

UC Santa Cruz

UC Santa Cruz Electronic Theses and Dissertations

Title

Contribution of Force Sensing at Fingertips on the Autonomous Learning of In-Hand Manipulation Without Vision

Permalink

<https://escholarship.org/uc/item/9q0741d8>

Author

Ojaghi, Pegah

Publication Date

2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

**CONTRIBUTION OF FORCE SENSING AT FINGERTIPS ON THE
AUTONOMOUS LEARNING OF IN-HAND MANIPULATION
WITHOUT VISION**

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Pegah Ojaghi

December 2022

The Dissertation of Pegah Ojaghi
is approved:

Dr. Michael Wehner, Chair

Dr. Luca de Alfaro

Dr. Francisco Valero-Cuevas

Peter Biehl
Vice Provost and Dean of Graduate Studies

Copyright © by

Pegah Ojaghi

2022

Table of Contents

List of Figures	v
List of Tables	x
Abstract	xi
Acknowledgments	xii
1 Introduction	1
1.1 Current state-of-the-art in in-hand manipulation and its limitations . . .	1
1.2 Contributions of the current work to the field	4
1.3 Consequences of this work	6
1.4 Limitations of this work	6
1.5 Thesis Outline	7
2 Establish an end-to-end simulation environment to enable in-hand ma- nipulation	8
2.1 Chapter summary	8
2.2 Simulation Environment	9
2.2.1 Tactile Information	12
3 Reinforcement Learning Algorithm for dexterous manipulation	15
3.1 Chapter summary	15
3.2 Model-free RL algorithm	16
3.2.1 Details on the reward function	16
3.3 Learning the policy	17
3.3.1 Proximal Policy Optimization Algorithm	17
3.4 Overview of Simulation Environment and Learning Trail	20
3.5 Curricula	21
3.5.1 Details on the curriculum	23

4	Identifying useful learning strategies	25
4.1	Chapter summary	25
4.2	The Effects of Learning Strategies	25
4.2.1	Curriculum 1	26
4.2.2	Curriculum 2	26
4.2.3	Curriculum 3	28
4.2.4	Curriculum 4	33
4.2.5	Curriculum 5	39
4.3	Curriculum drives manipulation performance	43
4.4	The second phase of learning dictates end performance	45
4.5	Rotating the ball is critical in learning how to lift the ball	46
5	Effect of sensory information	48
5.1	Chapter summary	48
5.2	In-hand manipulation performance	48
5.3	Effect of sensory information	49
5.4	Learning Trends	52
6	Conclusion	55
6.1	Learning	55
6.2	Sensory Modalities	56
6.3	Active sensing	57
6.4	Limitations, opportunities and future directions	58
	Bibliography	60

List of Figures

2.1	A three-finger robotic agent the MuJoCo environment.	10
2.2	Degrees of freedom (DOFs) of the hand: a three-finger robotic hand interacts with a 70 mm diameter ball in the MuJoCo environment (only one of the three fingers is shown for simplicity)	11
2.3	The ball can traverse on x and y directions with (without damping or friction) and rotate around the z axis (with damping and friction). The system can work with no tactile sensory, binary, normal force only, or both normal and tangential forces, which creates four cases for the tactile sensory conditions. These are null (No-tactile), contact or no contact (Binary-contact, 1 or 0), force normal to the finger pad (Normal-force, f_n), or the full contact force vector (3D-force, $f = [f_{t,1}, f_{t,2}, f_n]$).	14
3.1	Our reward is a combination of i) rewarding the angular velocity of the ball $\dot{\theta}_y$ in a positive direction (primary reward); and ii) penalizing the distance between the ball and desired heights $ z_d - z_b $ (penalty reward)).	17
3.2	Basic process of learning dexterous manipulation by RL.	21
3.3	Overview of simulation environment and learning.	22
3.4	The overview diagram of Proximal Policy Optimization (PPO) algorithm for in-hand manipulation. The ball's state is not passed into the PPO policy for learning.	24

4.1	Performance for Curriculum 1 [L R+L] for all options of tactile information. The joint distribution of mean ball’s height (mm) vs. the number of completed rotations in each Monte Carlo run for the last episode is shown. Each Monte Carlo run is color-coded based on the range of reward from lowest to highest reward among all tactile conditions. Curriculum 1 failed to lift the ball or rotate it in all MC runs. These results highlight the importance of the choice of Curriculum targeting a particular performance goal. For example, rotating the ball is most critical for the performance of a task.	27
4.2	Performance for Curriculum 2 [R R+L] for all options of tactile information. The learning policy enabled the robotic hand to rotate the ball in the first 1,000 episodes. After changing the reward to both rotation and lift, the hand still gets most of its reward via rotation since lifting is incorporated into the reward function only at the second half of the MC run. Although in general the robotic hand is not successful in lifting the ball in all MC runs, several instances of manipulation (both rotation and lift) are demonstrated.	29
4.3	Performance for Curriculum 2 [R R+L] for all options of tactile information. It shows the violin plot of completed rotations for each Monte Carlo run at every 250 episodes. The width of the plot indicates the frequency of the corresponding completed rotations and black points are shown each Monte Carlo runs.	30
4.4	Performance for Curriculum 2 [R R+L] for all options of tactile information. Boxplots of the aggregated reward for tactile information while the boxes’ orange lines show the median values, the edges of the boxes are the 25th and 75th percentiles. The reward gained per 2,000 episodes is visualized as boxplots grouped by the each tactile information aggregated over 60 Monte Carlo runs per box.	31

4.5	Performance for Curriculum 3 [R+L R+L] for all options of tactile information. The cumulative reward for the last episode of each MC run is color coded. Note that the desired manipulation performance is represented by those points inside the green box defining the desired ball height. Note the ceiling and floor effects for mean height as the ball can get caught against the palm, or be rolled against the ground, respectively.	32
4.6	Performance for Curriculum 3 [R+L R+L] for all options of tactile information. Lift success rate (the percentage of time the ball spent within 25% of the desired height over the duration of an episode). Boxplots, with median, across tactile information options for 60 MC runs at eight representative episodes, 250 episodes apart. Lift success rate (the percentage of time the ball spent within 25% of the desired height over the duration of an episode).	34
4.7	Performance for Curriculum 3 [R+L R+L] for all options of tactile information. Cumulative reward for each representative episode. Boxplots of the aggregated reward for tactile information while the boxes' orange lines show the median values, the edges of the boxes are the 25th and 75th percentiles. The reward gained per 2,000 episodes is visualized as boxplots grouped by the each tactile information aggregated over 60 Monte Carlo runs per box.	35
4.8	Performance for Curriculum 4 [R+L R] for all options of tactile information. The lack of reward for lifting in the second half of the 60 MC runs focuses the robotic hand on rotating the ball in its final performance. Surprisingly, the robotic hand continues to lift the ball even if not rewarded for it.	36
4.9	Performance for Curriculum 4 [R+L R] for all options of tactile information. It shows the violin plot of completed rotations for each Monte Carlo run at every 250 episodes. The width of the plot indicates the frequency of the corresponding completed rotations. Each black point represents one Monte Carlo run.	37

4.10	Performance for Curriculum 4 [R+L R] for all options of tactile information. Boxplots of the aggregated reward for tactile information while the boxes’ orange lines show the median values, the edges of the boxes are the 25th and 75th percentiles. The reward gained per 2,000 episodes is visualized as boxplots grouped by the each tactile information aggregated over 60 Monte Carlo runs per box.	38
4.11	Performance for Curriculum 5 [R+L L] for all options of tactile information. Note that the lack of reward for rotation in the second half of the 60 MC Runs now allows the hand to focus on placing the ball within the desired height range. Surprisingly, however, the robotic hand continues to rotate the ball even if not rewarded for it—likely because it has ‘learned’ rotation and lift in a coupled way. However, note that lift performance and learning rates are largely equivalent across tactile information options—with 3D-force being highest	40
4.12	Performance for Curriculum 5 [R+L L] for all options of tactile information. Lift success rate (the percentage of time the ball spent within 25% of the desired height over the duration of an episode). Boxplots, with median, across tactile information options for 60 MC runs at eight representative episodes, 250 episodes apart.	41
4.13	Performance for Curriculum 5[R+L L] for all options of tactile information. Cumulative reward for each representative episode. Boxplots of the aggregated reward for tactile information while the boxes’ orange lines show the median values, the edges of the boxes are the 25th and 75th percentiles. The reward gained per 2,000 episodes is visualized as boxplots grouped by the each tactile information aggregated over 60 Monte Carlo runs per box.	42
4.14	Pareto plots for the final performance from each curriculum highlight the divergence in manipulation performance across learning strategies. Final performance for all curricula across all tactile information options. . . .	44

5.1	<p>Pareto plots for the final performance across curricula and the four tactile information options. Each Pareto plot shows the mean final performance for all curricula and corresponding tactile information available to the learning policy: (a) No-tactile, (b) Binary-contact, (c) Normal-force, and (d) 3D-force. While curriculum drives learning to distinct regions, the tactile information available to the PPO policy (Fig. 3.3) also affects the robotic hand’s ability to learn manipulation. Interestingly, we see that learning happened even in the absence of tactile information (a), and that manipulation performance was not always best with 3D-force information (b-d). Note these Pareto plots only consider those final episodes when the ball was on average lifted to within 25% of the desired height.</p>	53
5.2	<p>Types of ‘learners’ for Curriculum 5 with 3D-Force sensing. Out of 60 Monte Carlo runs, four distinct types of learners were visually identified: those that after the change in reward from a combination to only lifting, a had their performance decrease before going on to exceed the performance at the end of 1K trials (10% of trials), b experienced a sudden increase in performance at the switch (53.33% of trials), c continuously improved their lifting performance (13.33% of trials), or d plateaued in their learning well within the first phase (18.33% of trials). Note that 5% of runs experienced no learning. The shaded region in the ‘Dip and Improve After Switch’ and ‘Improve After Switch’ highlight the change in performance when the reward changes after 1K episodes.</p>	54

List of Tables

2.1	Physical simulation parameters for the three-fingered robotic hand and the ball.	12
2.2	Tactile information options available to the learning algorithm.	13
3.1	Proximal-policy optimization (PPO) hyperparameters	20
3.2	We used five curricula that rewarded different combinations of rotation and lift during each half of a Monte Carlo (MC) run. These changes in the coefficients of the reward function define a progression of goals (i.e., curriculum learning) over the two halves of each run.	24

Abstract

Contribution of force sensing at fingertips on the autonomous learning of in-hand
manipulation without vision

by

Pegah Ojaghi

Autonomous dexterous manipulation (i.e., reorienting objects with the fingertips) remains beyond the *grasp* of robots. Dexterous manipulation (e.g., picking up a lemon to squeeze it) differs from *grasping* an object because it requires re-orienting the object with the fingertips without dropping it. Prior work has demonstrated autonomous learning to manipulate objects, but dropping them is prevented by an upward-facing palm, a table-top, or slowly introducing gravity.

In this dissertation, I demonstrated autonomous learning of dexterous manipulation of a ball with a downward-facing palm against gravity. I use a reinforcement learning algorithm (proximal-policy optimization) to demonstrate that a simulated downward-facing three-fingered robotic hand can autonomously learn to reach for and manipulate a 5 gram ball while rotating it at the desired height. Importantly, only curricula that rewarded ball rotation from the start succeeded. This dynamic interaction with the ball in the absence of vision, like child’s play, is likely a form of active sensing necessary to build useful end-to-end models for dexterous manipulation against gravity. Tactile information was useful in interesting ways for this ball: Lacking tactile information hindered—but did not prevent—learning, while Binary-contact and Normal-force sensing performed comparably to 3D-force. This dynamical interplay among curricula, tactile information and learning trends illuminate features of human manipulation and provide a path towards autonomous reach and manipulation by robots.

Acknowledgments

This thesis was only possible through the support and guidance of many people who I have been fortunate to have be a part of my life. First and foremost, I want to express my strongest gratitude to my mentors and advisors, Dr. Michael Wehner and Dr. Francisco J. Valero-Cuevas provided their support whenever I needed it. I would also like to thank my committee member Dr. Luca de Alfaro, for your time, guidance, and feedback.

I would also like to thank all Brain-Body Dynamics Laboratory (Valero lab) members, especially Dr. Ali Marjaninejad, Dr. Andrew Erwin, and Romina Mir, from whom I have learned a lot during these years and with whom I believe I have grown to be a better person. Also, I would like to thank all my non Brain-Body Dynamics Laboratory friends as well. Especially Dr. Mohammad Ovais Aziz-zanjani who has all been a source of inspiration and support to me while carrying out my research at University of California Santa Cruz. I truly appreciate all of the people in my life and feel extremely lucky to have such an amazing circle of support.

Finally, and most importantly, I would like to thank my mother and father for educating and raising me, my sister for their love, encouragement, and support over the years. Your sacrifices made it possible for me to be where I am right now. You are, and always will be, in my heart and mind. Without all of you none of this would have been possible. This dissertation is dedicated to all of you.

Chapter 1

Introduction

1.1 Current state-of-the-art in in-hand manipulation and its limitations

The human hand's ability to interact with the world is essential to our biomechanical, manipulative, perceptual, cognitive, psychological, social, linguistic, and artistic everyday activities [1, 2, 3]. This fantastic ability to manipulate objects of various shapes, sizes, and materials and control the objects' position has inspired scientists and inventors for centuries (e.g., [4]). Recently, dexterous manipulation of objects, one of the most complex types of biological movement and a fundamental everyday task for humans, has attracted the attention of many researchers in the robotic community. Building robots with dexterous hand manipulation ability provides tools for humans to perform repetitive and dangerous tasks while avoiding harm. The flexibility of dexterous hands in robots is critical when the goal is to blend the robots into a human-centric environment created with ergonomics in mind. The similarity of robotic hands to humans would facilitate engaging interaction, which is vital for education, caregiving [5] and helping the elderly and individuals living with disabilities.

Despite the necessity of dexterous manipulation for robotic systems achieving it for autonomous robots is still challenging and remains an open problem [6, 7, 1, 8, 9, 4, 10]. Today’s successful control algorithms for dexterous manipulation often require a combination of models of the physical system or an expert demonstration of the task [11, 12, 13, 14]. Therefore, current methods either leverage a prior model or derive one from large amounts of data [15, 16, 17, 18]. The conventional model-based control algorithms are one of these methods which generate trajectories for complex and dynamic in-hand manipulation [19, 20, 21]. However, these methods rely on accurate dynamics models and state estimates, which are often challenging to create for the intermittent contacts, friction, and nonlinearities characteristic of manipulation tasks in the unstructured real world. Even with a perfect model, most existing methods of in-hand manipulation are too slow to operate in real-time, making it impossible to adapt to an uncertain environment. Another approach for learning dexterous manipulation is to use a model learned from data, where the model either operates directly over the raw state or over a feature representation of the state [22]. Although these models are very data-efficient, they do not scale well to complicated nonlinear dynamics or high-dimensional state spaces.

In addition, vision is often necessary for these systems to perform well [11]. These methods, however, continue to be impractical for autonomous learning and performance of manipulation in unstructured human environments.

Reinforcement Learning (RL), which refers to learning to behave optimally in a stochastic environment by taking actions and receiving rewards, addresses these shortcomings and promises autonomous learning of in-hand manipulation with minimal human supervision. Reinforcement learning methods have made significant progress for dexterous manipulation to aid the development of intelligent agents with the ability to adapt to new circumstances rapidly and achieve goals in a wide range of environments [23]. RL approaches, either model-based policy or model-free policy, can circumvent the is-

sues with conventional model-based control algorithms for dexterous manipulation [24]. Model-based RL approach builds a predictive model of an environment and derives a controller from it. However, their primary drawback is inferior performance compared to their model-free counterparts [23] which is driven by the necessity to learn accurate models to find a good policy. Model-free RL techniques for in-hand manipulation learn a direct mapping from states to actions (learning complex policies from raw state inputs) [25, 26] with good performance on complex tasks [27, 28].

The aforementioned current studies on autonomous learning in dextrous robots are focused on either *grasp* or *manipulation* of an object resting on the upward-facing palm [29, 30, 31, 11]. It is essential to distinguish manipulation as the ability to hold an object with the fingertips to change its orientation dynamically. Moreover, manipulation is distinct from grasp, which is the static coupling of an object to the hand by applying finger forces [8, 4, 32].

Despite the previous studies' successful results, the upward-facing palm configuration makes the grasp inherently stable, and the possibility of dropping the objects held in these configurations is not likely. However, in-hand manipulation performed against gravity is a vitally important capability for robots to achieve real-world tasks. Another line of work studied the autonomous dexterous in-hand manipulation with a hand facing upward or downward [33]. Although this is the only work that manipulates an object with a hand facing downward without an underlying surface, it is essential to initialize the object in a stable configuration. Moreover, to improve the performance, the authors used training in a curriculum where gravity is slowly introduced (i.e., gravity curriculum).

Lastly, despite the excellent performance of the aforementioned autonomous learning methods with a hand facing upward or downward absence of tactile sensing imposes certain limitations on these approaches. In addition, vision is often necessary for these systems to perform well [11]. These methods, however, continue to be impractical for

autonomous learning and performance of manipulation in unstructured human environments. With visual uncertainty due to lighting, shadows, and occlusions, tactile information is necessary to perform an in-hand manipulation task. Rich tactile information available to humans and robots allows them to recognize and manipulate an object without vision. Tactile information can provide greater robustness to variations in object properties [34, 35, 36], perturbations [37], and sensing errors [38] beside the information that provides on the object’s pose and contact normals [39, 40]. Investigations on tactile sensing have the potential to improve robotic in-hand manipulation. Learning and execution of manipulation tasks depend on the availability of proper tactile information [9, 8]. Recent trends in using tactile information in the robotic community focus on the role of tactile information in grasping the object rather than dexterous manipulation [41, 42, 43, 44]. Even though there is strong evidence that tangential forces play a role in dexterous manipulation tasks in humans [34, 45, 46], the contribution of these tangential forces to autonomous dexterous manipulation in robots remains unexplored. Not knowing the optimal type and distribution of tactile sensors on the fingers or the object and the need to process large data sets are some of the reasons why tactile information has not made systematic headway in robotic manipulation [47].

To the best of our knowledge, one study has explored improved autonomous dexterous in-hand manipulation for rolling an object with tactile feedback [27]. Due to difficulty in modeling, the approach only trains on a physical robot which is slow and costly to run. However, this work manipulates an object resting on a table not held against gravity.

