor observations to predict the privileged information as shown in Fig. 1. We empirically observe that sensor observations alone are sufficient for the object whose geometry is in-distribution with the dataset. However, their reliability declines when there is uncertainty in object size, which is quite common when manipulating novel objects. To address this, we additionally incorporate vision inputs to improve the estimation of the privileged information. Directly using RGB images is avoided due to potential noise, and employing 3D point clouds is excluded due to their significant computational cost (see [16]). Instead, we leverage the object segmentation $\mathbf{s}_t$ derived from the RGB image, providing a compact but informative representation of the object.

Therefore, we define a student encoder that takes the history of the sensor observations, $[\mathbf{o}_{t-T}, \cdots, \mathbf{o}_t]$, and the history of the segmentation features, $[\mathbf{s}_{t-T}, \cdots, \mathbf{s}_t]$. Since $\mathbf{s}_t$ is high-dimensional, we first apply a Convolutional Neural Network (CNN) to compress the segmentation into a lower-dimensional feature representation $\mathbf{c}_t$. Using the temporal histories of $\mathbf{o}_t$ and $\mathbf{c}_t$, we use a Temporal Convolutional Network (TCN) [8] to estimate the privileged information. We train CNN and TCN jointly via supervised learning using datasets collected by rolling out the teacher policy in the simulator under domain randomization. The supervised learning objective is to minimize the following loss function:

$$l = ||\mathbf{p}_t - \tilde{\mathbf{p}}_t||^2, \tag{5}$$

where $\mathbf{p}_t$ is the ground-truth privileged information and $\tilde{\mathbf{p}}_t$ is the estimated output from the student encoder. It is worth noting that we do not initialize the history buffer with zeros at the beginning of the episode as other works do (e.g., [22, 48, 71]). Instead, we populate the buffer by repeating the initial observation and include this initialization scheme in the supervised learning dataset, which was critical for the student estimator to achieve accurate performance.

4 Experiment Setup

We validate our framework across two distinct tasks (see Fig. 5): **Pivoting with Wall**: pivoting a box using an external wall, **Pivoting without Wall**: pivoting a box without relying on external support. For the latter task, the table surface must provide very high friction. In simulation, increasing the friction coefficient alone was insufficient to replicate the real-world behavior. As a workaround, we add a thin virtual wall of height $1\,\mathrm{mm}$ to simulate the effect of high-friction contact (see Fig. 5b). In the hardware experiments, we test on a variety of previously unseen objects (see Fig. 6) to assess generalization beyond the training set. We define a trial as successful if the final orientation error satisfies $|\theta_e| \leq 0.087$ rad (i.e., $5°$). We describe the setup for each module below, with additional details provided in the appendix.

Demonstration Setup. We use the method proposed in [58], randomizing object and environment parameters to generate diverse demonstrations. For all tasks, we randomize the mass of the object, the friction constant of the object and the environment, and the size of the object. For each task, we collect 5000 demonstrations, which can be computed within a few minutes using 30 Intel i9-13900K CPU cores.

Teacher Policy Setup. We train the teacher policy in MuJoCo simulator [65] using robosuite [76] as a wrapper. The agent is trained using Soft Actor Critic (SAC) [26], implemented with tf2rl [45]. For SAC, we use Multi-Layer Perceptron (MLP) for both actor and critic networks. The simulation runs at 500 Hz, while the policy operates at 10 Hz. For each episode, we set the maximum episode length to 300 steps. Overall, training converges within 4 hours on a single NVIDIA RTX 4090. During training, we apply domain randomization over the objects' mass and size, the friction constants of both the object and the environment, and the controller gains used in OSC within robosuite. Furthermore, we introduce sensor noise to both privileged information and sensor observations to account for the estimation errors from the student estimator during deployment.

Student Estimator Setup. We first rollout the trained teacher policy over 2000 episodes and collect a dataset containing ground-truth privileged information, sensor observations, and corresponding segmentation images (640×480 resolution) of the object using MuJoCo's rendering functionality. During data collection, we augment the segmentation images by introducing noise, such as randomly flipping, translating, and rotating segmentation masks, to improve robustness. We then train the student estimator via behavior cloning, minimizing the loss function (5) over multiple epochs. We use $T = 5$ step history of the observations for training corresponding to 0.5 second. Overall, training converges with 10 epochs (1 hour roughly), depending on the range of domain randomization.

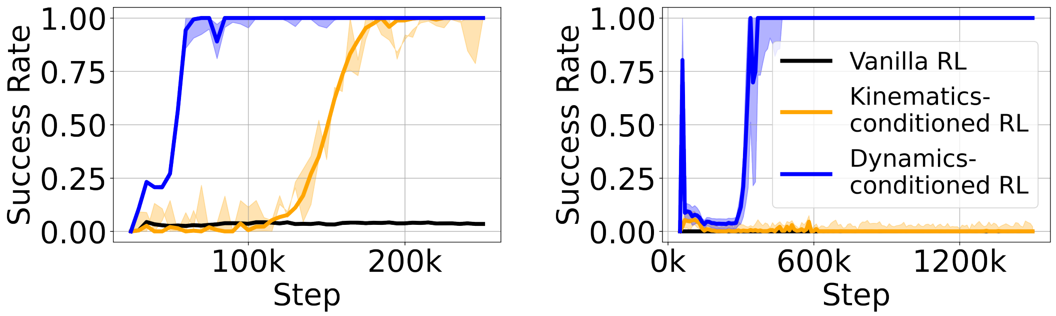
Hardware Setup. We use a 6 DoF MELFA robot [2] equipped with a stiffness controller and a 6-axis force/torque sensor. This hardware enables users to get robot end-effector positions and the force measurements in the world frame. For object segmentation, we use FastSAM [75] to generate multiple instance segmentations from an RGB image captured by an Intel RealSense D435 RGB-D camera [1]. To identify the target object, we filter the segmented instances under their corresponding point cloud information, under the assumption that a rough estimate of the SE(2) plane is available, as we focus exclusively on SE(2) planar manipulation.

Baselines. We implement an MPC baseline that uses privileged information, including object mass, size, and friction (identified offline), and object pose (estimated via AprilTags). At each timestep, MPC solves (1) in a receding-horizon manner, running at the same frequency as the teacher policy.