1.2 Contributions of the current work to the field

The limitations and needs mentioned in the previous section are why this thesis focuses on autonomously learning in-hand manipulation tasks against gravity with different levels of tactile information. Moreover, to successfully manipulate the object we did not

need to ‘initialize the object in a stable configuration’ which is needed in other works. As in biology, our end-to-end RL algorithm can learn quickly (i.e., be data-efficient), emphasizing tactile information for effective dexterous manipulation. The algorithm explores learning to manipulate objects against gravity while studying the effects of force sensing at the fingertips to accelerate learning and enhance performance. Moreover, it does so based on a data-driven approach that uses few shots (limited experience).

At its core, my work significantly extends that of prior approaches as the simulation environment includes the effects of gravitational acceleration (so there is a risk of dropping the ball), realistic contact dynamics, and our robotic agent learns to manipulate the object with few shots and limited information. In contrast to model-based approaches (which require full prior knowledge of the hand and object) and data-driven in-palm hand manipulation (which make the grasp inherently stable), my autonomous learning does not use *a priori* model. In addition, the algorithm shows how the influence of type of force sensing impact the agent’s ability to learn to manipulate the object autonomously. I show that providing the rotation from the start is necessary for learning manipulation.

I explore this idea of mastering tasks by encouraging the agent to start with task-specific exploration that would lead to active sensing in the context of autonomous robotic manipulation. Importantly, we evaluated different levels of tactile information that will guide and justify future work to implement this approach to hardware.

The way my work went beyond the state-of-the art, therefore, is by demonstrating for the first time a method with the ability to autonomously learn to manipulate an object against gravity while revealing the role of tactile information in in-hand manipulation. Moreover, it does so based on a data-driven approach that uses few shots (limited experience). Importantly, our work focuses on manipulating the objects against gravity with the risk of being dropped at any time. Lastly, my work underlines the importance of curricula in manipulation and shows how the right choice of a curriculum can enhance

performance and robustness across multiple tasks by facilitating active-sensing.

1.3 Consequences of this work

This work is a step toward achieving the autonomous learning and adaptation capabilities seen in biological systems with exceptional agility and energy efficiency in in-hand manipulation. By demonstrating the natural emergence of learning for grasp and manipulation in bio-robotic systems, we shed light on the implicit sensorimotor processes in biological systems that may grant humans unparalleled dexterity.

Broadly, this work advances neuroprosthetics in a novel bio-inspired way. It helps form a foundation for future studies on the role of tactile sensory information in human adaptability. Further, it provides valuable insight for developers and manufacturers of affordable consumer-products types of robots that do not rely on vision to operate in unstructured environments. These sensory-motor robots will allow significant progress in understanding hand disabilities and rehabilitation in multiple neurological conditions and strokes.

1.4 Limitations of this work

While our work pushes the field of autonomous manipulation forward, it has some limitations. We have only studied the robotic hand with three fingers due to the difficulty of controlling the hand with five fingers of such complexity. Using three fingers seems a good compromise for manipulation since it is the minimum number of fingers required to accomplish stable grasps. However, the scalability of this approach to systems with more fingers is an interesting question that should be addressed in future work. Moreover, here we mainly focused on learning in-hand manipulation in the agent actuated by servo motors and did not consider tendon-driven hand.

Evaluating the performance across a more comprehensive set of different manipulation tasks—is another interesting research topic that can be investigated in future work. In addition, here, we evaluated the adaptability of the system to different learning curricula but did not study how it performs in the deployment phase with uncertainty in the environment.

Lastly, we tested our approach in simulations only. But, as with many other studies looking to bridge the sim2real divide [48, 49], we used a realistic physics engine (i.e., MuJoCo) that enables future work to implement our approach in hardware. Developing the hardware system for this hand would be an essential next step (on which we are working) to test the full potential of our approach and study its real-world performance.

1.5 Thesis Outline

The following six chapters forming this dissertation are organized as follows: In Chapter 2, an overview of how to set up an end-to-end simulation environment to enable in-hand manipulation and a deeper insight into our algorithm pipeline are provided. Chapter 3 introduces our proposed learning algorithm, which enable autonomous learning in our servo-driven hand using limited experience. The autonomous learning is possible by the choice of useful curricula, complemented by meta-parameters in the Proximal Policy Algorithm. In Chapter 4, we study the effects of learning strategies in autonomously learning of in-hand manipulation. Chapter 5 evaluates the impact of different type of force sensing on the agent’s ability to autonomously learn to manipulate the object autonomously. Lastly, Chapter 6 summarizes all proposed algorithms, benefits, and conclusion.

Chapter 2

Establish an end-to-end simulation environment to enable in-hand manipulation

2.1 Chapter summary

To demonstrate autonomous in-hand manipulation with a downward-facing three-finger robotic hand without vision, we establish an end-to-end simulation environment. This breakthrough was made possible mostly by the choice of a useful curriculum combined with proper meta-parameters for the established reinforcement learning (proximal-policy optimization) algorithm [50]. To reveal the effect of dimensionality of force sensing at the fingertips on the autonomous learning of in-hand manipulation, we divided my work into designing a hand in simulation and learning dexterous in-hand manipulation policies using reinforcement learning. To check the effectiveness of different tactile sensory levels in in-hand manipulation, I responded to the fundamental question of ‘which task should our hand execute?’. We choose the dynamical manipulation task of lifting a ball

and rotating it along a horizontal rotation axis at a target height. This section first describes the design of our hand dynamics, tactile sensory information, the ball, and our environment. Next, we go deeper into our learning algorithm in the next chapter.

2.2 Simulation Environment

Designing a bio-inspired hand to perform autonomous learning for in-hand manipulation in simulation is challenging because of the unique functions and abilities of the human hand. Due to the high complexity of five fingers, controlling such a system is difficult [11]. Although prior methods of controlling the five-fingered hand have shown promising in-hand manipulation results in simulation, either they have not been transferred to a real-world robot [19, 20] or only trained on a physical robot [18, 27, 51]. Due to the fact that physical trials are so slow and costly to run, the learned behaviors are very limited. In this work, we designed a 3-fingered bio-inspired hand. Using three fingers seems a good compromise for manipulation since the minimum number of fingers is required to accomplish stable grasps in simulation and hardware.

Our autonomous learning manipulation is performed in a simulated environment in the MuJoCo physics engine, which allows us to implement reinforcement learning algorithms on a robotic agent in a realistic environment that includes contact dynamics (including penetration) and gravitational acceleration [52, 21]. The MuJoCo physics engine provides methods to mimic touch sensing at specified locations. This is based on specifying the tactile sensors’ active zones by so-called sites. Each site can be represented as either ellipsoid or a box. If a body’s contact point falls within a site’s volume and involves a geometry attached to the same body as the site, the corresponding contact force is included in the sensor reading.

We simulated a bio-inspired, three-fingered robotic hand with a palm and three identical servo-driven fingers: two adjacent fingers (analogous to the ‘index’ and ‘middle’

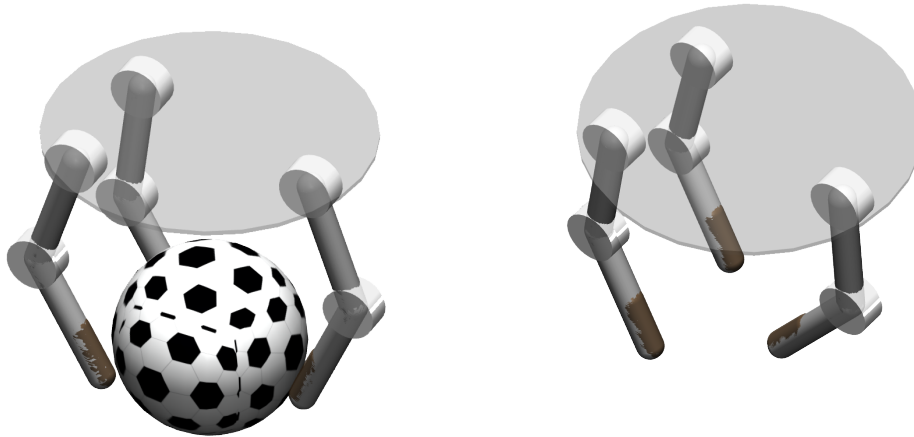
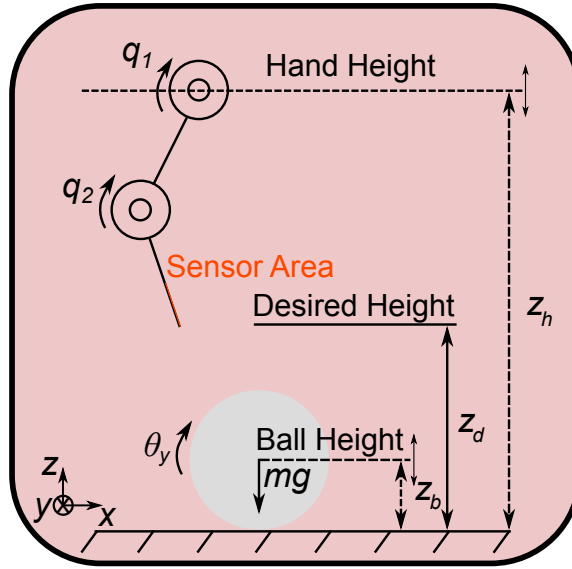


Figure 2.1: A three-finger robotic agent the MuJoCo environment.

fingers) and one opposing them (analogous to the ‘thumb’) (Figure 2.1). Each finger consisted of two joints that could rotate about the y -axis (q_1 and q_2 in Fig. 2.2), similar to the flexion or extension seen in human fingers. The size of the palm and length of each ‘phalanx’ was based on an average human hand [22, 27]. An additional servo motor was included at the base of the hand, which provides translational motion in the vertical direction (z_h).

Figure 2.2. depicts our simulated environment as well as the schematic of the hand. A three-finger robotic agent interacts with a ball in the MuJoCo environment via seven DOFs: two actuated rotational joints (q_1 and q_2) per finger plus the vertical position of the palm of the hand (z_h). This enables the agent to manipulate the ball by rotating it (θ_y) and lifting it (z_b) to a target height (z_t). To keep the ball’s motion in-plane, it is free to move vertically (z) and horizontally (x), and rotate in the plane (θ_y), but is lightly restricted in its translation in the lateral (y) direction and rotation about the (z) and (x) axes by a simulated stiffness. Viscous damping is applied to all of the ball’s translational and rotational degrees of freedom to slow down the dynamics and prevent numerical instabilities for the simulation of the rigid fingers and the ball.



Simulation Environment

Figure 2.2: Degrees of freedom (DOFs) of the hand: a three-finger robotic hand interacts with a 70 mm diameter ball in the MuJoCo environment (only one of the three fingers is shown for simplicity)

The robotic hand attempts to manipulate a 70 mm diameter, 5 gram ball, which starts each episode on the ground with the palm of the robotic hand at the height of 200 mm above the ground. The ball height z_b is defined at the center of the ball, and we specified a desired height for the ball z_d to be 25 mm above z_b . In other words, the desired height z_d is 60 mm above the ground. Through simulation constraints, the ball is limited to 2 translational DOFs (moving vertically z and horizontally x) and 1 rotational DOF (rotation about the θ_y direction; see figure 2.2). We included viscous damping in the translational and rotational DOFs of the ball to stabilize the simulation. We further limited the ball’s movement in the x direction by adding stiffness to the ball. We also use OpenAI baselines library and MuJoCo-py interface to implement reinforcement learning algorithms developed in Python. Physical parameters for all entities in the simulation must be specified (either directly or indirectly) including size, mass, stiffness, and damping. Relevant simulation parameters for the hand and ball are provided in

Table 2.1.

Entity	Parameter	Value
Hand	Mass	76 gram
	Finger Mass	17.8 gram
	Link length	50 mm
	Phalanx diameter	10 mm
	Palm width	20 mm
	Palm diameter	120 mm
	Initial hand height ($z_h(0)$)	200 mm
	Maximum translation (z_h)	130 mm
	Joint damping	5×10^{-6} N·s/mm
	Joint limits (q_1)	$[-45^\circ, 45^\circ]$
	Joint limits (q_2)	$[-90^\circ, 0^\circ]$
Ball	Mass	5 gram
	Diameter	70 mm
	Desired height (z_d)	60 mm
	Height (z_b)	35 mm
	Stiffness in x direction	1×10^{-3} N/mm
	Damping in x direction	1×10^{-4} N·s/mm
	Damping in z direction	1×10^{-3} N·s/mm
	Damping about y direction	5×10^{-4} N·s/rad

Table 2.1: Physical simulation parameters for the three-fingered robotic hand and the ball.

2.2.1 Tactile Information

We evaluate our algorithm performance in the presence four different sensory modalities. Our agent has four options for tactile information from the pad of each finger, which are added to the state vector. Tactile information is provided to the learning algorithm via the tactile force state vector for the simulated robotic hand ($\mathbf{s}_{\mathbf{h},\mathbf{f}}$). These are null (No-Tactile), Binary-contact (binary 1 vs. 0), force normal to the finger pad (Normal-Force, F_n), or the full 3D-Force vector ($\mathbf{s}_{\mathbf{h},\mathbf{f}} = [f_{t,1}, f_{t,2}, f_n]$). The tactile sites are only used on the internal side (i.e., the ‘pads’ of the fingertips) of the distal phalanx of each finger.

We used MuJoCo’s built-in features to record contact force magnitude (Normal-Force) (‘touch’) and 3D-Force sensing (‘force’) on the fingertips of all three fingers [52, 31]. Normal-force sensor sites at the soft fingertips provide a nonnegative scalar-value indicating the cumulative normal contact forces on the sensor area. The 3D-force sensor sites provide a 3D array of 3 orthogonal forces (one normal and two tangential to the sensor site for each sensor) of scalar values representing the 3D contact force vector. Binary sensors can only return one of two mutually exclusive values. Binary force sensing for the tactile information may report a switch on or off; when there is a contact, Binary tactile information will result in 1 as an output and 0 otherwise. In the No-tactile case, the state vector for the tactile information $s_{h,f}$ is null. The friction coefficient for dynamically generated contact pairs is also specified for all fingertips (a.k.a. soft contact with friction) [52].

As shown in Fig. 2.3, the possible contact tactile information at each fingertip is indicated by $\mathbf{s}_h, \mathbf{f} = [f_{t,1}, f_{t,2}, f_n]$ and it depends on tactile sensing available at fingertips. Table 2.2 indicates the detailed tactile information for the four tactile force sensing options.

Tactile Information in State Variable			
No-tactile	Binary-contact	Normal-force	3D-force
$s_{h,F} = 0$	$s_{h,f}$ $[\lceil \frac{1}{(1-\exp^{f_n})} - 0.5 \rceil, 0, 0]$	$= s_{h,f}$ $[0, 0, f_n]$	$= s_{h,f} = [f_{t1}, f_{t2}, f_n]$

Table 2.2: Tactile information options available to the learning algorithm.

The state of the system is 20-dimensional and consists of angles and velocities of all joints of the robotic hand as well as the position and velocities of the palm of the hand \mathbf{s}_h and position of the and velocity of the ball \mathbf{s}_b .

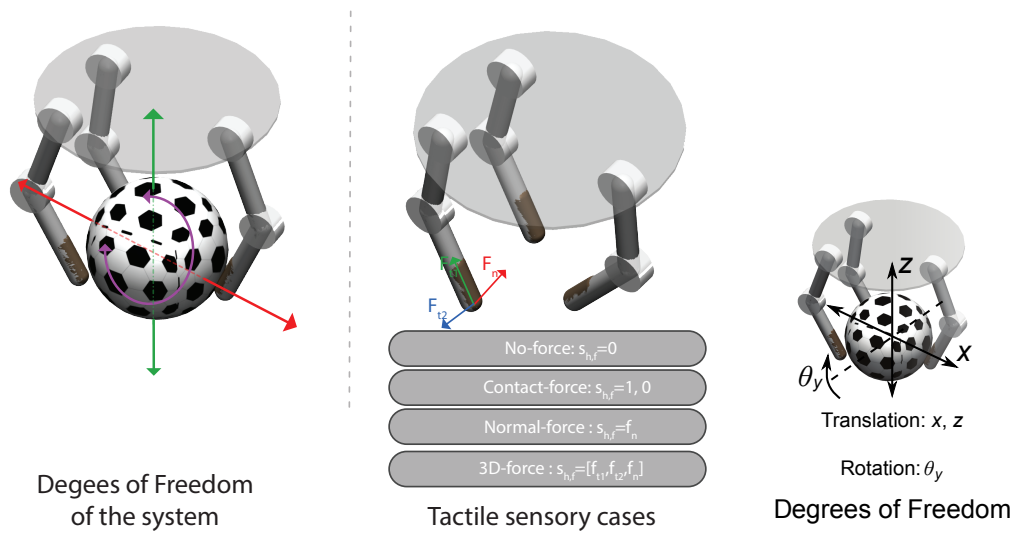


Figure 2.3: The ball can traverse on x and y directions with (without damping or friction) and rotate around the z axis (with damping and friction). The system can work with no tactile sensory, binary, normal force only, or both normal and tangential forces, which creates four cases for the tactile sensory conditions. These are null (No-tactile), contact or no contact (Binary-contact, 1 or 0), force normal to the finger pad (Normal-force, f_n), or the full contact force vector (3D-force, $f = [f_{t,1}, f_{t,2}, f_n]$).

Chapter 3

Reinforcement Learning Algorithm for dexterous manipulation

3.1 Chapter summary

We confront the particularly challenging task of dynamic manipulation, where the fingertips of a downward-facing hand interact with a ball to both lift and rotate it, and the object is at risk of falling at any time. Our key finding is that this challenging task can be learned autonomously using the industry-standard PPO algorithm, but only if a correct strategy (i.e., curriculum) is used. To study the effects of learning strategies, we investigated different combinations of rewards function during the learning phase. In this chapter, we demonstrate our data-driven learning approach which does not require an explicit model of the hand, ball or hand-ball interactions, as this is most critical in unstructured environments (cf. model-based approaches).

3.2 Model-free RL algorithm

The concept of reward engineering is developing a reward scheme to inject a notion of success into the system, which is at the core of RL [53]. Reward shaping involves carefully designing reward functions that provide the agent with rewards for progress towards the goal. In the next subsection, we discuss about the details on the reward function.