5 Results

Throughout our experiments, we aim to address the following questions:

1. Do demonstrations generated by CITO facilitate more effective and efficient learning?
2. How does the teacher policy's performance vary with different demonstrations?
3. How robust is the teacher policy compared to a baseline model-based method?



(a) With external wall

(b) Without external wall

Figure 2: Learning curves for different RL training runs. Solid lines indicate average success rates, and shaded regions denote standard deviation across three different random seeds. Every 10k step, the current policy is evaluated over 50 episodes, and the success rate is plotted.

Table 1: Number of successful attempts in real.

Mass	Kinematics-conditioned RL	Dynamics-conditioned RL
$50\,\mathrm{g}$	2 / 5	**5** / 5
$110\,\mathrm{g}$	**5** / 5	**5** / 5
$300\,\mathrm{g}$	0 / 5	**5** / 5

4. How accurately can the student estimator predict the privileged information?
5. Can the trained policies be successfully transferred to real-world hardware experiments?

Do demonstrations generated by CITO facilitate learning? Across the two tasks, we compare RL performance using different types of demonstrations, corresponding to the different reward formulations in (2). RL with kinematic demonstrations is comparable to prior works such as [46, 68], which only consider kinematically feasible trajectories. Overall, RL with dynamics-based demonstrations achieves the fastest learning as shown in Fig. 2. In particular, in the pivoting without external wall task, neither vanilla RL nor kinematics-conditioned RL was able to learn the skill. We attribute this to the task's tighter feasible action space. In contrast, dynamics-conditioned RL successfully learns the skill, benefiting from enriched demonstration with contact information.

How does the teacher policy's performance vary with different demonstrations? For the pivoting-with-wall task, we deploy both kinematics- and dynamics-conditioned RL policies on a real system using a box of mass $110\,\mathrm{g}$. During deployment, we vary the mass values used as privileged information. Table 1 shows the success rates over three trials. We observe that dynamics-conditioned RL consistently outperforms kinematics-conditioned RL. While both policies are trained with access to privileged information, the dynamics-conditioned policy benefits from demonstrations that include contact force references. This enables the policy to learn physically grounded interaction behaviors during training, leading to greater robustness against variations in dynamic properties. In contrast, the kinematics-conditioned policy is trained with demonstrations that satisfy only geometric feasibility, making it more sensitive to changes in object properties. These results highlight the importance of dynamics-aware demonstrations in contact-rich manipulation tasks.

How robust is the learned policy compared to MPC? We compare the robustness of a dynamics-conditioned RL policy against an MPC controller on the real-world pivoting-with-wall task. The true object length is $0.16\,\mathrm{m}$, and we introduce intentional mismatches in the assumed object length during deployment. For example, a $-5\,\mathrm{mm}$ offset means that the actual size of the box is shorter than what the controllers expect. As shown in Table 2, both MPC and RL succeed when the actual object is longer than expected ($+5\,\mathrm{mm}$), as the contact with the wall is still maintained. However, when the actual object is shorter than expected ($-5\,\mathrm{mm}$), MPC fails completely, while RL remains successful. This suggests that the learned policy exhibits greater tolerance to moderate discrepancies in privileged information. At larger mismatches ($-10\,\mathrm{mm}$), even RL fails. These results highlight the importance of accurate privileged information during deployment and motivate us to develop reliable estimators.

Table 2: Number of successful attempts in real.

	MPC	Dynamics-conditioned RL
$+5\,\mathrm{mm}$	5 / 5	5 / 5
$-5\,\mathrm{mm}$	0 / 5	5 / 5
$-10\,\mathrm{mm}$	0 / 5	0 / 5

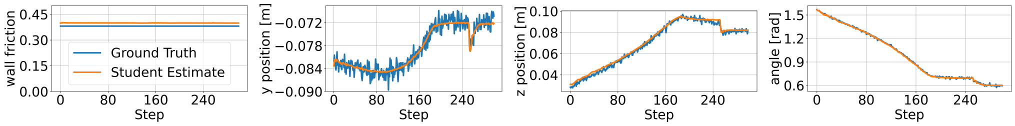


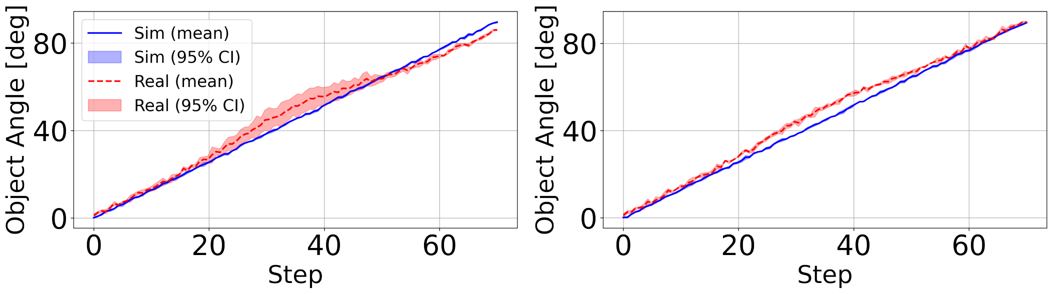
Figure 3: Comparison of our student estimator's predictions and the ground truth for the wall friction constant, y- and z-position of the object, and orientation along x-axis, for the pivoting with a wall.

How accurately can the student estimator predict the privileged information? We deploy the trained student estimator and the teacher policy in MuJoCo and collect both the ground-truth privileged information and the corresponding student estimator's predictions. Representative results are shown in Fig. 3, demonstrating that our student estimator can successfully predict the privileged information with reasonable accuracy.

Hardware Experiments. We deploy our teacher policy and student estimator on the real robot using zero-shot sim-to-real transfer. Overall, the policy successfully completes the desired task without access to privileged information as shown in Fig. 5 and Fig. 6.

Sim-to-Real Transfer. To evaluate sim-to-real transfer, we deploy the learned dynamics-conditioned RL policy on both the simulation and physical hardware for the two pivoting tasks. The resulting object orientation trajectories over three trials are shown in Fig. 4.

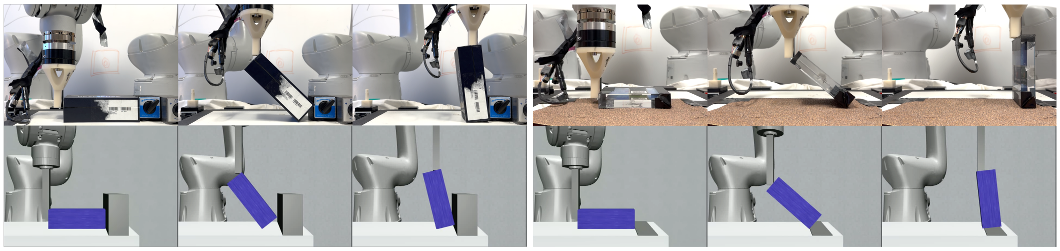
Overall, although there is some sim-to-real gap for both tasks, the robot could successfully perform the tasks on the physical hardware as shown in the attached supplemental video. We observe a larger sim-to-real gap for the pivoting with external wall task than the pivoting without wall task. This is because for the pivoting with wall task, the object induces the sliding contact between the object and the wall, and between the object and the table, which are relatively challenging to model precisely in simulator (e.g., MuJoCo), leading to a larger sim-to-real gap. In contrast, the pivoting-without-wall task does not involve sliding contacts, resulting in better sim-to-real transfer.



(a) Pivoting with external wall

(b) Pivoting without external wall

Figure 4: Comparison of the object angle in simulation and the real-world during pivoting. We execute the same policy both in simulation and in hardware and collect the object orientation during manipulation over 3 trials. Due to sensor discrepancies and physical modeling differences (i.e., sim-to-real gap), the resulting actions and motion can differ between simulation and hardware.



(a) Pivoting with external wall

(b) Pivoting without external wall

Figure 5: Snapshots of successful pivoting manipulation in simulation and real-world.

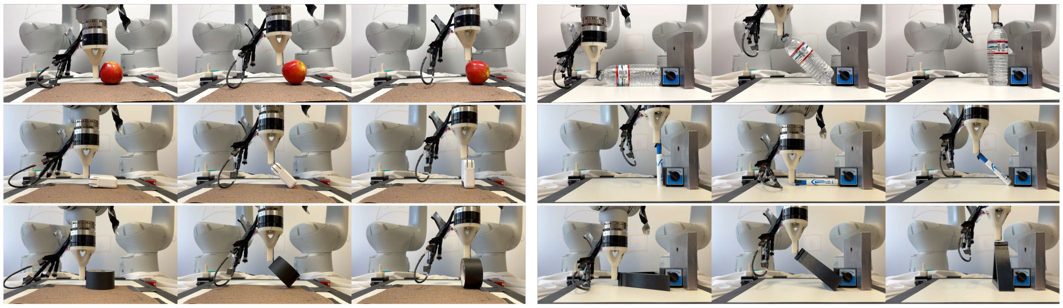


Figure 6: Snapshots of successful pivoting manipulation in real-world over various different objects.

6 Conclusion

In this paper, we present a framework for learning closed-loop controllers and estimators for contact-rich pivoting manipulation. We first leverage CITO to generate high-quality demonstrations, including object and robot states, contact forces, and extrinsic contact location. Then, we perform demonstration-guided RL using these demonstrations for training a teacher policy, enabling sample-efficient learning. Furthermore, we train a student estimator using only proprioception, vision, and force sensing, in order to predict the privileged information the teacher policy uses. Our framework is evaluated over several tasks, including the comparison against several baselines, and achieves successful zero-shot sim-to-real transfer in real-world experiments.

7 Limitations

Our work has the following limitations. First, we evaluate our framework exclusively on the pivoting task and do not demonstrate results for other non-prehensile manipulation tasks such as pushing and sliding. This choice was intentional to isolate and analyze key system components. However, our method does not assume task-specific priors and is applicable to a broader range of non-prehensile tasks, as long as CITO can generate dynamically feasible demonstrations, which is possible via the approach in [58] or other CITO methods such as [47].

Second, all evaluations in this work are performed on convex objects (e.g., boxes), and we do not report results for non-convex geometries. While none of our framework's modules rely on convexity assumptions, handling non-convex objects introduces additional complexity in contact reasoning.

Finally, during real-world deployment, we occasionally observed slight object slip (i.e., incipient slip [18, 56, 29]) relative to the robot, resulting in task failure. This issue is quite challenging: the slip must be large enough to produce detectable changes in sensor signals, allowing the student estimator to recognize it, yet small enough to avoid complete contact loss. This limitation is not a significant issue for other works focused on table-top manipulation [47], since objects are inherently stable. Addressing this limitation would likely require higher-resolution sensing or slip-specific estimation modules—for example, integrating visuotactile sensing (e.g., GelSight [70]) or augmenting the student model with incipient slip prediction capabilities.

References

[1] Depth Camera D435 — intelrealsense.com. `https://www.intelrealsense.com/depth-camera-d435/`. [Accessed 23-04-2025].

[2] Factory Automation - Mitsubishi Electric Americas — us.mitsubishielectric.com. `https://us.mitsubishielectric.com/fa/en/products/rbt/collaborative-robot/`. [Accessed 19-04-2025].

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

[4] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[5] Lars Ankile, Anthony Simeonov, Idan Shenfeld, Marcel Torne, and Pulkit Agrawal. From imitation to refinement–residual rl for precise assembly. *arXiv preprint arXiv:2407.16677*, 2024.

[6] Alp Aydinoglu, Philip Sieg, Victor M Preciado, and Michael Posa. Stabilization of complementarity systems via contact-aware controllers. *IEEE Transactions on Robotics*, 38(3):1735–1754, 2021.

[7] Alp Aydinoglu, Adam Wei, Wei-Cheng Huang, and Michael Posa. Consensus complementarity control for multi-contact mpc. *IEEE Transactions on Robotics*, 2024.