3.2.1 Details on the reward function

In our algorithm, the goal is reached when the agent rotates the ball while keeping it against gravity between target height span which is ± 25 percentage of that desired height (25 mm). Since we care about manipulating (rotating to be specific) the ball against gravity at the desired height range (between [18.75, 31.25] mm), we used a combination of primary (positive) reward and punishment (penalty or negative reward) at every time step.

Angular velocity of the ball $\dot{\theta}_y$ would be the primary reward, and the absolute distance of the state from the reference state of having the ball at the fixed desired position ($z_d = 25$ mm, Fig. 2.2) would be the punishment. We choose the dynamical manipulation task of rotating and lifting along a horizontal rotation axis at the desired height (and adding a penalty proportional to the distance between the current height and the desired height). The reward function is described by

$$Reward_t = c_R \dot{\theta}_{y,t} - c_L |z_{h,t} - z_d|, \quad (3.1)$$

where $c_R = 0.51$ and $c_L = 0.49$. These coefficients change when we wanted to focus on only on aspect of the task (rotation or lifting; see details on the Table 3.1). Although we use dynamical manipulation task of rotating as a primary reward, to make the interpretation of the results easier we calculated completed round of rotations as our performance measurement—instead of reward rotation itself. Not all of the simulated state variables

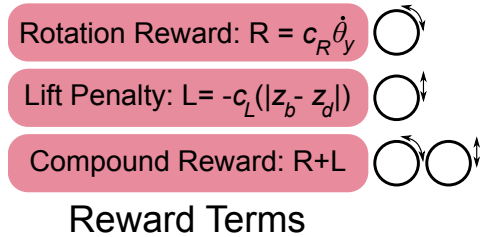


Figure 3.1: Our reward is a combination of **i**) rewarding the angular velocity of the ball $\dot{\theta}_y$ in a positive direction (primary reward); and **ii**) penalizing the distance between the ball and desired heights $|z_d - z_b|$ (penalty reward)).

are used in the PPO policy. Although the ball’s state is used for the reward function, it is not passed into the PPO policy for learning. In other words, the dynamic interaction with the ball happens in the absence of vision (or any other direct, constant monitoring means for the balls velocity or position).

3.3 Learning the policy

We use a model-free RL algorithm to learn the policy to autonomously learn in-hand manipulation of a ball against gravity through utilizing tactile sensory information. The proposed algorithm consists of our agent interacting with an environment (ball) and receiving a reward depending on the action. This loop continues until the learning period ends. The algorithm’s primary purpose is to learn a behavior that maximizes the accumulated reward (lifting+rotation reward). We used end-to-end Proximal Policy Optimization (PPO) as our main autonomous learning algorithm.

3.3.1 Proximal Policy Optimization Algorithm

PPO is a set of policy gradient methods that optimize a surrogate objective function using multiple minibatch updates per data sample [50, 54]. The objective function to

optimize is the sum of several loss functions and is given by

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)], \quad (3.2)$$

The $L_t^{CLIP}(\theta)$ is the surrogate objective function and ensures that the policy updates will not be too large. While the L_t^{VF} is a squared-error loss, it ensures that the loss from both policy and value functions of the neural networks are accounted for. The S denotes the entropy bonus term, which encourages a more random policy (i.e., more exploration), so a larger entropy coefficient c_2 will encourage more exploration [54].

To implement PPO, we use the PPO1 implementation from OpenAI’s stable baselines repository [55] with MultiLayer Perceptron (MLP) Artificial Neural Network (ANN) as for the actor-critic map.

To autonomously learn in-hand manipulation of a ball against gravity through utilizing tactile information, we use a model-free RL algorithm to learn the policy. We used PPO as our main algorithm as it seeks to find a balance between the ease of implementation, sample complexity, and ease of adjustment, trying to update at each step to minimize the cost function while assuring that the new policies are not too far from last policies [50, 56]. It has also been adopted as one of the default methods OpenAI owing to its excellent performance [57, 55].

At every time step t , the robotic hand observes the state of the hand $\mathbf{s}_{h,t}$ and the state of the ball $\mathbf{s}_{b,t}$, predicts the optimized action, executes it \mathbf{a}_t , and a reward is used r_t . The state $\mathbf{s}_{h,t}$ contains the angle and angular velocity q_t, \dot{q}_t of each finger and the position and linear velocity of the palm at every time step t . The overview diagram of the Proximal Policy Optimization algorithm in this work is shown in Figure 3.3b. Each Curriculum was evaluated for 60 independent MC runs, which were repeated for the four tactile conditions. Each independent MC is chosen with 60 different seeds. For each tactile information, the initial seed for the random number generator was held constant across

different curricula. In other words, the first MC run seed was exactly the same for all curricula. For example, Curriculum 1, Normal-force MC run 25, has the same parameters as Curriculum 3, 3D-force MC run 25.

To achieve a level of performance that is desirable, in the process of training our RL model, we tune the PPO hyperparameters. The clipped surrogate loss of the PPO algorithm prevents divergence, as discussed in [50]. Although, it may prematurely shrink exploration variance when performing the updates over multiple iterations. PPO also adds an entropy loss term that penalizes low variance to prevent these issues with low variance and premature convergence. It showed that higher entropy loss weight minimizes the risk of getting stuck in the local optimum. However, if entropy loss weight is too big, it may result in noisy policy and deteriorated average performance. Therefore, fine-tuning the entropy loss term for PPO can become tricky. Based on the results of different entropy loss weights for policy’s standard deviation in [58], we optimized the entropy loss term to find a balance between variance and average performance.

Additionally, the entropy loss meta-parameter is tuned to find a balance between variance and average performance. PPO uses the generalized advantage estimator (GAE) to significantly reduce the variance of policy gradient estimates at the cost of some tolerable level of bias. GAE is parameterized by $\lambda \in [0, 1]$ which enables the PPO agent with a mechanism to control policy updates according to the significance of each sampled state and therefore enhance learning reliability [59]. Changing this hyperparameter enables PPO to find a balance between variance and bias of policy gradient estimates [60]. In our work, this trade-off was achieved by changing the lambda meta-parameter to relatively demote rewards achieved later in the episode (when the ball may have been dropped) and instead emphasizing immediate rewards at every point in time (as is the case in real life).

The number of optimization epochs, GAE parameter λ , and the entropy coefficient are

set to values shown in Table 3.1. All other parameters are kept at their default values per PPO implementation defaults. The hyperparameters for the PPO algorithm are listed in Table 3.1. Hyperparameters that are not defaults have been chosen empirically (trial and error and carefully going over resulting performances).

Hyperparameter	Value
Adam stepsize	1×10^{-5}
Number of epochs	8
Discount (γ)	0.99
Entropy coefficient	0.02
Advantage estimation (λ)	0.85
Minibatch size	64

Table 3.1: Proximal-policy optimization (PPO) hyperparameters

3.4 Overview of Simulation Environment and Learning Trail

A learning trail starts with the ball on the ground and hand and fingertips suspended above the ball. Each MC run consisted of 2,000 episodes lasting 10 s sampled at 0.01 s (left), which leads to 5-hours and 33-minutes of simulated time. Each learning trial was split into two equal halves where the reward function changed between the two halves of the MC run. As it is shown in Figure 3.2 and 3.3 (b) in each episode, the agent takes an action a in state s and receives reward r . It also observes that the state has changed to a new state s' .

At the beginning of every episode, we define the curriculum reward and tactile information and the PPO seeding. Each training period consists of one independent Monte Carlo run, which has 2,000 training episodes. At the start of each Monte Carlo run, the PPO policy is provided a random seed, and the policy is updated throughout the

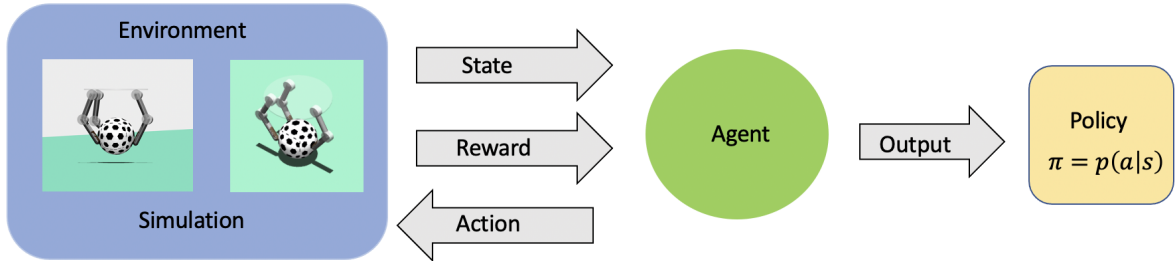
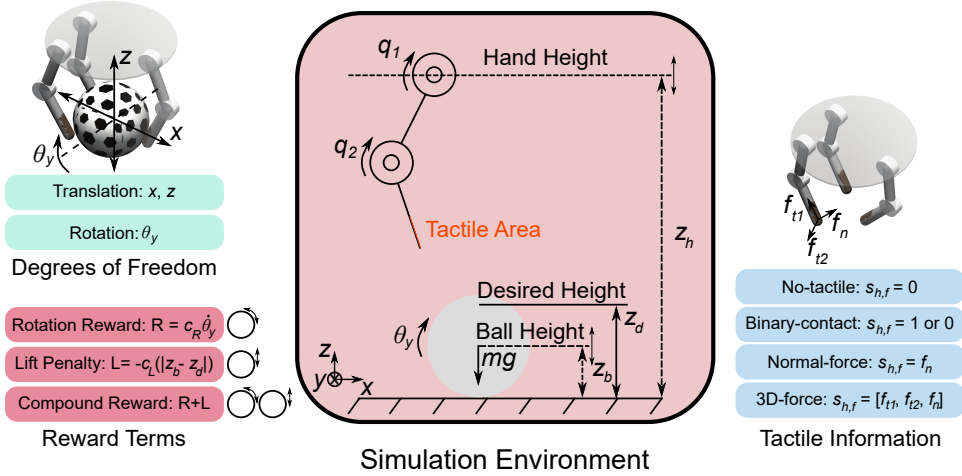


Figure 3.2: Basic process of learning dexterous manipulation by RL.

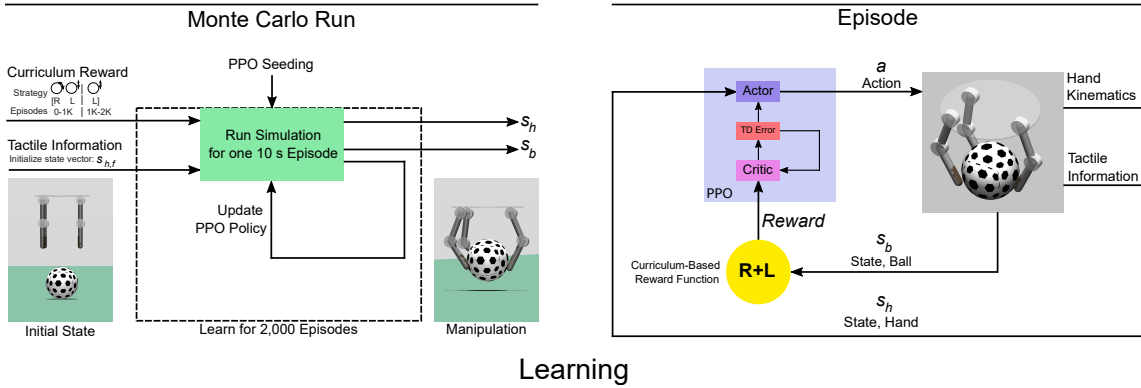
2,000 training episodes. (Figure 3.3 (b) left). In our implementation, the newest policy parameters from the optimizer at the beginning of every episode are updated and used to generate the next training episodes.

3.5 Curricula

Learning in a meaningful order (usually a cascade of tasks of increasing complexity, or curriculum learning), is a cornerstone of biological learning that has been transferred and applied to machine learning and artificial intelligence [61, 62]. Our approach leverages curriculum learning, training the robot with different subtasks to autonomously learn the desired in-hand dynamic manipulation. In each MC run for each curriculum, the reward function changed after 1,000 episodes—as shown via rotation and lift icons in Table 3.2. We considered five distinct curricula that are differed in the behavior (rotation and lift) rewarded in two halves of MC runs. This is illustrated by a circle with a curved arrow (rotation) and a vertical arrow (lift) throughout the paper and pictured in the second column of the Table 3.2). As shown in the last column from Equation (3.1) by changing c_R and c_L variables, we update the reward function in two equal half of the MC run in each curriculum.



(a) *Degrees of freedom (DOFs) of the hand and ball:* A three-finger robotic hand interacts with a ball in the MuJoCo environment. *Reward Options:* Instantaneous reinforcement comes from the combination of **i**) rewarding the angular velocity of the ball $\dot{\theta}_y$ in a positive direction (primary reward); and **ii**) penalizing the distance between the ball and desired heights $|z_d - z_b|$ (penalty reward). *Tactile Information:* The learning policy has four sensing options for tactile information from the pad of each finger (Tactile Area), which are added to the state vector for the hand (s_h). These are null (*No-tactile*), contact or no contact (*Binary-contact*, 1 or 0), force normal to the finger pad (*Normal-force*, f_n), or the full contact force vector (*3D-force*, $\mathbf{f} = [f_{t1}, f_{t2}, f_n]$).



(b) In each MC run for each curriculum, the reward function changed after 1,000 episodes—as shown via rotation and lift icons in all results figures.

Figure 3.3: Overview of simulation environment and learning.

3.5.1 Details on the curriculum

To autonomously learn in-hand manipulation, we used combinations of reward functions to emphasize one or both critical features of manipulation: lifting and rotating the ball. Instantaneous reinforcement comes from the combination of **(i)** rewarding the angular velocity of the ball $\dot{\theta}_y$ in a positive direction (primary reward); and **(ii)** penalizing the distance between the ball and target height $|z_t - z_b|$ (penalty reward) (Figure 3.3). These reward functions define the agent’s goal during two 1,000 episode learning phases.

To study the effects of learning strategies, we used five curricula (Table 3.2), each rewarding different combinations of rotation and lift during each half of a Monte Carlo (MC) run. The coefficients of the reward function (c_R and c_L in equation (3.1)) explicitly weight rotation (R) and lift (L) rewards, respectively (Fig. 3.3b). Based on the fact that our algorithm is rewarded for lifting and rotating, the learning phase is divided into two phases, the first 1,000 episodes and the second 1,000 episodes, in which each phase is 10,000 seconds. We made different curricula by changing these coefficients at the halfway point of the MC run (e.g., Curriculum 5 is [R+L|L]). Note these curricula were executed as independent MC runs for each of the four sensing options for tactile information from the pad of each finger (Fig. 3.3).

















	Reward During First Half of the MC Run	Reward During Second Half of the MC Run	Coefficients of equation (3.1) for the [first second] halves of each MC Run
Curriculum 1		 	[L R+L] [$c_R = 0, c_L = 0.49$ $c_R = 0.51, c_L = 0.49$]
Curriculum 2		 	[R R+L] [$c_R = 0.51, c_L = 0$ $c_R = 0.51, c_L = 0.49$]
Curriculum 3	 	 	[R+L R+L] [$c_R = 0.51, c_L = 0.49$ $c_R = 0.51, c_L = 0.49$]
Curriculum 4	 		[R+L R] [$c_R = 0.51, c_L = 0.49$ $c_R = 0.51, c_L = 0$]
Curriculum 5	 		[R+L L] [$c_R = 0.51, c_L = 0.49$ $c_R = 0, c_L = 0.49$]

Table 3.2: We used five curricula that rewarded different combinations of rotation and lift during each half of a Monte Carlo (MC) run. These changes in the coefficients of the reward function define a progression of goals (i.e., curriculum learning) over the two halves of each run.

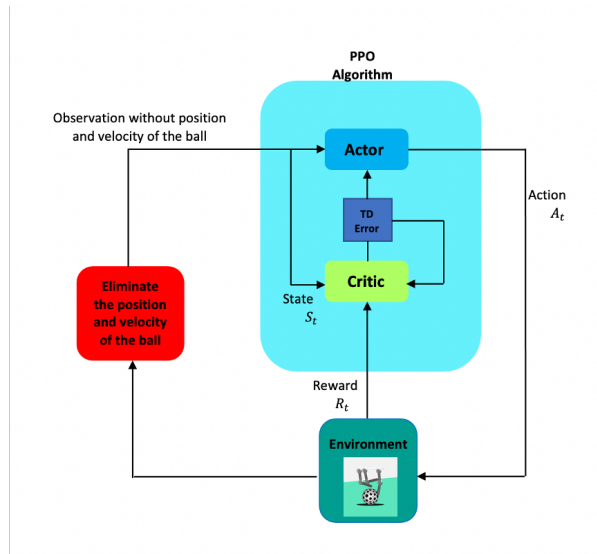


Figure 3.4: The overview diagram of Proximal Policy Optimization (PPO) algorithm for in-hand manipulation. The ball’s state is not passed into the PPO policy for learning.

Chapter 4

Identifying useful learning strategies

4.1 Chapter summary

Using a three-finger robotic hand in simulation, we demonstrate what, to our knowledge, is the first example of autonomous learning to pick up and manipulate an object against gravity without vision. The in-hand manipulation is possible by choice of useful curricula, complemented by meta-parameters in the PPO algorithm to emphasize immediate rewards. In this chapter, we investigate the importance of each Curriculum to lead to distinct final performances.

4.2 The Effects of Learning Strategies

Using the RL algorithm, a curriculum-learning-based approach adopted in our work, we demonstrate *autonomous learning of autonomous dexterous manipulation* in simulation with gravity. The primary focus of our work is (1) how the influence of the right strategy (Curriculum) impacts the simulated three-fingered robotic agent’s ability to learn to manipulate (lift and rotate) a ball autonomously. To study the effects of the right learning strategies, we used five curricula (Table 3.2), each with different rewarding

combinations of rotation and lifted during each half of a Monte Carlo (MC) run. Each Curriculum was evaluated for 60 independent Monte Carlo runs, which were repeated for the four tactile conditions. Between phases of training (phase 1: episodes 1-1,000 and phase 2: 1,000-2,000), the reward function generally changed depending upon which of the five Curricula was being run. The coefficients of the reward function (c_R and c_L in equation (3.1)) explicitly weight rotation (R) and lift (L) rewards, respectively (Fig. 3.3b). We made different curricula by changing these coefficients at the halfway point of the MC run (e.g., Curriculum 1 is [L|L+R]).

4.2.1 Curriculum 1

Since our algorithms learn in-hand manipulations in the context of rewards received from interactions with the ball. Reward functions play a central role in specifying how an agent should act. In this Curriculum, the hand is expected to explore lifting first and then explore both rotation and lifting.

In this Curriculum, we changed the weight rotation (R) and lifted (L) rewards at the halfway point of the MC run such that it is indicated as [L|L+R]. In this Curriculum, the rotation is not rewarded from the start for learning in-hand manipulation. Based on the results in Curriculum 1 in Figure 4.1, the results show that the hand failed to lift the ball or rotate it a full revolution in all MC runs.

4.2.2 Curriculum 2

In this Curriculum, at the halfway point of the MC run, the weight reward is changed such that the rotation given in the first half. This Curriculum is indicated as [R|L+R]. By comparing the results of These, we demonstrate that while lifting the ball seems like a ‘simpler’ task, it does not appear to be possible before the system ‘understands’ the dynamics of manipulation. First, our agent plays with the object to understand the

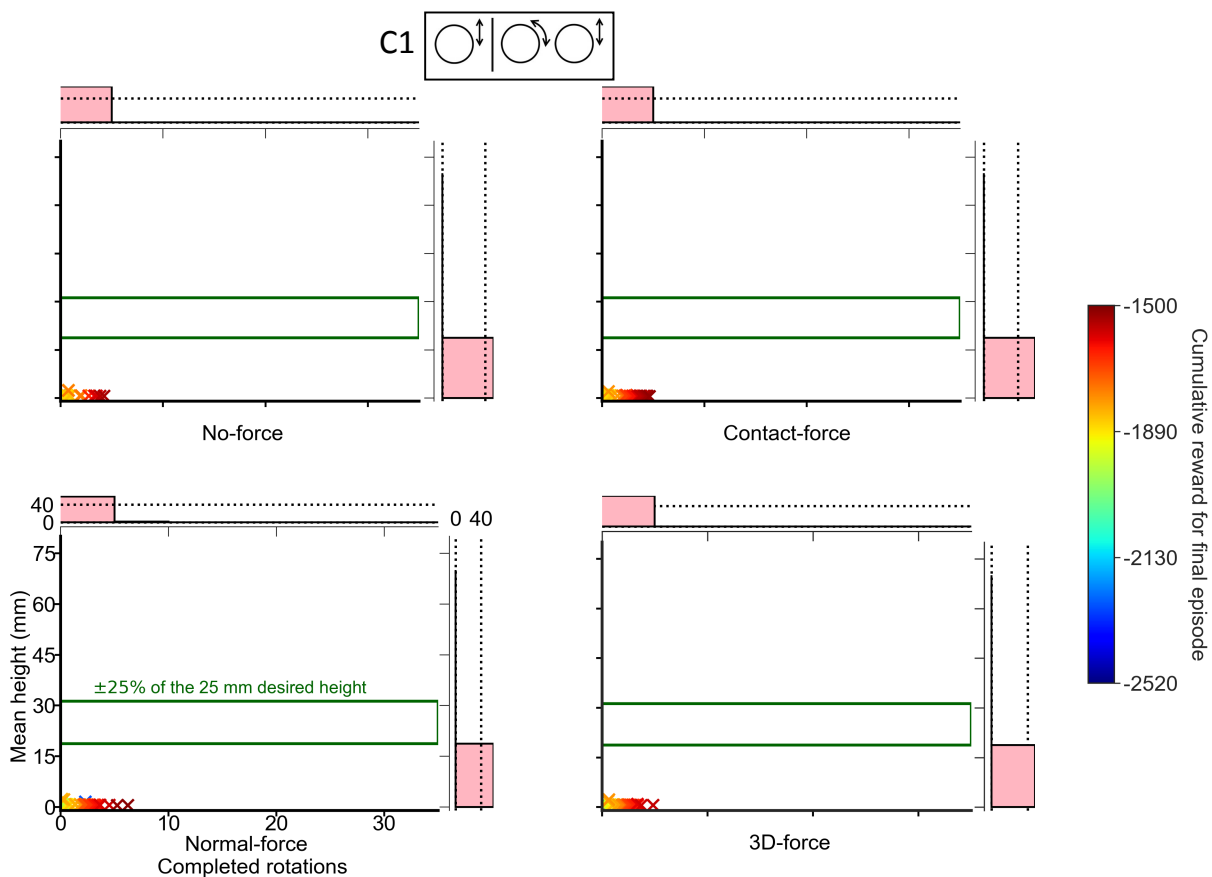


Figure 4.1: Performance for Curriculum 1 [L| R+L] for all options of tactile information. The joint distribution of mean ball’s height (mm) vs. the number of completed rotations in each Monte Carlo run for the last episode is shown. Each Monte Carlo run is color-coded based on the range of reward from lowest to highest reward among all tactile conditions. Curriculum 1 failed to lift the ball or rotate it in all MC runs. These results highlight the importance of the choice of Curriculum targeting a particular performance goal. For example, rotating the ball is most critical for the performance of a task.

dynamics of the world around it, similar to what we do when we are toddlers [4].

By comparing the results of these curricula, we realized that manipulation is the significant first step to a successful lifting. By rewarding (and therefore encouraging), the exploration of the full dynamics of manipulation, the agent is enabled to learn the combined manipulation task of rotating and lifting the ball. (cf. Figures 4.1 and 4.2).

The violin plots in Figure 4.3 show the distribution of completed rotations during learning phase. It shows that the hand gets most of its reward via rotation since lifting is incorporated into the reward function only at the second half of the MC run. The boxplots of total aggregated reward this curriculum is shown in Figure 4.4.

4.2.3 Curriculum 3

In this Curriculum, we don't change the weight rotation (R) and lifted (L) rewards at the halfway point of the MC run such that, so it is indicated as [L+R|L+R]. According to this curriculum, our agent tries to achieve a goal based on its defined reward function (the goal is to rotate the ball while keeping the ball's height within the target height range). Furthermore, Curriculum 3, which might be considered the 'best' in terms of manipulation is a compromise for performance that achieves a balanced lift and rotation manipulation schema (Figure 4.5). The joint distribution of performance during the last episode of 60 MC runs (mean ball's height (mm) vs. the number of completed rotations) is shown in this figure. Surprisingly, however, Contact-force produces the most MC runs with high rewards inside the desired height range. 3D-force, had the least number of MC runs that were caught against the palm.

In Curriculum 3, the mean number of data points inside the target range for all options of tactile information is 25, while it has 14 numbers of completed rounds of rotations. In all Curricula except Curriculum 3, the reward function evolves and changes over the learning period (phase 1 and phase 2). In Curriculum 3, lifting and rotating are

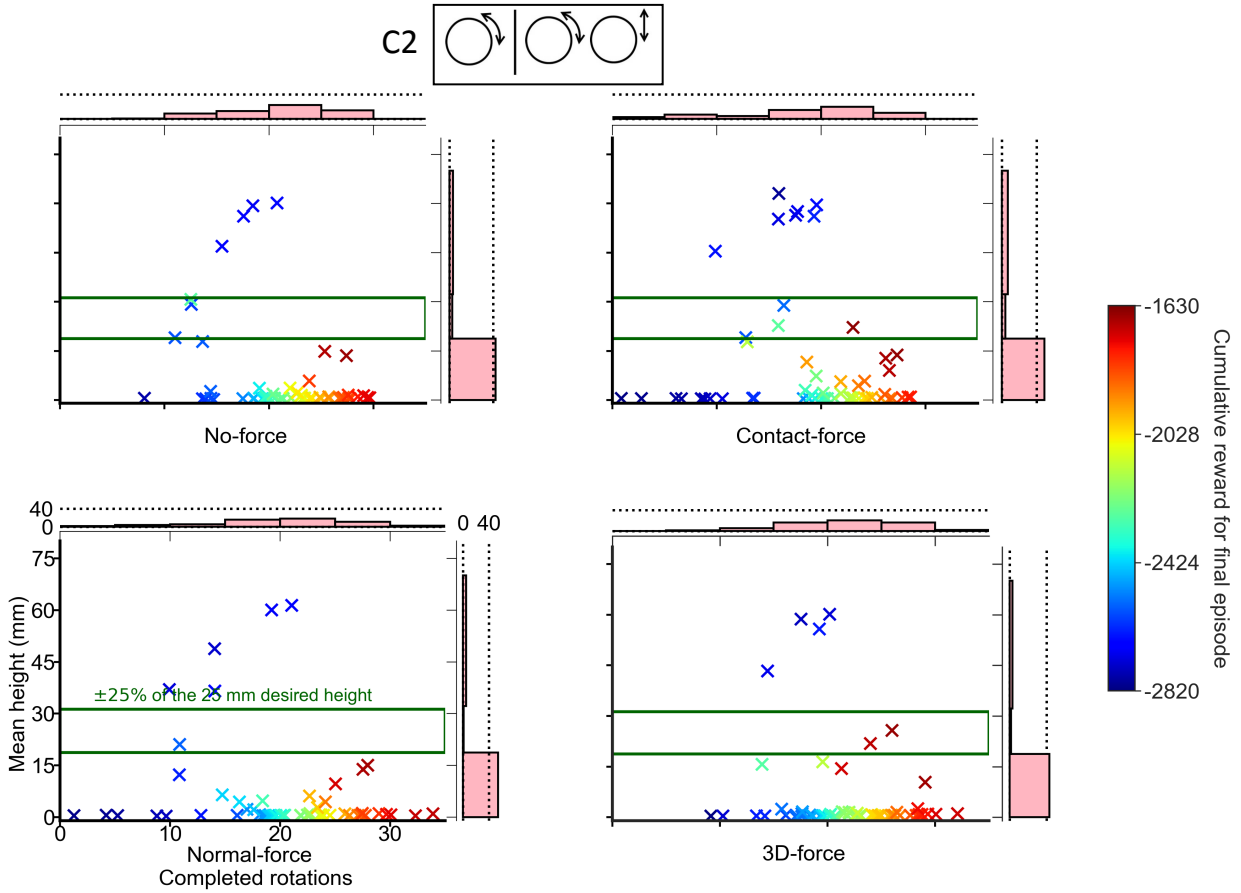


Figure 4.2: Performance for Curriculum 2 [R| R+L] for all options of tactile information. The learning policy enabled the robotic hand to rotate the ball in the first 1,000 episodes. After changing the reward to both rotation and lift, the hand still gets most of its reward via rotation since lifting is incorporated into the reward function only at the second half of the MC run. Although in general the robotic hand is not successful in lifting the ball in all MC runs, several instances of manipulation (both rotation and lift) are demonstrated.

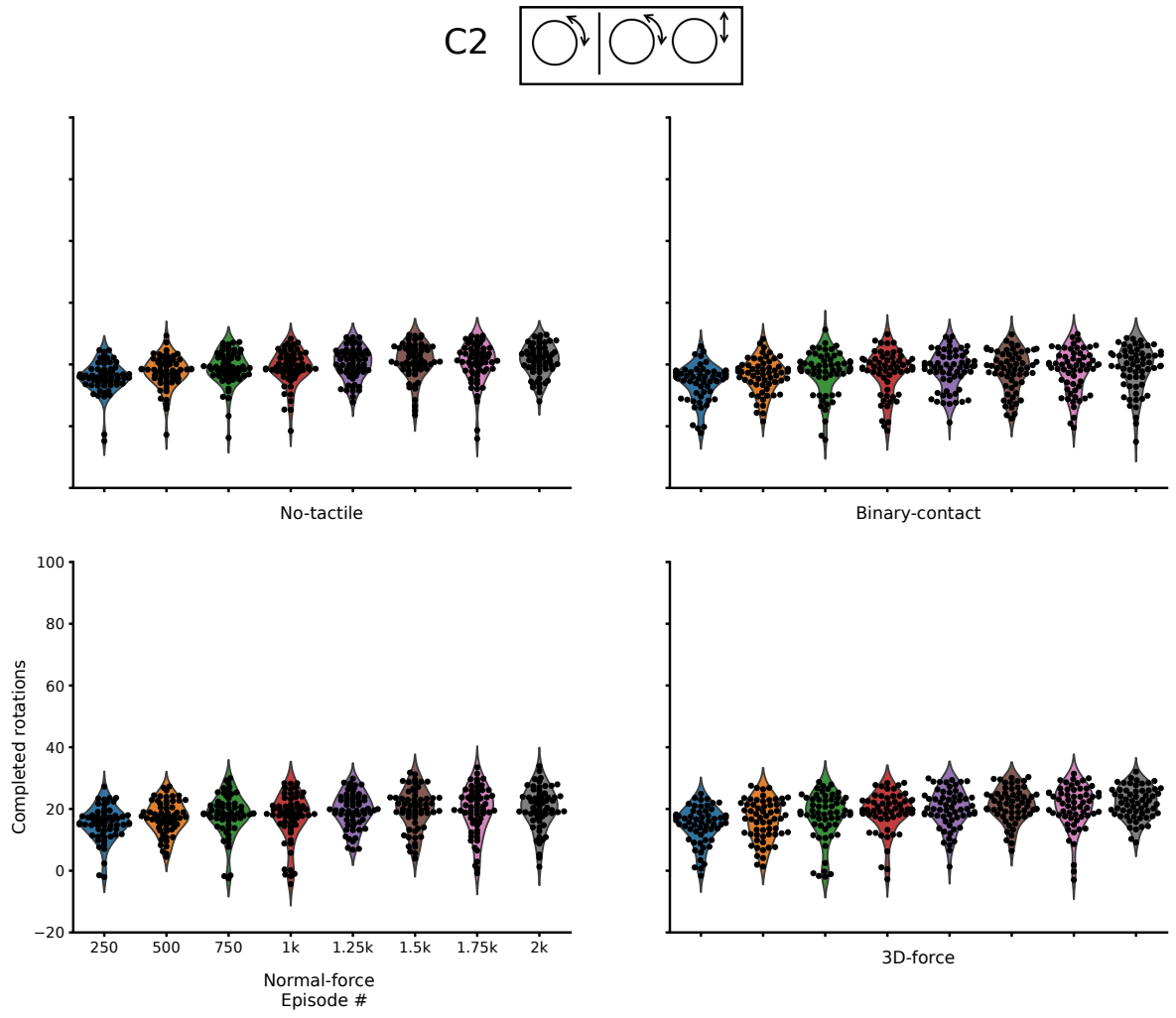


Figure 4.3: Performance for Curriculum 2 [R|R+L] for all options of tactile information. It shows the violin plot of completed rotations for each Monte Carlo run at every 250 episodes. The width of the plot indicates the frequency of the corresponding completed rotations and black points are shown each Monte Carlo runs.

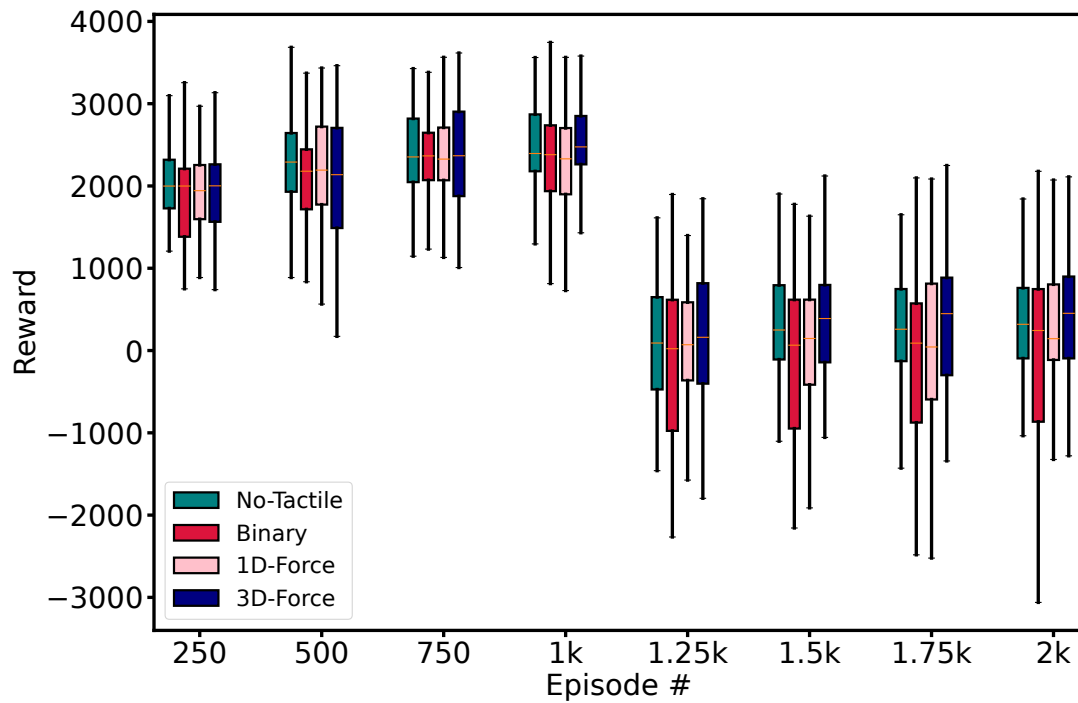


Figure 4.4: Performance for Curriculum 2 [R|R+L] for all options of tactile information. Boxplots of the aggregated reward for tactile information while the boxes' orange lines show the median values, the edges of the boxes are the 25th and 75th percentiles. The reward gained per 2,000 episodes is visualized as boxplots grouped by the each tactile information aggregated over 60 Monte Carlo runs per box.