[8] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[9] Maria Bauza, Jose Enrique Chen, Valentin Dalibard, Nimrod Gileadi, Roland Hafner, Murilo F Martins, Joss Moore, Rugile Pevceviciute, Antoine Laurens, Dushyant Rao, et al. Demostart: Demonstration-led auto-curriculum applied to sim-to-real with multi-fingered robots. *arXiv preprint arXiv:2409.06613*, 2024.

[10] Cristian C Beltran-Hernandez, Damien Petit, Ixchel G Ramirez-Alpizar, and Kensuke Harada. Variable compliance control for robotic peg-in-hole assembly: A deep-reinforcement-learning approach. *Applied Sciences*, 10(19):6923, 2020.

[11] Aude Billard and Danica Kragic. Trends and challenges in robot manipulation. *Science*, 364 (6446):eaat8414, 2019.

[12] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[13] Jan Bruedigam, Ali Adeeb Abbas, Maks Sorokin, Kuan Fang, Brandon Hung, Maya Guru, Stefan Georg Sosnowski, Jiuguang Wang, Sandra Hirche, and Simon Le Cleac'h. Jacta: A versatile planner for learning dexterous and whole-body manipulation. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 994–1020. PMLR, 06–09 Nov 2025. URL `https://proceedings.mlr.press/v270/bruedigam25a.html`.

[14] Claire Chen, Zhongchun Yu, Hojung Choi, Mark Cutkosky, and Jeannette Bohg. Dexforce: Extracting force-informed actions from kinesthetic demonstrations for dexterous manipulation. *arXiv preprint arXiv:2501.10356*, 2025.

[15] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on robot learning*, pages 66–75. PMLR, 2020.

[16] Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8 (84):eadc9244, 2023.

[17] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[18] Siyuan Dong, Daolin Ma, Elliott Donlon, and Alberto Rodriguez. Maintaining grasps within slipping bounds by monitoring incipient slip. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3818–3824. IEEE, 2019.

[19] Juan Del Aguila Ferrandis, Joao Moura, and Sethu 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`.

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

[21] Yuni Fuchioka, Zhaoming Xie, and Michiel Van de Panne. Opt-mimic: Imitation of optimized trajectories for dynamic quadruped behaviors. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5092–5098. IEEE, 2023.

[22] Yuni Fuchioka, Cristian C Beltran-Hernandez, Hai Nguyen, and Masashi Hamaya. Robotic object insertion with a soft wrist through sim-to-real privileged training. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9159–9166. IEEE, 2024.

[23] Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.

[24] Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3389–3396. IEEE, 2017.

[25] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024. URL `https://www.gurobi.com`.

[26] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[27] Ankur Handa, Arthur Allshire, Viktor Makoviychuk, Aleksei Petrenko, Ritvik Singh, Jingzhou Liu, Denys Makoviichuk, Karl Van Wyk, Alexander Zhurkevich, Balakumar Sundaralingam, et al. Dextreme: Transfer of agile in-hand manipulation from simulation to reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5977–5984. IEEE, 2023.

[28] Francois R Hogan and Alberto Rodriguez. Reactive planar non-prehensile manipulation with hybrid model predictive control. *The International Journal of Robotics Research*, 39(7):755–773, 2020.

[29] Francois R Hogan, Jose Ballester, Siyuan Dong, and Alberto Rodriguez. Tactile dexterity: Manipulation primitives with tactile feedback. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 8863–8869. IEEE, 2020.

[30] Yifan Hou, Zeyi Liu, Cheng Chi, Eric Cousineau, Naveen Kuppuswamy, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Adaptive compliance policy: Learning approximate compliance for diffusion guided control. *arXiv preprint arXiv:2410.09309*, 2024.

[31] Yunfan Jiang, Chen Wang, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Transic: Sim-to-real policy transfer by learning from online correction. In *Conference on Robot Learning*, 2024.

[32] Shiyu Jin, Diego Romeres, Arvind Ragunathan, Devesh K Jha, and Masayoshi Tomizuka. Trajectory optimization for manipulation of deformable objects: Assembly of belt drive units. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10002–10008. IEEE, 2021.

[33] Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Champion-level drone racing using deep reinforcement learning. *Nature*, 620(7976):982–987, 2023.

[34] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987. doi: 10.1109/JRA.1987.1087068.

[35] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. 2021.

[36] Simon Le Cleac'h, Taylor A Howell, Shuo Yang, Chi-Yen Lee, John Zhang, Arun Bishop, Mac Schwager, and Zachary Manchester. Fast contact-implicit model predictive control. *IEEE Transactions on Robotics*, 40:1617–1629, 2024.

[37] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[38] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

[39] Toru Lin, Kartik Sachdev, Linxi Fan, Jitendra Malik, and Yuke Zhu. Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids. *arXiv preprint arXiv:2502.20396*, 2025.

[40] Matthew T Mason. Toward robotic manipulation. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:1–28, 2018.

[41] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science robotics*, 7(62):eabk2822, 2022.

[42] João Moura, Theodoros Stouraitis, and Sethu Vijayakumar. Non-prehensile planar manipulation via trajectory optimization with complementarity constraints. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 970–976. IEEE, 2022.

[43] Michael Noseworthy, Bingjie Tang, Bowen Wen, Ankur Handa, Chad Kessens, Nicholas Roy, Dieter Fox, Fabio Ramos, Yashraj Narang, and Iretiayo Akinola. Forge: Force-guided exploration for robust contact-rich manipulation under uncertainty. *IEEE Robotics and Automation Letters*, 2025.

[44] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407, 2011. doi: 10.1109/ICRA.2011.5979561.

[45] Kei Ota. Tf2rl. `https://github.com/keiohta/tf2rl/`, 2020.

[46] Kei Ota, Devesh Jha, Tadashi Onishi, Asako Kanezaki, Yusuke Yoshiyasu, Yoko Sasaki, Toshisada Mariyama, and Daniel Nikovski. Deep reactive planning in dynamic environments. In *Conference on Robot Learning*, pages 1943–1957. PMLR, 2021.

[47] Tao Pang, HJ Terry Suh, Lujie Yang, and Russ Tedrake. Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models. *IEEE Transactions on robotics*, 2023.

[48] Haozhi Qi, Ashish Kumar, Roberto Calandra, Yi Ma, and Jitendra Malik. In-hand object rotation via rapid motor adaptation. In *Conference on Robot Learning*, pages 1722–1732. PMLR, 2023.