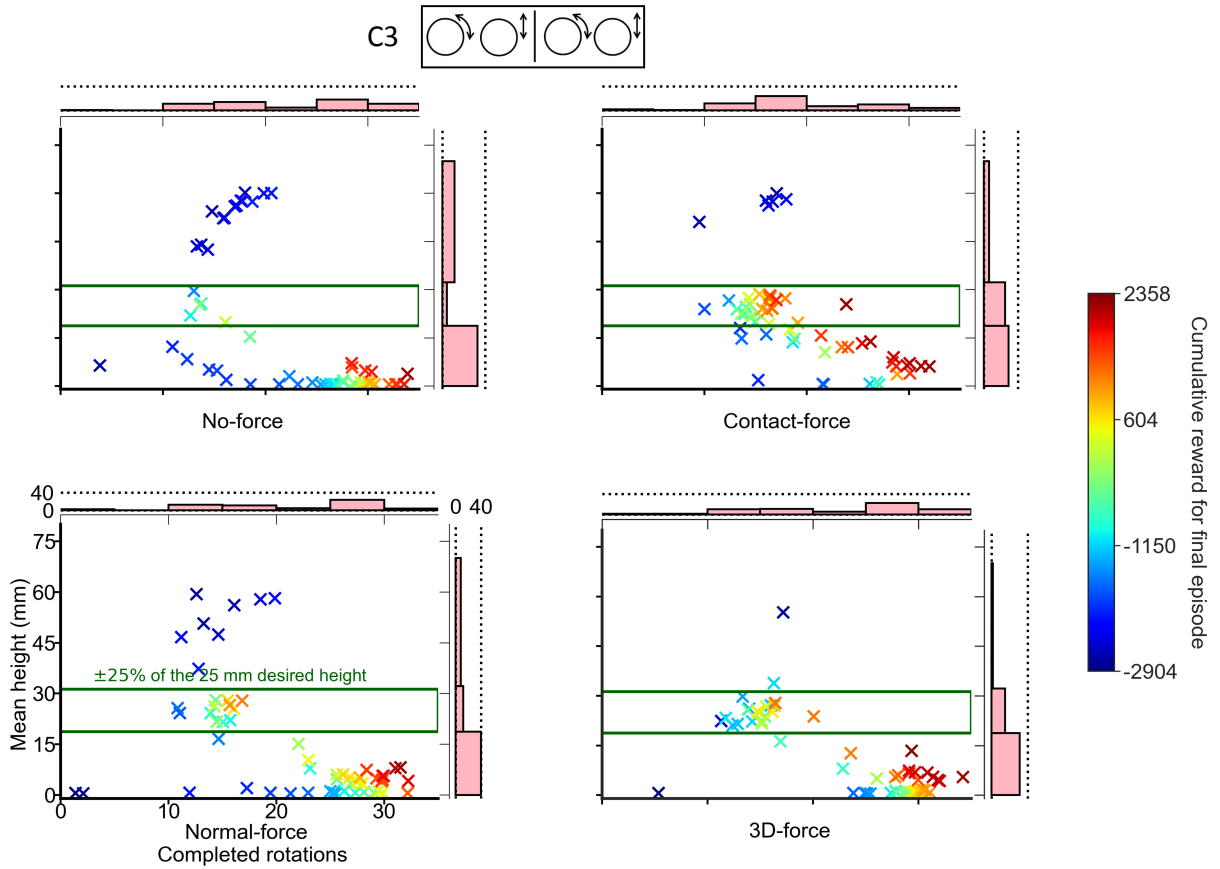


Figure 4.5: Performance for Curriculum 3 [R+L|R+L] for all options of tactile information. The cumulative reward for the last episode of each MC run is color coded. Note that the desired manipulation performance is represented by those points inside the green box defining the desired ball height. Note the ceiling and floor effects for mean height as the ball can get caught against the palm, or be rolled against the ground, respectively.

rewarded over the entire learning period (2,000 episodes). We demonstrate how changing the reward function during two phases in each curriculum affects the agent’s resulting manipulation performance. As it is shown in Figure 4.6, Contact-force and 3D-force are relatively equivalent in their learning rates; and better than No-tactile and Normal-force for Curriculum 3. The number of Monte Carlo runs with the highest rewards range (dark red hue) is achieved in Contact-force and 3D-force. Despite the fact that aggregated reward of lifting and rotating for Contact-force and 3D-force are similar, the total reward is achieved by (i) keeping the ball at a target height range for Binary, while for 3D-force it is achieved by (ii) rotating the ball on the ground (Figure 4.6).

The number of Monte Carlo runs with the highest rewards range (dark red hue) is achieved in Contact-force and 3D-force. Despite the fact that aggregated reward of lifting and rotating for Binary and 3D-Force are similar, the total reward is achieved by (i) keeping the ball at a target height range for Binary, while for 3D-force it is achieved by (ii) rotating the ball on the ground (Figure 4.6).

The resulting cumulative rewards plotted along with boxplots at eight representative episodes in Figure 4.7. All runs during different episodes showed the aggregated reward of Binary, and 3D-Force are similar.

4.2.4 Curriculum 4

In this Curriculum, at the halfway point of the MC run, the weight reward is changed such that the lifting is not given in the second half is indicated as [R+L|R]. Although the second phase of the learning does not incorporate lifting reward, we still see some successful runs with keeping the ball at the target height range in Contact-force and 3D-force. For instance, by switching off the lifting reward the second phase of Curriculum 4, we still see success in keeping the ball in the target range and rotating it in No-tactile, Contact-force and 3D-force. Interestingly, Curriculum 4 is not rewarded for lifting at the

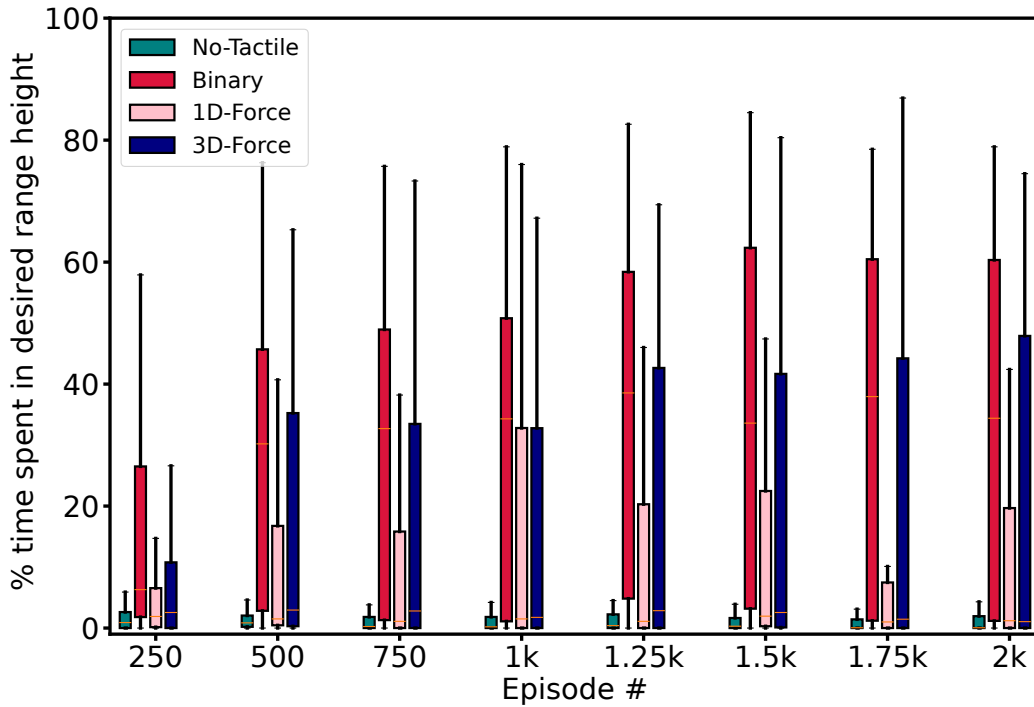


Figure 4.6: Performance for Curriculum 3 [R+L|R+L] for all options of tactile information. Lift success rate (the percentage of time the ball spent within 25% of the desired height over the duration of an episode). Boxplots, with median, across tactile information options for 60 MC runs at eight representative episodes, 250 episodes apart. Lift success rate (the percentage of time the ball spent within 25% of the desired height over the duration of an episode).

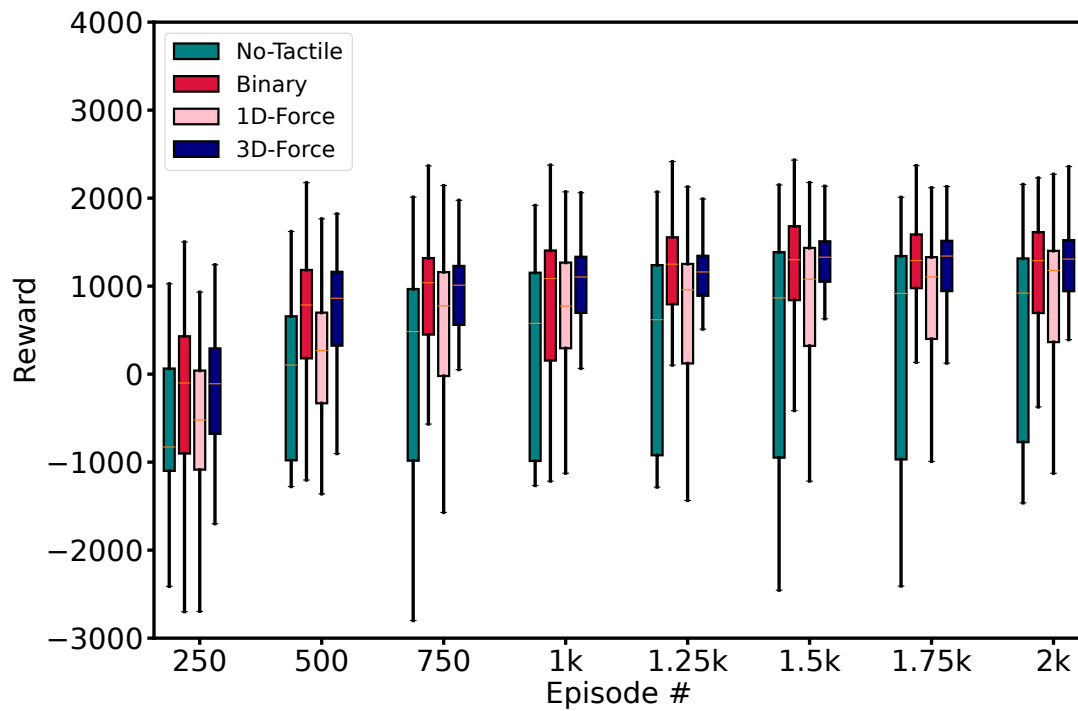


Figure 4.7: Performance for Curriculum 3 [R+L|R+L] for all options of tactile information. Cumulative reward for each representative episode. Boxplots of the aggregated reward for tactile information while the boxes' orange lines show the median values, the edges of the boxes are the 25th and 75th percentiles. The reward gained per 2,000 episodes is visualized as boxplots grouped by the each tactile information aggregated over 60 Monte Carlo runs per box.

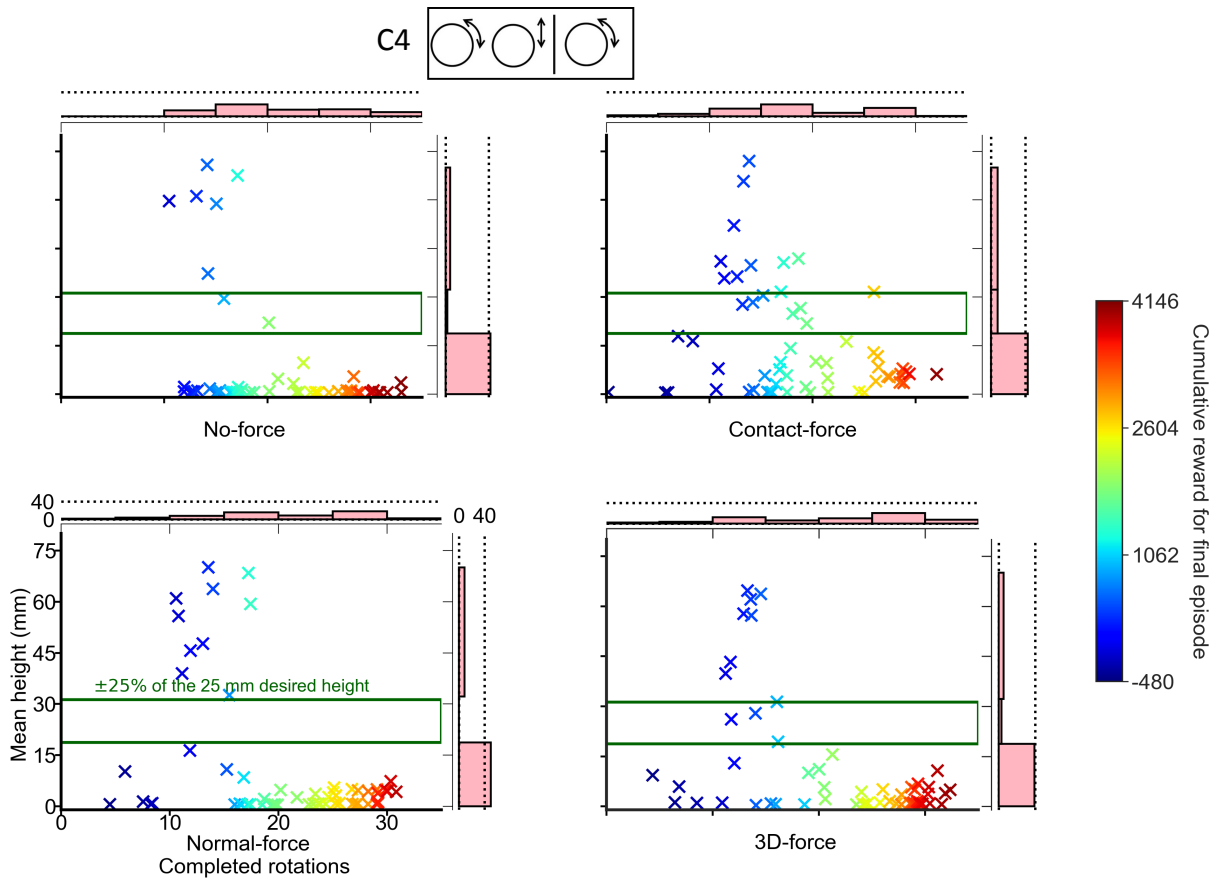


Figure 4.8: Performance for Curriculum 4 [R+L|R] for all options of tactile information. The lack of reward for lifting in the second half of the 60 MC runs focuses the robotic hand on rotating the ball in its final performance. Surprisingly, the robotic hand continues to lift the ball even if not rewarded for it.

end of its curriculum but occasionally lifts as well.

We plotted the results on a violin plot in Figure 4.9 to represent the distribution of completed rotations obtained at every 250 episodes. The width of the shape represents the frequency of the number of the completed rotations. Figure 4.9 shows that the violin plot of all tactile information be wide and short, which means that there is little difference between the best and worst tactile information.

As it is shown in Figure 4.10, there is an increase in the total reward for all levels of tactile sensory information at the second phase as the reward function does not

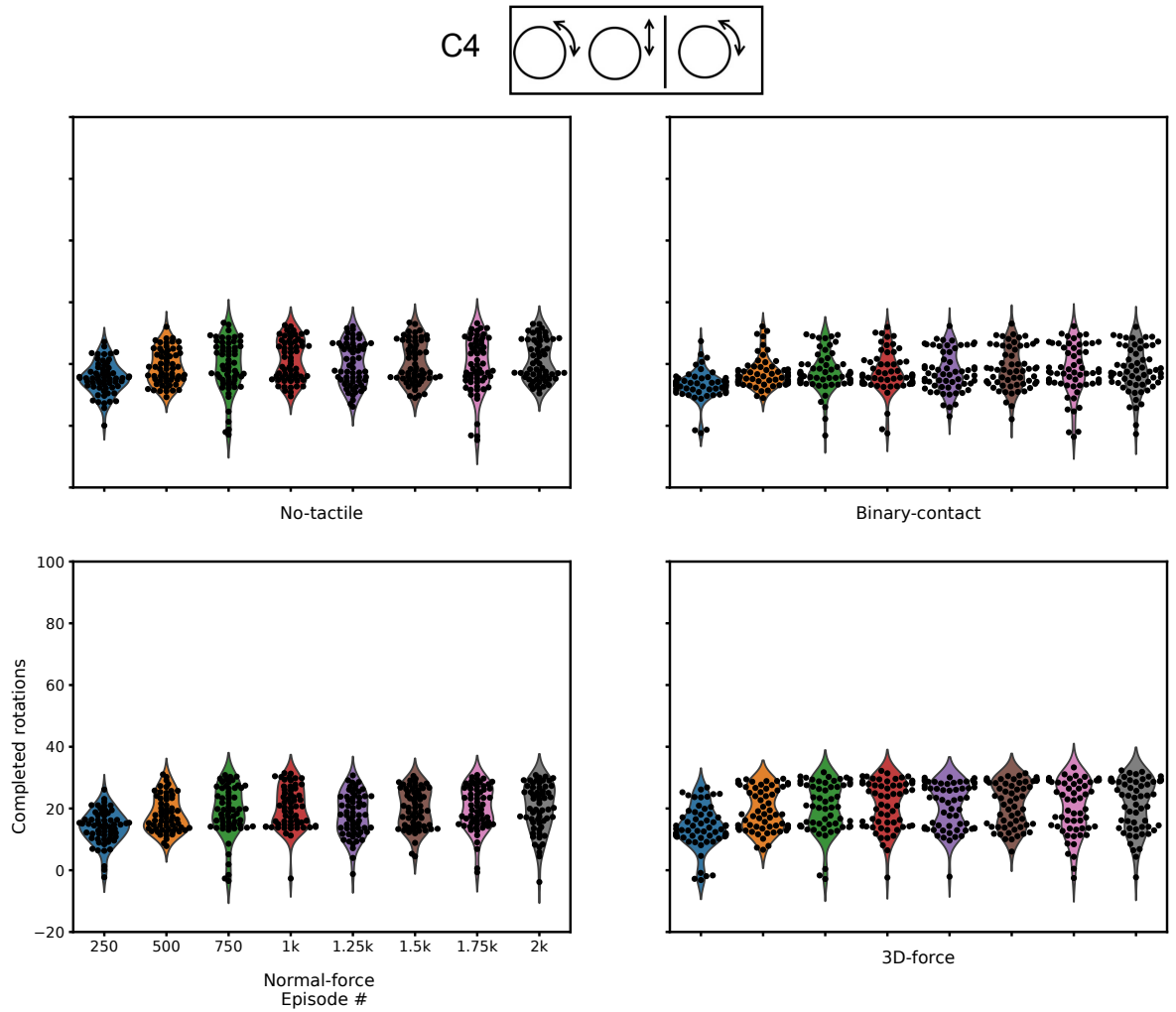


Figure 4.9: Performance for Curriculum 4 [R+L|R] for all options of tactile information. It shows the violin plot of completed rotations for each Monte Carlo run at every 250 episodes. The width of the plot indicates the frequency of the corresponding completed rotations. Each black point represents one Monte Carlo run.