[49] Haozhi Qi, Brent Yi, Sudharshan Suresh, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. General in-hand object rotation with vision and touch. In *Conference on Robot Learning*, pages 2549–2564. PMLR, 2023.

[50] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. doi: 10.15607/RSS.2018.XIV.049.

[51] Alberto Rodriguez. The unstable queen: Uncertainty, mechanics, and tactile feedback. *Science Robotics*, 6(54):eabi4667, 2021.

[52] Mingyo Seo, Steve Han, Kyutae Sim, Seung Hyeon Bang, Carlos Gonzalez, Luis Sentis, and Yuke Zhu. Deep imitation learning for humanoid loco-manipulation through human teleoperation. In *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE, 2023.

[53] Yuki Shirai, Xuan Lin, Yusuke Tanaka, Ankur Mehta, and Dennis Hong. Risk-aware motion planning for a limbed robot with stochastic gripping forces using nonlinear programming. *IEEE Robotics and Automation Letters*, 5(4):4994–5001, 2020. doi: 10.1109/LRA.2020.3001503.

[54] Yuki Shirai, Devesh K Jha, Arvind U Raghunathan, and Diego Romeres. Robust pivoting: Exploiting frictional stability using bilevel optimization. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 992–998. IEEE, 2022.

[55] Yuki Shirai, Xuan Lin, Alexander Schperberg, Yusuke Tanaka, Hayato Kato, Varit Vichathorn, and Dennis Hong. Simultaneous contact-rich grasping and locomotion via distributed optimization enabling free-climbing for multi-limbed robots. In *Proc. 2022 IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, pages 13563–13570, 2022. doi: 10.1109/IROS47612.2022.9981579.

[56] Yuki Shirai, Devesh K Jha, Arvind U Raghunathan, and Dennis Hong. Tactile tool manipulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12597–12603. IEEE, 2023.

[57] Yuki Shirai, Devesh K Jha, and Arvind U Raghunathan. Robust pivoting manipulation using contact implicit bilevel optimization. *IEEE Transactions on Robotics*, 40:3425–3444, 2024.

[58] Yuki Shirai, Arvind Raghunathan, and Devesh K Jha. Hierarchical contact-rich trajectory optimization for multi-modal manipulation using tight convex relaxations. *2025 IEEE International Conference on Robotics and Automation*, 2025.

[59] Yuki Shirai, Tong Zhao, HJ Suh, Huaijiang Zhu, Xinpei Ni, Jiuguang Wang, Max Simchowitz, and Tao Pang. Is linear feedback on smoothed dynamics sufficient for stabilizing contact-rich plans? *2025 International Conference on Robotics and Automation (ICRA)*, 2025.

[60] Jean-Pierre Sleiman, Jan Carius, Ruben Grandia, Martin Wermelinger, and Marco Hutter. Contact-implicit trajectory optimization for dynamic object manipulation. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 6814–6821. IEEE, 2019.

[61] Jean-Pierre Sleiman, Mayank Mittal, and Marco Hutter. Guided reinforcement learning for robust multi-contact loco-manipulation. In *8th Annual Conference on Robot Learning (CoRL 2024)*, 2024.

[62] Entong Su, Chengzhe Jia, Yuzhe Qin, Wenxuan Zhou, Annabella Macaluso, Binghao Huang, and Xiaolong Wang. Sim2real manipulation on unknown objects with tactile-based reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9234–9241. IEEE, 2024.

[63] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

[64] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017. doi: 10.1109/IROS.2017.8202133.

[65] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

[66] Mel Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017.

[67] Lasitha Wijayarathne, Ziyi Zhou, Ye Zhao, and Frank L Hammond. Real-time deformable-contact-aware model predictive control for force-modulated manipulation. *IEEE Transactions on Robotics*, 39(5):3549–3566, 2023.

[68] Haoyu Xiong, Quanzhou Li, Yun-Chun Chen, Homanga Bharadhwaj, Samarth Sinha, and Animesh Garg. Learning by watching: Physical imitation of manipulation skills from human videos. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7827–7834. IEEE, 2021.

[69] Zhengtong Xu, Raghava Uppuluri, Xinwei Zhang, Cael Fitch, Philip Glen Crandall, Wan Shou, Dongyi Wang, and Yu She. Unit: Data efficient tactile representation with generalization to unseen objects. *IEEE Robotics and Automation Letters*, 2025.

[70] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.

[71] Kevin Zakka, Baruch Tabanpour, Qiayuan Liao, Mustafa Haiderbhai, Samuel Holt, Jing Yuan Luo, Arthur Allshire, Erik Frey, Koushil Sreenath, Lueder A Kahrs, et al. Mujoco playground. *arXiv preprint arXiv:2502.08844*, 2025.

[72] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 5628–5635. Ieee, 2018.

[73] Xiang Zhang, Siddarth Jain, Baichuan Huang, Masayoshi Tomizuka, and Diego Romeres. Learning generalizable pivoting skills. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5865–5871. IEEE, 2023.

[74] Xiang Zhang, Changhao Wang, Lingfeng Sun, Zheng Wu, Xinghao Zhu, and Masayoshi Tomizuka. Efficient sim-to-real transfer of contact-rich manipulation skills with online admittance residual learning. In *Conference on Robot Learning*, pages 1621–1639. PMLR, 2023.

[75] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023.

[76] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, Yifeng Zhu, and Kevin Lin. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

A CITO Details

In this work, we use the CITO (1), as presented in [58]. Given a task description, defined by the initial and goal poses in SE(2), along with privileged information (e.g., object mass, friction, and size, environment friction), the optimization problem in (1) is solved through a sequence of three optimization problems. The first optimization problem is as follows.

$$\begin{aligned}
&\min_{\bar{\mathbf{q}}_t^o, \dot{\bar{\mathbf{q}}}_t^o,} \sum_{t=0}^{T} \|\bar{\mathbf{q}}_t^o - \bar{\mathbf{q}}_t^{o,\mathrm{ref}}\|_Q^2 \\
&\text{s. t. }, h_1\left(\bar{\mathbf{q}}_t^o, \bar{\mathbf{q}}_{t+1}^o, \dot{\bar{\mathbf{q}}}_t^o\right) = \mathbf{0},
\end{aligned} \tag{6}$$

where h_1 is the set of constraints, including velocity constraints, bounds on variables, and signed distance function-based constraints to ensure collision avoidance between the object and the environment. The optimization problem in (6) is used to obtain a kinematically feasible object pose trajectory and the corresponding extrinsic contact trajectory between the object and the environment. The optimization problem in (6) is solved using SNOPT [23].

Second, after fixing the object pose trajectory $\bar{\mathbf{q}}_t^o$ to the solution obtained in the first stage, the following optimization problem is formulated to account for non-smooth constraints due to contact dynamics:

$$\begin{aligned} &\text{Find } \ \bar{\mathbf{q}}_t^r, \dot{\bar{\mathbf{q}}}_t^r, \bar{\mathbf{y}}_t \\ &\text{s. t. }, h_2\left(\bar{\mathbf{q}}_t^r, \bar{\mathbf{q}}_{t+1}^r, \dot{\bar{\mathbf{q}}}_t^r, \bar{\mathbf{y}}_t\right) = \mathbf{0}, \end{aligned} \tag{7}$$

where h_2 represents the set of constraints used for considering non-smooth constraints, including contact making/breaking constraints, linearized force and moment balance constraints, and friction cone constraints. By solving (7), we obtain the object and robot trajectories that are not only kinematically feasible but also respect non-smooth contact constraints under linearized quasistatic dynamics. This optimization problem is a mixed-integer linear problem, which is efficiently solved using Gurobi [25].

Finally, given the solution obtained from (7), we consider the following optimization problem.

$$\begin{aligned} &\text{Find } \ \bar{\mathbf{q}}_t, \dot{\bar{\mathbf{q}}}_t, \bar{\mathbf{y}}_t \\ &\text{s. t. }, h_3\left(\bar{\mathbf{q}}_t^r, \bar{\mathbf{q}}_{t+1}^r, \dot{\bar{\mathbf{q}}}_t^r, \bar{\mathbf{y}}_t\right) = \mathbf{0}, \end{aligned} \tag{8}$$

where h_3 includes non-smooth sticking-sliding contact constraints using complementarity constraints as well as the original (not linearized) force and moment balance constraints. During solving (8) the robot's positions are locally adjusted to satisfy the nonlinear force and moment balance constraints and sticking-sliding complementarity constraints. This optimization problem is solved through SNOPT. Note that, for certain combinations of dynamics parameters (e.g., mass, friction), the solver may return an infeasible solution. In such cases, we do not include these infeasible solutions in the demonstration dataset.

It is worth noting that the solution obtained by sequentially solving the three optimization problems described above satisfies the full dynamics function f_{dyn} in (1) and is referred to as *dynamically feasible*. In contrast, if we solve the same sequence of optimization problems while removing all constraints involving contact forces—such as force and moment balance constraints and friction cone constraints—the resulting solution is referred to as *kinematically feasible* and satisfies the relaxed dynamics function f_{kin}.

In summary, solving (1) involves a sequence of the three optimization problems described above, allowing for efficient computation by decoupling different sets of constraints across the subproblems. See [58] for more details. Finally, we summarize the parameters used in the above optimization problems in Table 3.

Table 3: Hyperparameter setup for student estimator.

Parameter	Value
Optimizer	SNOPT for (6) and (8) and Gurobi for (7)
T	60 for pivoting with-wall task and 150 for without-wall task
time interval for integration	0.1 s

B Training Details in Simulation

In this section, we provide implementation details for training the teacher policy. The simulation environment is built using MuJoCo [65] with robosuite framework [76]. We use Soft Actor Critic (SAC) [26] to train the teacher policy. The training parameters are summarized in Table 4.

The coordinate is illustrated in Fig. 7. In this work, we operate within the SE(2) group, restricting manipulation to the $y - z$ plane.

Table 4: Hyperparameter setup for the teacher policy. Note that $\alpha_{i \in [1, \cdots, 8]}$ are the coefficients of the reward terms used for reward computation in (2).

Parameter	Value
total # of steps	300k for pivoting with-wall task and 1500k for without-wall task
batch size	4096
max # of step for timeout	300
Networks	[128, 128] MLP
learning rate for policy	1e-4
learning rate for Q function	3e-4
discount factor	0.9
replay buffer size	1e6
# of episodes for evaluation	50
# of episodes for warmstart	50k
$[\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8]$	[1, 0.075, 10, -1, -50, -50, -0.005, 5]

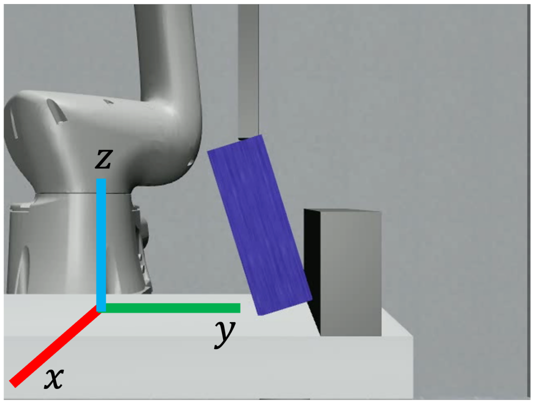


Figure 7: Definition of world frame used in this work.

B.1 Domain Randomization

During the training of the teacher policy, we perform domain randomization and add sensor noises to robustify the policy, which is summarized in Table 5.

Table 5: Dynamics randomization and sensor noise. $\mathcal{N}(\mu, \sigma)$ denotes a Gaussian distribution with mean μ and standard deviation σ, and $\mathcal{U}(a, b)$ denotes a uniform distribution over the interval of $[a, b]$. A $+$ symbol indicates that the sampled noise is added to the original parameter value.