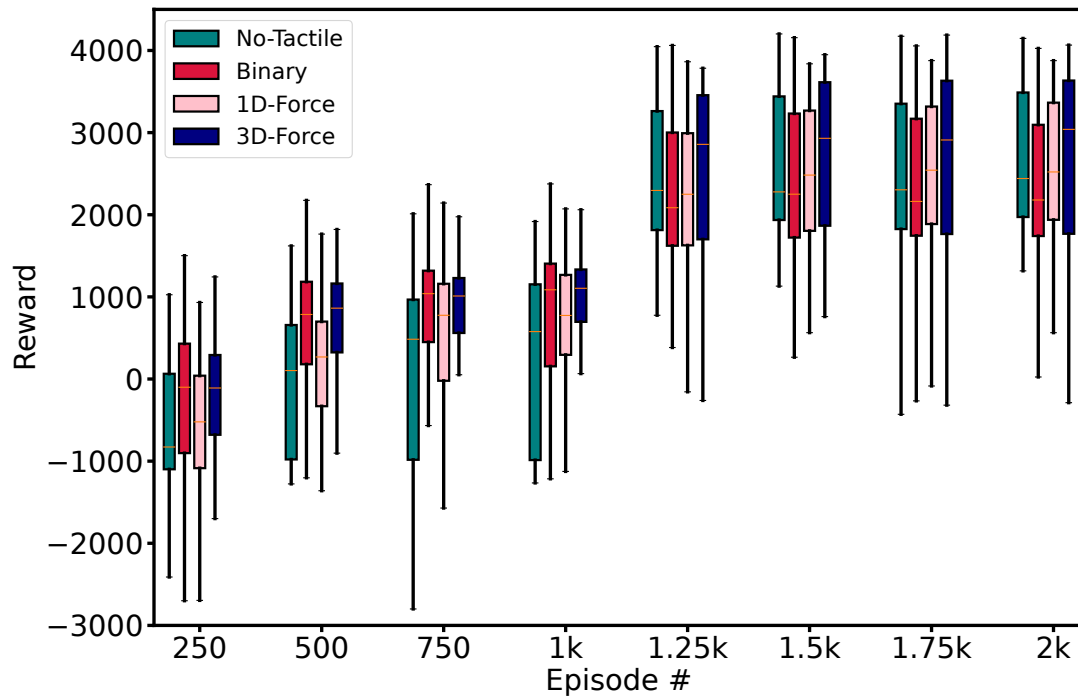


Figure 4.10: Performance for Curriculum 4 [R+L|R] for all options of tactile information. Boxplots of the aggregated reward for tactile information while the boxes' orange lines show the median values, the edges of the boxes are the 25th and 75th percentiles. The reward gained per 2,000 episodes is visualized as boxplots grouped by the each tactile information aggregated over 60 Monte Carlo runs per box.

incorporate lifting reward.

4.2.5 Curriculum 5

In this Curriculum, we changed the weight rotation (R) and lifted (L) rewards at the halfway point of the MC run such that it is indicated as [R+L|L]. Consider Curricula 3 and 5 which rewarded both lifting and rotation initially, and include the goal of lifting the ball within the desired height range during the second half of the MC runs. They perform as expected with Curriculum 5 being the better lifter as it only rewards lifting (but rotates, albeit slower, nonetheless; Fig. 4.11). The mean number of data points inside the target range for Curriculum 5 is 42, while only 6 numbers of completed rounds of rotations for all tactile information.

Curriculum 5 shows, on average, that lift success naturally increases upon switching from a combined reward to one that only rewards lifting (Fig. 4.13).

We used two visualizations to understand and gain insight from all curricula and sensory information. The first is based on the joint distribution of the mean ball's height (mm) vs. the number of completed full rotations at the final episode over 60 Monte Carlo runs, of Figures 4.11. The spectrum of colors in Figures 4.5 shows the measurement of rewards for different runs in all levels of sensory information, where the highest reward is in a dark red hue. The bar plots show the distribution of the runs, and the height distribution is binned in three categories: below, above, and within the target range.

The second visualization is the boxplots of total aggregated reward and the success rate as a percentage of time the ball's height spent in the target range in each curriculum, of Figures 4.12 and 4.13.

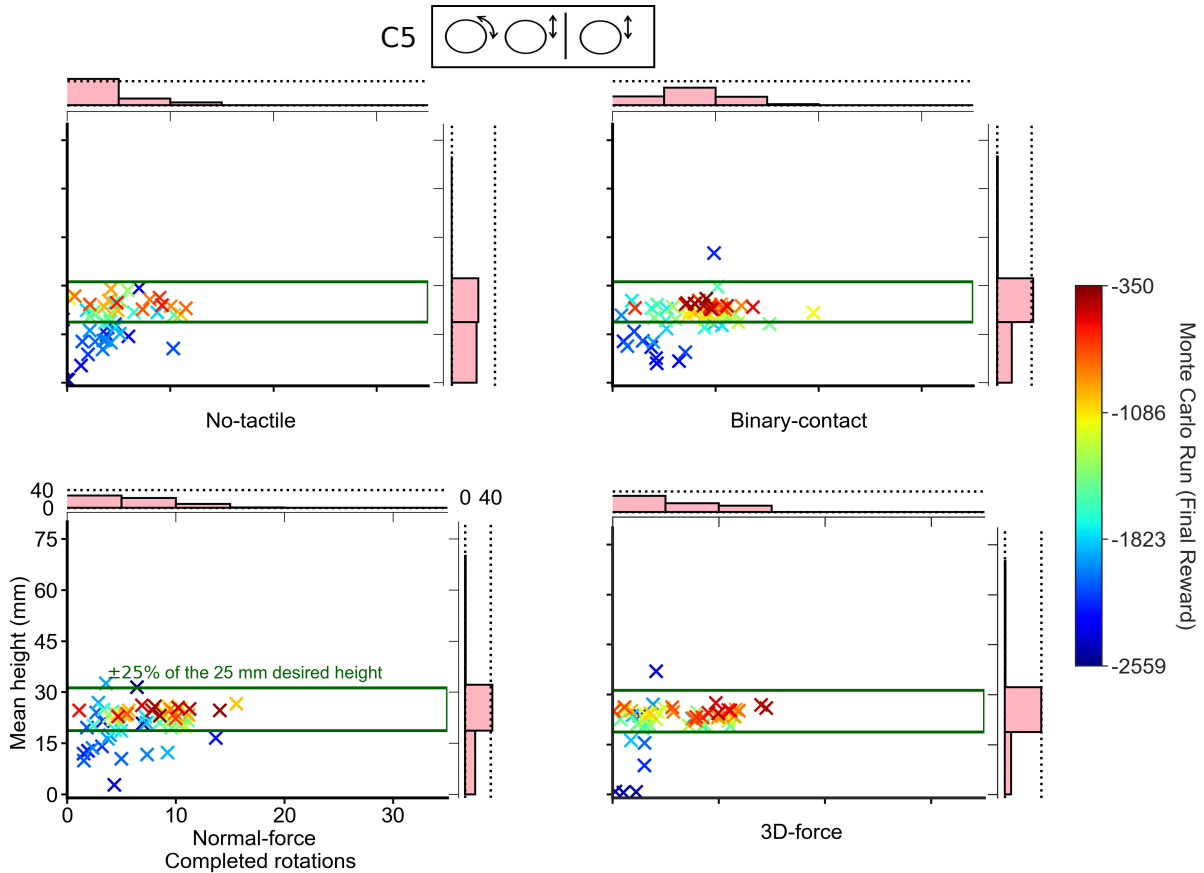


Figure 4.11: Performance for Curriculum 5 [R+L|L] for all options of tactile information. Note that the lack of reward for rotation in the second half of the 60 MC Runs now allows the hand to focus on placing the ball within the desired height range. Surprisingly, however, the robotic hand continues to rotate the ball even if not rewarded for it—likely because it has ‘learned’ rotation and lift in a coupled way. However, note that lift performance and learning rates are largely equivalent across tactile information options—with 3D-force being highest

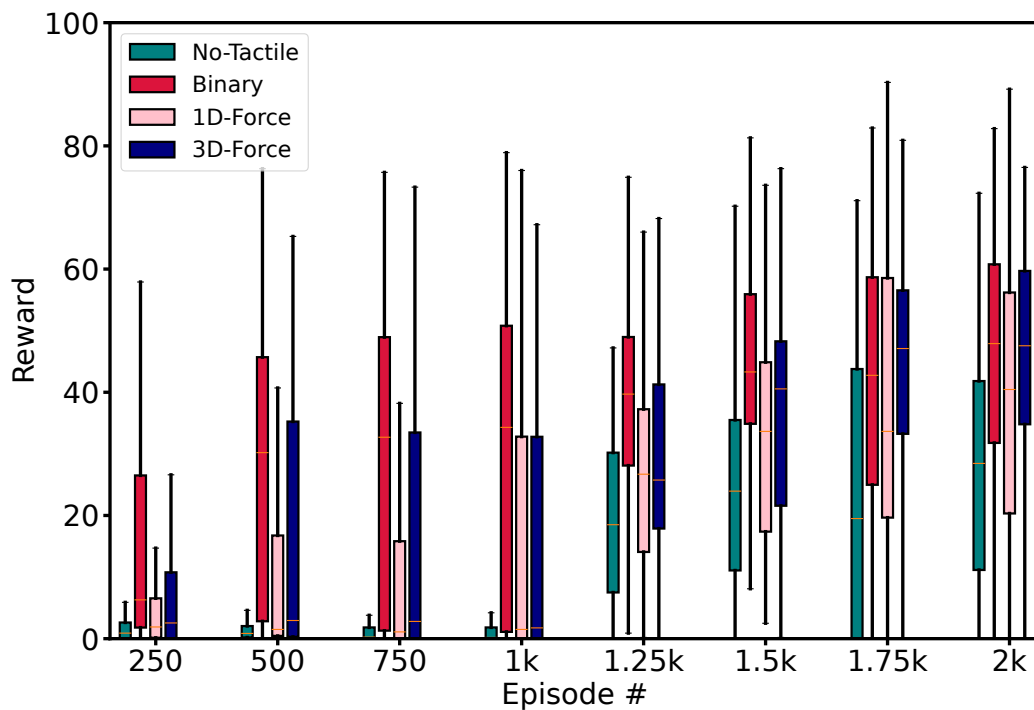


Figure 4.12: Performance for Curriculum 5 [R+L|L] for all options of tactile information. Lift success rate (the percentage of time the ball spent within 25% of the desired height over the duration of an episode). Boxplots, with median, across tactile information options for 60 MC runs at eight representative episodes, 250 episodes apart.

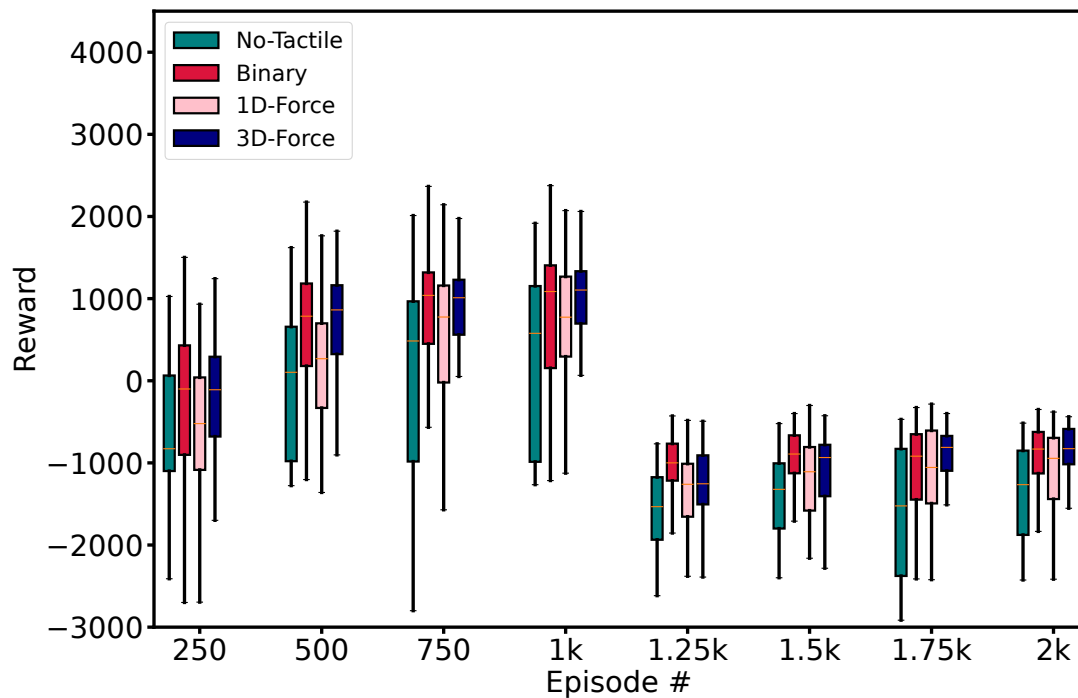


Figure 4.13: Performance for Curriculum 5[R+L|L] for all options of tactile information. Cumulative reward for each representative episode. Boxplots of the aggregated reward for tactile information while the boxes' orange lines show the median values, the edges of the boxes are the 25th and 75th percentiles. The reward gained per 2,000 episodes is visualized as boxplots grouped by the each tactile information aggregated over 60 Monte Carlo runs per box.

4.3 Curriculum drives manipulation performance

We trained our algorithm in five curricula and evaluated the performance on all curricula with cumulative results across all sensory modalities. Results are shown in Figure 4.14, which depicts a Pareto graph for the mean numbers of data points inside the target height range against the mean number of completed rounds of rotation for the data points inside the target height range. The Pareto graph in Figure 4.14 enables one to evaluate the performance and the effects of each curriculum on the overall success of our agent. This figure shows that curriculum-learning makes a significant difference in manipulation performance.

Our main finding, which runs counter to intuition, is that using rotation reward of the ball from the start is necessary to learn manipulation (both lifting and rotation). In contrast, a case that starts by only rewarding lifting, as in Curriculum 1, will not learn to manipulate the ball autonomously. The finding that manipulation is not learned in Curriculum 1 demonstrates that while lifting the ball seems like a ‘simpler’ task than rotating the ball; it does not appear to be possible before the system ‘understands’ the dynamics of manipulation. Our findings from Curriculum 2 suggest that our agent must first play with the object to understand the dynamics of the world around it, similar to what we do when we are toddlers [4]—as such, rotating the ball is a significant first step to a successful lifting.

As mentioned above, the first curriculum demonstrates the total reward function to be cumulative of lifting reward and rotating reward for the whole 2,000 episodes over four types of sensory information. While any height over 50 mm is when the ball has been locked in the hand, we may still see some manipulation reward (rotating the ball) through hand form closure [8, 63].

Curriculum 3 and Curriculum 5 had a final performance that could be useful for autonomous manipulation. This is opposed to for example Curriculum 2, in which ‘lifting’

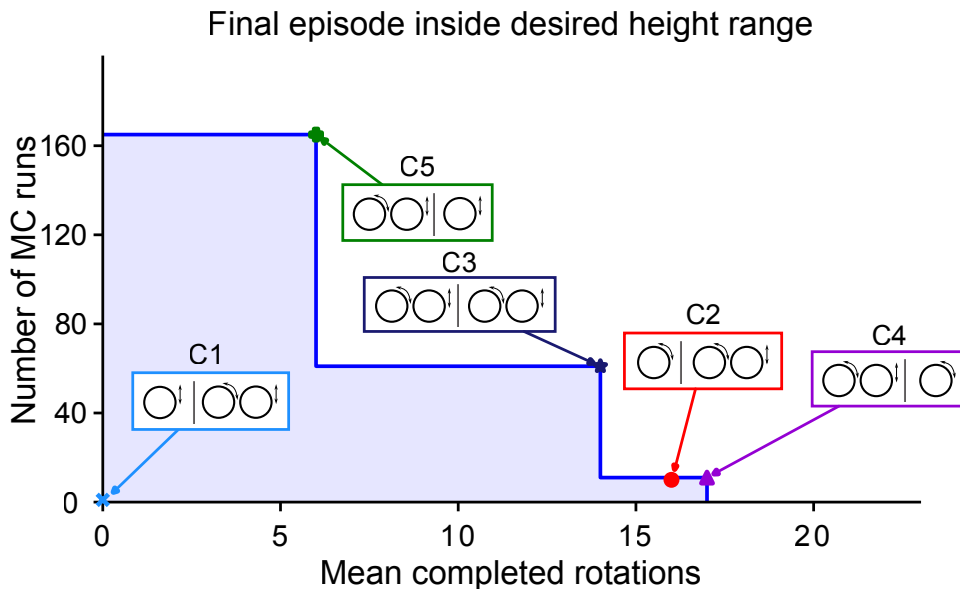


Figure 4.14: Pareto plots for the final performance from each curriculum highlight the divergence in manipulation performance across learning strategies. Final performance for all curricula across all tactile information options.

is only added in the second phase of learning, and so the agent cannot learn and perform the lifting task to its full extent compared to Curricula 3 and 5. However, it does learn to lift the ball to a degree, which is in contrast to the agent neither learning to lift nor rotate the ball in Curriculum 1 (only lifting rewarded first). Note these curricula were executed as independent MC runs for each of the four sensing options for tactile information from the pad of each finger (Fig. 3.3).

Our RL algorithm does not encourage the agent to start with a task-specific exploration since our agent learned autonomously without human intervention. In essence, continual learning performs incrementally where the new tasks leverage the knowledge learned in the previous tasks. Based on Curriculum 1 and Curriculum 2 in Figure 4.14, learning a ‘new’ task (lifting/or rotation) only for the second 1K episodes will not allow the agent to succeed. Curricula 3-5 enable the agent to learn and explore all the venues of the action space and exploit the knowledge at any given time step for the first half of the designated time (first 1,000 episodes), but only Curriculum 3 and Curriculum 5 had a

final performance that could be useful for autonomous manipulation. This is opposed to for example Curriculum 2, in which ‘lifting’ is only added in the second phase of learning, and so the agent cannot learn and perform the lifting task to its full extent compared to Curricula 3 and 5. However, it does learn to lift the ball to a degree, which is in contrast to the agent neither learning to lift nor rotate the ball in Curriculum 1 (only lifting rewarded first). Note that the desired manipulation performance is represented by those points inside the green box defining the desired ball height. These are plotted in Figs. 4.14.