Parameter	Range
object mass	$\mathcal{U}(0.04, 0.4)$ kg
friction for table and wall	$\mathcal{U}(0.01, 0.4)$
friction for objects	$\mathcal{U}(0.2, 0.7)$
friction for robots	$\mathcal{U}(0.7, 1.7)$
object size scale	$\mathcal{U}(0.95, 1.05)$
proportional gain k_p in OSC	$\mathcal{U}(2000, 8000)$
derivative gain k_d in OSC	see below
initial object position along y-axis	+$\mathcal{U}(-0.015, 0.015)$ m
initial robot position	+$\mathcal{U}(-0.015, 0.015)$ m
object position observation noise	+$\mathcal{N}(0, 0.015)$
robot position observation noise	+$\mathcal{N}(0, 0.00075)$
contact force observation noise	+$\mathcal{N}(0, 0.2)$

For the derivative gain k_d in operational space control (OSC) [34], we compute it based on the sampled proportional gain k_p to achieve critical damping using the relation $k_d = 2\sqrt{k_p}$.

It is worth noting that we represent object orientation using quaternions and apply domain randomization to account for sensor noise in orientation estimates. Specifically, we perturb the ground-truth quaternion $\mathbf{q} \in \mathbb{R}^4$ by composing it with a small random rotation:

$$\tilde{\mathbf{q}} = \delta\mathbf{q} \otimes \mathbf{q}$$

where $\tilde{\mathbf{q}}$ is the noisy quaternion, $\delta\mathbf{q}$ is a perturbation quaternion, and $\otimes$ denotes quaternion multiplication. The perturbation quaternion $\delta\mathbf{q}$ is constructed using a random axis-angle rotation. We first sample a unit axis $\mathbf{u} \in \mathbb{R}^3$ from a Gaussian distribution and normalize it:

$$\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{axis}}^2 \mathbf{I}), \quad \mathbf{u} \leftarrow \frac{\mathbf{u}}{\|\mathbf{u}\|}$$

Next, we sample a rotation angle θ (in degrees) from a clipped Gaussian distribution:

$$\theta \sim \text{clip}\left(\mathcal{N}(\mu_\theta, \sigma_\theta^2), -\theta_{\max}, \theta_{\max}\right)$$

We then convert the axis-angle representation to a unit quaternion via the exponential map:

$$\delta\mathbf{q} = \exp(\theta \cdot \mathbf{u})$$

In our implementation, we use the following parameters:

$$\sigma_{\text{axis}} = 0.1, \quad \mu_\theta = 0^\circ, \quad \sigma_\theta = 2^\circ, \quad \theta_{\max} = 5^\circ$$

This procedure injects bounded rotational noise into the observed quaternion while preserving unit norm and avoiding discontinuities.

B.2 Termination Conditions

An episode is terminated when any of the following conditions are met:

1. *Successful task completion*: A trial is considered successful if the final orientation error satisfies $|\theta_e| \leq 0.087$ radians (i.e., 5°).
2. *Significant deviation from the SE(2) plane*: If the object's x-position p_x deviates by more than $0.05\,\mathrm{m}$ from its initial value, i.e., $|p_x - p_x(t = 0)| \geq 0.05\,\mathrm{m}$, or if the z-position drops below the table surface, $p_z \leq p_z^{\text{table}}$, the episode is terminated and a penalty of -100 is applied.
3. *Timeout*: The episode exceeds the maximum number of steps as defined in Table 4.

C Student Estimator Details

In this section, we provide details about the training procedure for the student estimator.

C.1 Data Collection

To construct the dataset for student estimator training, we rollout the trained teacher policy in simulation and record the ground-truth privileged information, sensor observations, and corresponding object segmentation masks under domain randomization. We use the same range of domain randomization used during teacher policy training Table 5. Since segmentation masks are not used during teacher policy training, we introduce additional uncertainties to simulate realistic conditions, including:

- **Erosion/Dilation:** Morphological operations applied with random kernel sizes to simulate over- and under-segmentations.
- **Partial Mask Dropout:** Circular regions within the mask are randomly removed to mimic occlusions or partial detection failures.
- **Full Mask Dropout:** With a small probability, the entire mask is dropped (set to all zeros) to simulate complete sensor failure or occlusion.

- **Flip Noise:** Individual pixels are randomly flipped to simulate salt-and-pepper noise or detector flickering.
- **Edge Perturbation:** Object boundaries are randomly jittered to simulate segmentation boundary inaccuracies.
- **Spatial Augmentation (Affine):** Random affine transformations are applied to the mask, simulating viewpoint shifts and calibration noise.
- **Gaussian Blur:** A blur filter is applied to soften sharp edges and simulate optical imperfections.

The configuration of the segmentation domain randomization is summarized in Table 6.

Table 6: Segmentation mask domain randomization parameters used during student data collection.

Noise Type	**Parameter**	**Value**
Erosion/Dilation	Probability for erosion/dilation	0.7
	Kernel size choices	$\{3, 5, 7\}$
	Erosion vs. dilation split	0.5
Random Holes	Number of holes	3
	Hole radius range	$[3, 9]$ pixels
	Hole probability	0.5
Full Mask Dropout	Probability	0.05
Flip Noise	Pixel flip probability	0.01
Edge Perturbation	Edge noise probability	0.75
	Edge point noise probability	0.1
Spatial Augmentation (Affine)	Rotation range	$\pm 2.5°$
	Translation range	$\pm 7.5\%$
	Scaling range	$[0.95,\ 1.05]$

C.2 Student estimator training

Given the dataset collected in Section C.1, we train a student estimator composed of a CNN followed by a TCN. The CNN takes as input a binary segmentation mask of size $1 \times 480 \times 640$ and consists of three convolutional layers with kernel sizes of $(3, 3, 3)$, and strides of $(2, 2, 1)$, and output channels of $(16, 32, 64)$, respectively. An adaptive average pooling layer reduces the spatial dimensions to 8×8, followed by a fully connected layer that produces a 1×128 feature vector. The TCN processes the temporal sequence of CNN features concatenated with proprioceptive and force features. It consists of three layers of 1D dilated causal convolutions, each with 128 channels and a kernel size of 2, and dilation rates of 1, 2, and 4. We consider two types of privileged information: time-invariant dynamics parameters (i.e., mass and size of the object), and time-varying values such as the object pose. To accommodate this distinction, the student estimator employs two separate fully connected layers—one for predicting the time-invariant variables and another for the time-varying privileged quantities (e.g., object pose). The output dimensions of each head match the corresponding target variables. We find that this separation leads to improved estimation performance.