4.4 The second phase of learning dictates end performance

The role of the reward function is fundamental in the ability to learn in-hand manipulation (lifting and rotating) against gravity as it defines the goal our agent strives to achieve during training. Based on the reward function in the second phase of learning, the performance of each curriculum at the end of learning changes. In each curriculum, the algorithm learns by being rewarded either for lifting the ball, rotating the ball, or both in each phase. In all Curricula except Curriculum 3, the reward function evolves and changes over the learning period (phase 1 and phase 2). In Curriculum 3, lifting and rotating are rewarded over the entire learning period (2,000 episodes). We demonstrate how changing the reward function during two phases in each curriculum affects the agent’s resulting manipulation performance (Figure 4.14). Manipulation performance at the end of each curriculum specifies a location on a Pareto plot (Fig. 4.14).

Considering the goal of lifting the ball within the target height range in the second phase, Curricula 3 and 5 (Figures 4.5 and 4.11) are far better lifters than Curricula 2 and 4. An interesting consequence of curriculum learning is that without changing the

reward halfway through, it might not be possible to obtain the performance in Curriculum 5 even with different reward parameters as demonstrated by essentially no learning seen in Curriculum 1. This is of consequence as Curriculum 5 has the best performance in terms of lifting the ball and keeping it at a specific height. Interestingly, it also rotates the ball slower than Curriculum 3 in achieving this lifting performance. In contrast, in Curricula 2 and 4, the mean number of data points inside the target range is 3 and 4, respectively, while the complete round of rotation is 16 and 17, respectively.

When only lift *or* rotation are rewarded in the first half of the MC run even though both features of manipulation are rewarded in the second half of the MC run (as in Curricula 1 and 2), manipulation is not achieved. In contrast, rewarding rotation *and* lift from the start enables manipulation, regardless of what is rewarded in the second half of the MC run as demonstrated in Curricula 3-5.

4.5 Rotating the ball is critical in learning how to lift the ball

Our main finding, which runs counter to intuition, is using a curriculum that rewards rotation of the ball from the start is necessary to learn manipulation (both lifting and rotation). In contrast, a curriculum that starts by only rewarding lifting, as in Curriculum 1, will not learn to autonomously manipulate the ball, and will actually not learn to lift or rotate the ball. The finding that manipulation is not learned in Curriculum 1 in Figure 4.14 demonstrates that while lifting the ball seems like a ‘simpler’ task than rotating the ball, it does not appear to be possible before the system ‘understands’ the dynamics of manipulation. Our findings from Curriculum 2 suggest that our agent must first play with the object to understand the dynamics of the world around it, similar to what we do when we are toddlers [4]. As such, rotating the ball is the significant first step to

a successful lifting. As shown in Figure 4.14, Curricula 2-5 by rewarding rotation from the beginning (and therefore encouraging) the exploration of the full dynamics of the environment, the agent is enabled to learn the combined manipulation task of rotating and lifting the ball (cf. Figs. 4.14 and 4.11).

Chapter 5

Effect of sensory information

5.1 Chapter summary

Using the RL algorithm, we demonstrate autonomous learning of autonomous dexterous manipulation in simulation with gravity. This section discusses the results of available tactile information (i.e., force) at the fingertips for in-hand manipulation. We evaluate four types of tactile sensing included in the state variable used in the PPO algorithm to determine the effects of sensory information on learning manipulation. This section studies the effect of rotating the ball on learning how to lift the ball. We demonstrate how the right choice of curriculum can enable achieving the desired manipulation performance in previous chapters.

5.2 In-hand manipulation performance

As it is mentioned in the previous chapter, we choose the dynamical manipulation task of lifting a ball and rotating it along a horizontal rotation axis at a target height. Our algorithm would learn by being rewarded for both lifting and rotating. A learning trial consisted of 2,000 episodes, split into two equal learning phases. This task is performed

with four types of tactile information at its soft fingertips. To evaluate the effectiveness of different tactile sensory information, we divide our work into five different curricula. The details of the curricula are shown in Table 3.2.

To investigate the performance of our autonomous learning in-hand manipulation using different tactile sensory information, the joint distribution of the mean ball’s height (mm) vs. the number of completed rotations in each Monte Carlo run for the last episode are shown in (e.g. Figure 4.5 and Figure 4.11) . Each joint distribution graph consists of data for 60 independent Monte Carlo runs. Each Monte Carlo run is color-coded based on the range of rewards from lowest to highest reward among all tactile conditions. The spectrum of colors in the figures shows the measurement of rewards for different Monte Carlo runs, where the highest reward is shown in dark red. As it mentioned earlier, the goal is reached when the agent rotates the ball while keeping it against gravity between height $[18.75, 31.25]mm$ which is the duty cycle of $\pm\%25$ of that target height (25 (mm) from the ground). This duty cycle of $\pm\%25$ of that target height indicates the green box in Figure 4.5. The results discussed in further details in next sections for each case.

5.3 Effect of sensory information

We evaluate our algorithm performance on all curricula in the presence four different sensory modalities. Our agent has one of four levels of tactile information available at its fingertips: No-Tactile (i.e., null), Contact-force (detecting contact), 1D-force (normal force) and 3D-force vector (normal plus tangential forces). The Pareto front in Figure 5.1 enables one to evaluate the difference in the effects of each tactile condition on manipulation performance.

Pareto plots for the final performance from each curriculum highlight the divergence in manipulation performance across tactile information are shown in Figure 5.1. Mean final performance for all successful curricula and tactile information options. The tactile

information available to the PPO policy (Fig. 3.3) affects the robotic hand’s ability to learn manipulation. Surprisingly, learning happened even in the absence of tactile information (No-tactile) and manipulation performance was not always best with 3D-force information. Note these Pareto plots only consider those final episodes when the ball was on average lifted to within 25% of the desired height. For complete final performance outside of the desired height range, see Figs. 4.5 and 4.11 and Supplementary Figs. 4.2 and 4.8, which show similar trends.

Based on the results in Figure 5.1, Curriculum 5 shows that 3D-force learned to lift best (50 counts inside the target height range); however, the 1D-force and Contact-force are similar (with 42 counts). In contrast, No-tactile has the worst performance compared with any other condition with tactile information (32 counts). Comparing the results in Curriculum 3 shows that sensory information is beneficial, but interestingly tactile sensing with Contact-force information had higher performance in terms of both rotating and lifting the ball compared with 1D-force and 3D-force (the mean number of data points inside the target range is 25, while it has 14 numbers of completed rounds of rotations). Our results in all curricula show that tactile information makes a difference, although for Curricula 2 and 4 the number of runs that learned to have a final performance within the mean height range was small.

The results show that 3D-force tactile information at the fingertips was useful—but not strictly necessary or always led to the best performance (Figs. 5.1, 4.5, and 4.11). From Figure 5.1 we observe that having any tactile information (Contact-force, 1D-force, 3D-force) always leads to better performance than the case of No-tactile information. Still, it is interesting that the agent can learn to manipulate without any tactile information (e.g., C5).

It was surprising to us that the different options of tactile information proved to be of secondary importance compared to curriculum. This idea was reinforced by the seminal

work by Johansson and Westling [64, 65] demonstrating that numbing the fingerpads with temporary anaesthetic greatly impairs fine control of grasp, as evidenced by their once famous video showing how an adult loses the ability to light a match. Moreover, active state estimation is a tenet of linear feedback control to optimize control actions. Our results in Fig. 5.1 fly in the face of these longstanding notions about the critical importance of tactile information for manipulation.

The resolution of this paradox comes from other lessons in biological and *nonlinear* control in the form of switching, wait-and-act, or hybrid control [66] where the system alternates between open- and closed-loop control. An engineering example is how adding a ‘deadband’ around the target temperature in a thermostat eliminates chatter. A biological example is how humans perform stick-balancing where we wait for the stick to fall over a certain amount before making a corrective movement [67]. This alternative control approach, then, explains why continual feedback is neither necessary nor, at times, better. Rather, intermittent Binary-contact does well for our nonlinear dynamic manipulation case with intermittent contact between the fingertips and ball. Moreover, Normal-force and 3D-force are not always better, and even the open-loop No-tactile control can work, albeit worse than the others. This important result that provides evidence to revise the role of tactile information will allow freer thinking for engineers (and bioroboticists) creating the next generation of dexterous hands.

As it is shown in Fig. 4.14) and No-tactile being lowest (but which still learns to lift!). However, note that lift performance and learning rates are largely equivalent across tactile information options—with 3D-force being highest (detail in Fig. 4.14) and No-tactile being lowest (but which still learns to lift!)

5.4 Learning Trends

Curriculum 5 shows, on average, that lift success naturally increases upon switching from a combined reward to one that only rewards lifting (Fig. 4.13). But average lift success does not tell the whole story. We visually examined how individual trials within Curriculum 5 for 3D-force responded to the switch in reward, and find that the MC runs fall into distinct learning trends (see Fig. 5.2). Most surprisingly, the majority (53.33%) of individual MC runs show mostly *no* success in lifting the ball before the switch, even though Curriculum 5 rewards it from the start. But, after the switch in reward, the success in lifting the ball improves quickly and dramatically (see Fig. ??). This learning trend, which we call ‘Covert Learning’, highlights that learning is happening but is not overtly demonstrated (even if rewarded). The other learning trends are also interesting in their own right as they show the expected increase in lift performance from the start.

We reviewed the learning trends in detail we find interesting trends. We see that certain MC runs within a curriculum can exhibit ‘covert learning’ that is revealed only after the reward function changes (Fig. 5.2).

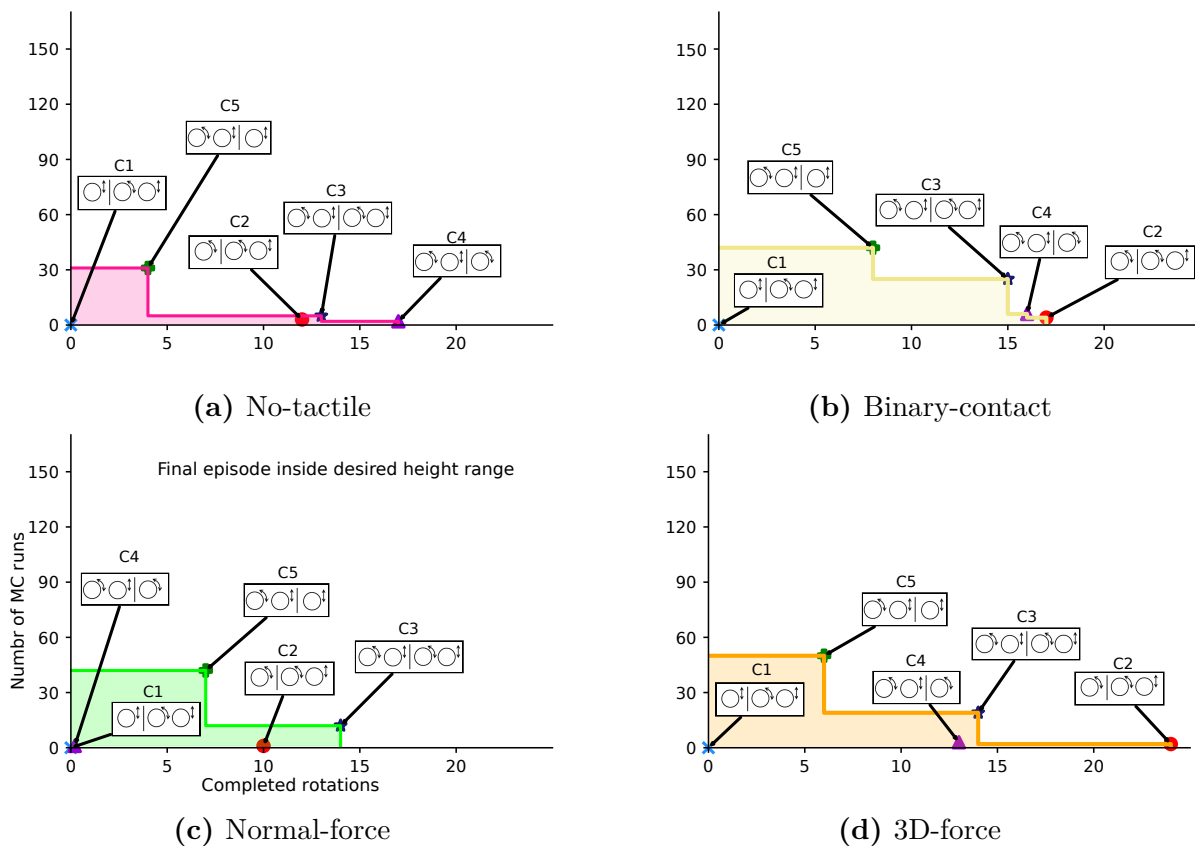
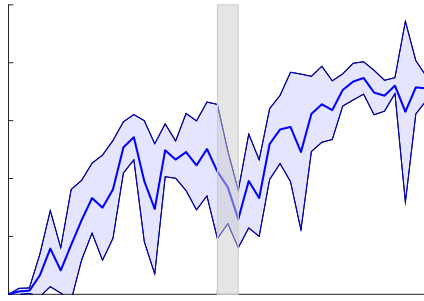
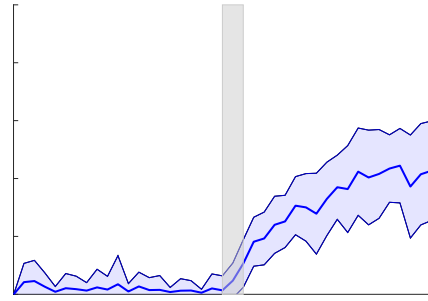


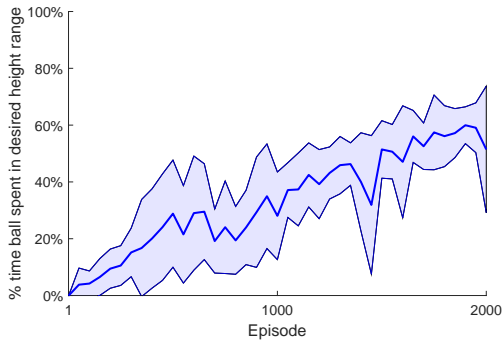
Figure 5.1: Pareto plots for the final performance across curricula and the four tactile information options. Each Pareto plot shows the mean final performance for all curricula and corresponding tactile information available to the learning policy: (a) No-tactile, (b) Binary-contact, (c) Normal-force, and (d) 3D-force. While curriculum drives learning to distinct regions, the tactile information available to the PPO policy (Fig. 3.3) also affects the robotic hand’s ability to learn manipulation. Interestingly, we see that learning happened even in the absence of tactile information (a), and that manipulation performance was not always best with 3D-force information (b-d). Note these Pareto plots only consider those final episodes when the ball was on average lifted to within 25% of the desired height.



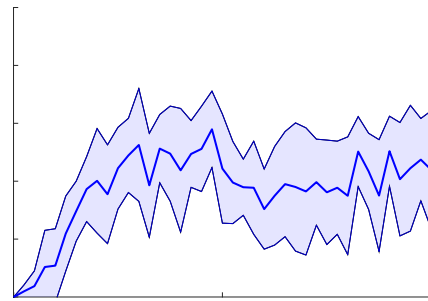
(a) Dip and Improve After Switch



(b) Improve After Switch



(c) Continuous Improvement



(d) Plateau Early

Figure 5.2: Types of ‘learners’ for Curriculum 5 with 3D-Force sensing. Out of 60 Monte Carlo runs, four distinct types of learners were visually identified: those that after the change in reward from a combination to only lifting, **a** had their performance decrease before going on to exceed the performance at the end of 1K trials (10% of trials), **b** experienced a sudden increase in performance at the switch (53.33% of trials), **c** continuously improved their lifting performance (13.33% of trials), or **d** plateaued in their learning well within the first phase (18.33% of trials). Note that 5% of runs experienced no learning. The shaded region in the ‘Dip and Improve After Switch’ and ‘Improve After Switch’ highlight the change in performance when the reward changes after 1K episodes.

Chapter 6

Conclusion

The main contributions of this work are: i) we show the importance of curriculum learning and active sensing/exploration in learning complicated tasks of manipulation (rotation and lifting against gravity); we extend our work by exploring the role of the reward function used on the ability to learn a combined in-hand manipulation task; ii) we show the significant contributions of different levels of tactile sensory information in the combined manipulation tasks.

6.1 Learning

The data intensive nature of reinforcement learning necessitates use of simulated environments for training model[48]. Here we show how a model-free, open-loop approach allows autonomous learning to produce effective movements in in-hand manipulation.

Many model-free approaches use RL algorithms where control parameters are tuned based on a reward function and extensive interactions with the environment [68, 69]. Furthermore, this proposed methodology does not require analytical dynamic or kinematic models and the learned skills generalize to novel objects with a slight loss in performance [27, 10]. As mechanical systems begin to approximate the human-hand more faithfully

[70, 71], reinforcement learning will be necessary for these devices to be useful. This further follows the fact that precise prior knowledge of the system and the environment is not usually available for dynamical tasks in the physical world [48, 72, 73]. Our autonomous learning uses RL algorithms which does not use a prior model. It uses the end-to-end proximal policy optimization (PPO), where the control parameters are tuned based on our reward functions during each phase of the learning period. In fact, PPO has been identified as one of the most robust approaches against reward perturbances [56].

To enable the agent to operate in a real-world scenario, we tackled the manipulation problem with the hand facing downwards without external support. In this setup, we demonstrate the effect of different curricula on autonomous learning.

Comparing all curricula, we see how the curriculum drives manipulation performance. As Curriculum 1 and 2 suggests, learning a 'new' task only for 1,000 episodes (the second 1,000 episodes) will not allow the agent to succeed. In contrast, if we enable the agent to explore all the avenues to achieve reward, over the 2,000 episodes, we can make the hand do one task perfectly on its known. This result is shown in Curricula 3-5, where we allow the agent to learn and explore all the avenues and explore the action space and exploit the knowledge at any given time step for the first half of the designated time. Looking at the rewards of the last episode of these three curricula through Figure 4.8 and 4.11 we see agent was able to achieve the highest rewards based on the designated reward function for the second 1,000 episodes).

6.2 Sensory Modalities

Besides sensor systems that help the robots to structure their environment, like cameras, radar sensors, etc., a sensor system on the robot's surface that can detect mechanical contacts of the robot with its environment is needed [74, 75]. Contact information from

tactile sensors attached to the fingertips has been utilized for manipulation recognition [76, 77]. For the tactile sensors applied on robots, most of the sensors are designed for the fingertip, which measures the force or contact during manipulation task [77, 78].