Then, the model is trained by minimizing the mean square error between the ground-truth and predicted values by the student estimator. Fig. 8 shows the learning curve of the validation loss during training. The hyperparameters used for training the student estimator are summarized in Table 7.

Table 7: Hyperparameter setup for student estimator.

Parameter	Value
total # of epochs	20
batch size	256
initial learning rate	1e-3
learning rate schedule	ReduceLROnPlateau from PyTorch
optimizer	Adam

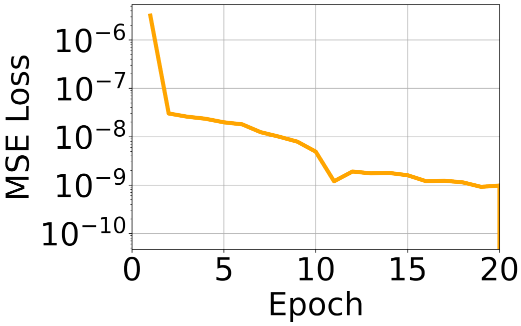


Figure 8: Student estimator validation loss over epochs.

D Ablation Study

D.1 Effect of linear and quadratic reward terms during teacher policy training

In Section 3, we mention that using linear and quadratic terms in r_p in (2) is important to ensure that the robot completes the pivoting task. To validate this claim, we conducted an ablation study using dynamics-conditioned RL, evaluating three reward variants: (1) linear only, (2) quadratic only, and (3) both linear and quadratic terms in r_p, under settings with and without domain randomization. Table 8 shows the mean and standard deviation of the terminal object angle over 50 evaluation episodes.

When domain randomization is disabled, the policy trained with the linear term alone in r_p successfully completes the pivoting task. In contrast, using only the quadratic term leads to task failure, likely due to the difficulty in reward shaping—quadratic rewards are sparse and less informative during early training. On the other hand, when domain randomization is enabled, policies trained with only the linear term exhibit significantly degraded performance. In this case, combining linear and quadratic terms improves performance substantially. We hypothesize that the quadratic component offers a stronger gradient signal when the agent is close to the goal, helping to overcome the increased noise due to domain randomization.

Table 8: Comparison of terminal object angle using different reward formulation with/without domain randomization. In the terminal angle, we show its mean with standard deviation over 50 episodes.

Reward type	Enable domain randomization	Terminal angle [deg]
Linear term	No	88.1 ±0.21
Quadratic term	No	0.0 ±0.10
Linear + Quadratic term	No	88.9 ±0.20
Linear term	Yes	70.1 ±0.59
Quadratic term	Yes	0.0 ±0.71
Linear + Quadratic term	Yes	88.2 ±0.44

D.2 Pivoting with wall task without domain randomization

In Section 5, we present the result of the training curve using different RL training runs for two tasks. For the results in Fig. 2, we consider domain randomization, and thus it is possible that the pivoting with external wall task could not be trained due to the large domain randomization. Hence, we show the result for the pivoting with wall task under no domain randomization as shown in Fig. 9.

Fig. 9 shows that all RL using different reward equations could successfully learn the skill. Among them, dynamics-conditioned RL exhibits the fastest learning rate. This confirms that while vanilla RL can succeed when the training environment is noise-free, providing dynamics-consistent demonstrations significantly improves the learning efficiency by offering more informative reward signals.

We emphasize that for the pivoting without wall task, even under no domain randomization, vanilla RL and kinematically-conditioned RL fail to learn. This supports our claim that non-prehensile

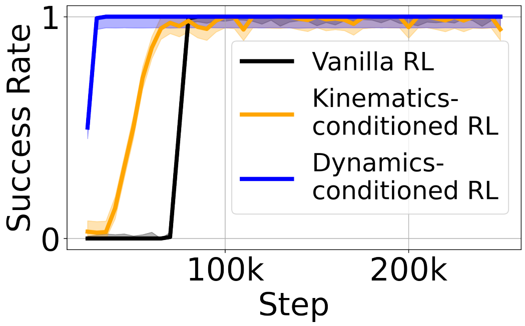


Figure 9: Learning curves for different RL training runs for pivoting-with-wall task. Solid lines indicate average success rates, and shaded regions denote standard deviation across three different random seeds. Every 10k step, the current policy is evaluated over 50 episodes, and the success rate is plotted.

manipulation tasks have very narrow feasible action regions. Therefore, leveraging demonstrations that satisfy complex contact constraints plays an important role in improving learning efficiency.

D.3 Student Estimator Performance

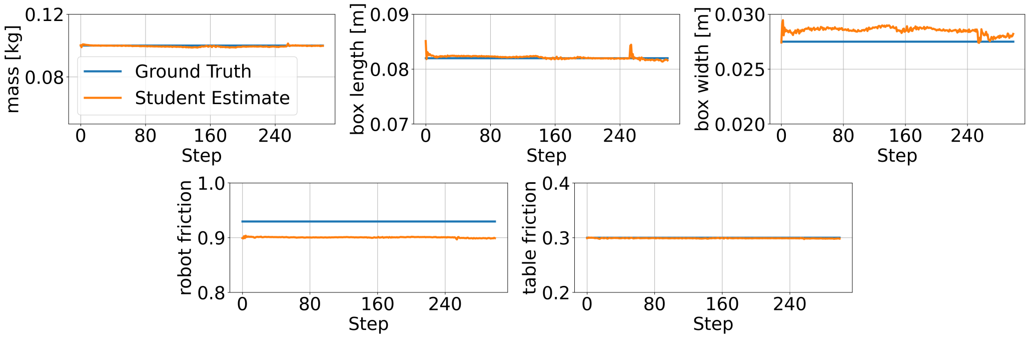


Figure 10: Comparison of our student estimator's predictions and the ground truth for the box mass, the box length, the box width, the robot friction constant, and the table friction constant, for the pivoting with a wall.

In Section 5, we present a subset of our student estimator results due to page limitations. We show the remaining privileged information figures in Fig. 10. Overall, we observe that our student estimator successfully predicts the privileged information with reasonable accuracy.