We evaluate the dynamical manipulation task of lifting a ball and rotating it along a horizontal rotation axis at a target height to check the effectiveness of different tactile sensory information in in-hand manipulation. Although our results in all curricula show that the dimensionality of tactile information is a determinant of success, it depends on our goal to show which sensory condition is better than the others. For the same number of training epochs across sensory conditions, the 3D force-sensing enabled significant improvement in learning rate and final performance compared to the null condition and other sensory information on Curriculum 5. However, we did not find a further improvement in learning rates or performance increase for this sensory condition on Curriculum 3.

6.3 Active sensing

Mastery in a manipulation task is often achieved by practice, which is performing a task repeatedly with a specific goal in mind. Task-specific random exploration (active sensing) enables an agent to learn the specific tasks that it would not be able to learn otherwise [79]. Active sensing in robotics incorporates how to make decisions for the next actions to maximize information gain and minimize costs [80]. Here, we have explored this idea of mastering tasks by encouraging the agent to start with task-specific exploration that would lead to active sensing in the context of autonomous robotic manipulation.

Exploring the idea of mastering a task by encouraging the agent to start with task-specific exploration that would lead to active sensing is one of the fundamental decisions in choosing our reward. We show that providing a curriculum that from the start rewards rotating the ball, as opposed to lifting the ball, is necessary to learn manipulation. We

believe this is because rewarding rotation first encourages a form of *active sensing* [79] in which our agent can learn to lift the ball against gravity while rotating it at a target height. By rewarding (and therefore encouraging) exploration of the full dynamics of manipulation (i.e., the grasp matrix of the system [63]), the implicit model being built approximates full rank and is, therefore, more practical. This is also perhaps analogous to the observability and persistence of excitation in control theory [81].

In the context of Machine Learning, a human usually specifies a curriculum to be followed by the agent where prior knowledge provides a rank ordering of the functional components of the task based on assumed complexity. In our case, we did not specify such rank-ordering as the agent learned autonomously without human intervention. In retrospect, one would have thought lifting the ball by learning form closure is “simpler” than lifting and spinning the ball, which requires force closure [8, 81]. However, we find that exploring how to rotate the ball (a more “complex” task) seems to be, in fact, critical to learning how to lift the ball (a “simpler” task).

Looking at the success rate (the percentage of time being spent between the targeted height range of 18.75 and 31.25mm), we see the importance of learning to lift the ball lies under starting to explore the environment (action space); in our designed task this happens through the rotational task. This is being extracted by looking at all five curricula; comparing the lifting task of the first Curriculum and curricula 3 and 5, we see learning to lift is not applicable without any exploitation of the fingers.

6.4 Limitations, opportunities and future directions

While our work pushes the field of autonomous manipulation forward, it has some limitations. First, our work is done in simulation. But, as with many other studies looking to bridge the sim2real divide [48, 49], we used a realistic physics engine (i.e., MuJoCo) that enables future work to implement our approach in hardware. We also

made some kinematic simplifications by using a three-fingered robotic hand (as opposed to five), but three is the minimum number of fingers required for manipulation, and it is common for useful robotic hands to have fewer than five fingers [27, 71]. Admittedly, we added some stabilizing stiffness and damping to reduce the effective kinematic degrees of freedom to keep the ball from rolling away and facilitate learning (our hand does not move laterally). These constraints could likely be relaxed in future work at the expense of increased run time. However, these reduced kinematic degrees of freedom do not in themselves annul the proof-of-principle we present: a robotic hand that learned dynamic dexterous manipulation.

We did not study the generalizability of our results to other objects (e.g., tools), which leaves room for promising future work. Finally, our learning approach seems to implicitly couple the tasks of rotating and lifting the ball in a way that cannot be learned separately (rotation can be learned individually but lift cannot). While our approach is practical and useful for dynamic manipulation, it does not solve the problem of autonomously generating static grasps, which was not our focus. Finding a way to bridge autonomous learning for the combined abilities of grasp and manipulation could be an exciting adaptation of our approach to autonomous learning.

Bibliography

- [1] Francisco J Valero-Cuevas. Why the hand? *Progress in Motor Control*, pages 553–557, 2009.
- [2] Kathleen Gibson. Tools, language and intelligence: Evolutionary implications. *Man*, pages 255–264, 1991.
- [3] A Iriki and M Taoka. Triadic niche construction: A scenario of human brain evolution extrapolating tool-use and language from control of the reaching actions. *Philos. Trans. R. Soc. Lond. B*, 367:10–23, 2012.
- [4] Valerio Ortenzi, Marco Controzzi, Francesca Cini, Juxi Leitner, Matteo Bianchi, Maximo A Roa, and Peter Corke. Robotic manipulation and the role of the task in the metric of success. *Nature Machine Intelligence*, 1(8):340–346, 2019.
- [5] Cynthia Breazeal and Brian Scassellati. How to build robots that make friends and influence people. In *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No. 99CH36289)*, volume 2, pages 858–863. IEEE, 1999.
- [6] Roger N Lemon, RS Johansson, and G Westling. Corticospinal control during reach, grasp, and precision lift in man. *Journal of Neuroscience*, 15(9):6145–6156, 1995.

- [7] Christine L MacKenzie and Thea Iberall. *The grasping hand*. Elsevier, 1994.
- [8] Francisco J Valero-Cuevas and Marco Santello. On neuromechanical approaches for the study of biological and robotic grasp and manipulation. *Journal of neuroengineering and rehabilitation*, 14(1):1–20, 2017.
- [9] Aude Billard and Danica Kragic. Trends and challenges in robot manipulation. *Science*, 364(6446), 2019.
- [10] Minas V Liarokapis and Aaron M Dollar. Learning task-specific models for dexterous, in-hand manipulation with simple, adaptive robot hands. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2534–2541. IEEE, 2016.
- [11] 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.
- [12] Kendall Lowrey, Svetoslav Kolev, Jeremy Dao, Aravind Rajeswaran, and Emanuel Todorov. Reinforcement learning for non-prehensile manipulation: Transfer from simulation to physical system. In *2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, pages 35–42. IEEE, 2018.
- [13] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pages 1101–1112. PMLR, 2020.
- [14] Ilija Radosavovic, Xiaolong Wang, Lerrel Pinto, and Jitendra Malik. State-only imitation learning for dexterous manipulation. *arXiv preprint*, 2020.

- [15] Tingguang Li, Krishnan Srinivasan, Max Qing-Hu Meng, Wenzhen Yuan, and Jeanette Bohg. Learning hierarchical control for robust in-hand manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8855–8862. IEEE, 2020.
- [16] Ivaylo Popov, Nicolas Heess, Timothy Lillicrap, Roland Hafner, Gabriel Barth-Maron, Matej Vecerik, Thomas Lampe, Yuval Tassa, Tom Erez, and Martin Riedmiller. Data-efficient deep reinforcement learning for dexterous manipulation. *arXiv preprint arXiv:1704.03073*, 2017.
- [17] Ana-Maria Cretu, Pierre Payeur, and Emil M Petriu. Soft object deformation monitoring and learning for model-based robotic hand manipulation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(3):740–753, 2011.
- [18] Pietro Falco, Abdallah Attawia, Matteo Saveriano, and Dongheui Lee. On policy learning robust to irreversible events: An application to robotic in-hand manipulation. *IEEE Robotics and Automation Letters*, 3(3):1482–1489, 2018.
- [19] Igor Mordatch, Zoran Popović, and Emanuel Todorov. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, pages 137–144, 2012.
- [20] Yunfei Bai and C Karen Liu. Dexterous manipulation using both palm and fingers. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1560–1565. IEEE, 2014.
- [21] Vikash Kumar, Yuval Tassa, Tom Erez, and Emanuel Todorov. Real-time behaviour synthesis for dynamic hand-manipulation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6808–6815. IEEE, 2014.

- [22] Vikash Kumar, Emanuel Todorov, and Sergey Levine. Optimal control with learned local models: Application to dexterous manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 378–383. IEEE, 2016.
- [23] Shane Legg and Marcus Hutter. Universal intelligence: A definition of machine intelligence. *Minds and machines*, 17(4):391–444, 2007.
- [24] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *arXiv preprint arXiv:1906.08253*, 2019.
- [25] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.
- [26] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- [27] Herke Van Hoof, Tucker Hermans, Gerhard Neumann, and Jan Peters. Learning robot in-hand manipulation with tactile features. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 121–127. IEEE, 2015.
- [28] Henry Zhu, Abhishek Gupta, Aravind Rajeswaran, Sergey Levine, and Vikash Kumar. Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3651–3657. IEEE, 2019.
- [29] Deirdre Quillen, Eric Jang, Ofir Nachum, Chelsea Finn, Julian Ibarz, and Sergey Levine. Deep reinforcement learning for vision-based robotic grasping: A simulated

- comparative evaluation of off-policy methods. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6284–6291. IEEE, 2018.
- [30] Shirin Joshi, Sulabh Kumra, and Ferat Sahin. Robotic grasping using deep reinforcement learning. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1461–1466. IEEE, 2020.
- [31] Andrew Melnik, Luca Lach, Matthias Plappert, Timo Korthals, Robert Haschke, and Helge Ritter. Tactile sensing and deep reinforcement learning for in-hand manipulation tasks. In *IROS Workshop on Autonomous Object Manipulation*, 2019.
- [32] Sunny Katyara, Fanny Ficuciello, Darwin Caldwell, Bruno Siciliano, and Fei Chen. Leveraging kernelized synergies on shared subspace for precision grasp and dexterous manipulation. *arXiv preprint arXiv:2008.11574*, 2020.
- [33] Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. *arXiv preprint arXiv:2111.03043*, 2021.
- [34] Tatsuya Ooka and Kinya Fujita. Virtual object manipulation system with substitutive display of tangential force and slip by control of vibrotactile phantom sensation. In *2010 IEEE Haptics Symposium*, pages 215–218. IEEE, 2010.
- [35] Jonna Laaksonen, Ekaterina Nikandrova, and Ville Kyrki. Probabilistic sensor-based grasping. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2019–2026. IEEE, 2012.
- [36] Hao Dang and Peter K Allen. Stable grasping under pose uncertainty using tactile feedback. *Autonomous Robots*, 36(4):309–330, 2014.
- [37] Hong Zhang and Ning Nicholas Chen. Control of contact via tactile sensing. *IEEE Transactions on Robotics and Automation*, 16(5):482–495, 2000.

- [38] Kaijen Hsiao, Sachin Chitta, Matei Ciocarlie, and E Gil Jones. Contact-reactive grasping of objects with partial shape information. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1228–1235. IEEE, 2010.
- [39] Pierre Payeur, Codrin Pasca, A-M Cretu, and Emil M Petriu. Intelligent haptic sensor system for robotic manipulation. *IEEE Transactions on Instrumentation and Measurement*, 54(4):1583–1592, 2005.
- [40] Yevgen Chebotar, Oliver Kroemer, and Jan Peters. Learning robot tactile sensing for object manipulation. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3368–3375. IEEE, 2014.
- [41] Miao Li, Yasemin Bekiroglu, Danica Kragic, and Aude Billard. Learning of grasp adaptation through experience and tactile sensing. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3339–3346. Ieee, 2014.
- [42] Taro Takahashi, Toshimitsu Tsuboi, Takeo Kishida, Yasunori Kawanami, Satoru Shimizu, Masatsugu Iribe, Tetsuharu Fukushima, and Masahiro Fujita. Adaptive grasping by multi fingered hand with tactile sensor based on robust force and position control. In *2008 IEEE International Conference on Robotics and Automation*, pages 264–271. IEEE, 2008.
- [43] Joseph M Romano, Kaijen Hsiao, Günter Niemeyer, Sachin Chitta, and Katherine J Kuchenbecker. Human-inspired robotic grasp control with tactile sensing. *IEEE Transactions on Robotics*, 27(6):1067–1079, 2011.
- [44] Leif P Jentoft, Qian Wan, and Robert D Howe. Limits to compliance and the role of tactile sensing in grasping. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6394–6399. IEEE, 2014.

- [45] Roland S Johansson and G Westling. Signals in tactile afferents from the fingers eliciting adaptive motor responses during precision grip. *Experimental brain research*, 66(1):141–154, 1987.
- [46] Charlotte Häger-Ross, Kelly J Cole, and Roland S Johansson. Grip-force responses to unanticipated object loading: load direction reveals body-and gravity-referenced intrinsic task variables. *Experimental Brain Research*, 110(1):142–150, 1996.
- [47] Zeynep Celik-Butler, Ravinder S Dahiya, Manuel Quevedo-Lopez, Yong Xu, and Sigurd Wagner. Guest editorial: Special issue on flexible sensors and sensing systems. *IEEE Sensors Journal*, 13(10):3854–3856, 2013.
- [48] Ali Marjaninejad, Darío Urbina-Meléndez, Brian A Cohn, and Francisco J Valero-Cuevas. Autonomous functional movements in a tendon-driven limb via limited experience. *Nature machine intelligence*, 1(3):144–154, 2019.
- [49] Felix Ruppert and Alexander Badri-Spröwitz. Learning plastic matching of robot dynamics in closed-loop central pattern generators. *Nature Machine Intelligence*, pages 1–9, 2022.
- [50] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [51] Vikash Kumar, Abhishek Gupta, Emanuel Todorov, and Sergey Levine. Learning dexterous manipulation policies from experience and imitation. *arXiv preprint arXiv:1611.05095*, 2016.
- [52] 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.

- [53] Kyle Mills, Pooya Ronagh, and Isaac Tamblyn. Finding the ground state of spin hamiltonians with reinforcement learning. *Nature Machine Intelligence*, 2(9):509–517, 2020.
- [54] Jeppe Theiss Kristensen and Paolo Burelli. Strategies for using proximal policy optimization in mobile puzzle games. In *International conference on the foundations of digital games*, pages 1–10, 2020.
- [55] Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. <https://github.com/openai/baselines>, 2017.
- [56] Jingkang Wang, Yang Liu, and Bo Li. Reinforcement learning with perturbed rewards. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6202–6209, 2020.
- [57] Cheng-Yen Tang, Chien-Hung Liu, Woei-Kae Chen, and Shingchern D You. Implementing action mask in proximal policy optimization (ppo) algorithm. *ICT Express*, 6(3):200–203, 2020.
- [58] Perttu Hämmäläinen, Amin Babadi, Xiaoxiao Ma, and Jaakko Lehtinen. Ppo-cma: Proximal policy optimization with covariance matrix adaptation. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2020.
- [59] Gang Chen, Yiming Peng, and Mengjie Zhang. An adaptive clipping approach for proximal policy optimization. *arXiv preprint arXiv:1804.06461*, 2018.
- [60] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

- [61] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [62] Jeffrey L Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99, 1993.
- [63] Richard M Murray, Zexiang Li, and S Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 2017.
- [64] Roland S Johansson and Goran Westling. Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects. *Experimental Brain Research*, 56(3):550–564, 1984.
- [65] Roland S Johansson, Charlotte Häger, and Ronald Riso. Somatosensory control of precision grip during unpredictable pulling loads. *Experimental Brain Research*, 89(1):192–203, 1992.
- [66] John Guckenheimer. A robust hybrid stabilization strategy for equilibria. *IEEE Transactions on Automatic Control*, 40(2):321–326, 1995.
- [67] Juan Luis Cabrera and John G Milton. Human stick balancing: tuning lévy flights to improve balance control. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 14(3):691–698, 2004.
- [68] Mathias Thor and Poramate Manoonpong. Versatile modular neural locomotion control with fast learning. *Nature Machine Intelligence*, 4(2):169–179, 2022.
- [69] Jeff Clune, Kenneth O Stanley, Robert T Pennock, and Charles Ofria. On the performance of indirect encoding across the continuum of regularity. *IEEE Transactions on Evolutionary Computation*, 15(3):346–367, 2011.

- [70] Ashish D Deshpande, Zhe Xu, Michael J Vande Weghe, Benjamin H Brown, Jonathan Ko, Lillian Y Chang, David D Wilkinson, Sean M Bidic, and Yoky Matsuoka. Mechanisms of the anatomically correct testbed hand. *IEEE/ASME Transactions on Mechatronics*, 18(1):238–250, 2011.
- [71] Andrew Morgan, Kaiyu Hang, Bowen Wen, Kostas E Bekris, and Aaron Dollar. Complex in-hand manipulation via compliance-enabled finger gaiting and multi-modal planning. *IEEE Robotics and Automation Letters*, 2022.
- [72] Josh Bongard, Victor Zykov, and Hod Lipson. Resilient machines through continuous self-modeling. *Science*, 314(5802):1118–1121, 2006.
- [73] Duy Nguyen-Tuong, Jan Peters, Matthias Seeger, and Bernhard Schölkopf. Learning inverse dynamics: a comparison. In *European symposium on artificial neural networks*, number CONF, 2008.
- [74] Oliver Kerpa, Karsten Weiss, and Heinz Worn. Development of a flexible tactile sensor system for a humanoid robot. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 1, pages 1–6. IEEE, 2003.
- [75] Di Guo, Fuchun Sun, Bin Fang, Chao Yang, and Ning Xi. Robotic grasping using visual and tactile sensing. *Information Sciences*, 417:274–286, 2017.
- [76] Darío Urbina-Meléndez, Jiaoran Wang, Daniel Wang, Ali Marjaninejad, and Francisco J Valero-Cuevas. Estimating center of pressure of a bipedal mechanism using a proprioceptive artificial skin around its ankles. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 4522–4528. IEEE, 2021.

- [77] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.
- [78] Nawid Jamali, Marco Maggiali, Francesco Giovannini, Giorgio Metta, and Lorenzo Natale. A new design of a fingertip for the iCub hand. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2705–2710. IEEE, 2015.
- [79] Kevin M Lynch, Hitoshi Maekawa, and Kazuo Tanie. Manipulation and active sensing by pushing using tactile feedback. In *IROS*, volume 1, 1992.
- [80] Lyudmila Mihaylova, Tine Lefebvre, Herman Bruyninckx, Klaas Gadeyne, and Joris De Schutter. A comparison of decision making criteria and optimization methods for active robotic sensing. In *International Conference on Numerical Methods and Applications*, pages 316–324. Springer, 2002.
- [81] Michael Green and John B Moore. Persistence of excitation in linear systems. *Systems & control letters*, 7(5):351–360, 1986.