Physical Gameplay in Robotics: Advancing Robotic Skills Through Game-Based Challenges

Boling Yang

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

Doctor of Philosophy

University of Washington

2023

Reading Committee:
Joshua R. Smith, Chair
Byron Emereth Boots, Chair
Abhishek Gupta

Program Authorized to Offer Degree:
Paul G. Allen School of Computer Science and Engineering

University of Washington

Abstract

Physical Gameplay in Robotics:
Advancing Robotic Skills Through Game-Based Challenges

Boling Yang

Co-Chairs of the Supervisory Committee:
Joshua R. Smith
Paul G. Allen School of Computer Science and Engineering and Department of Electrical and Computer Engineering

Byron Emereth Boots
Paul G. Allen School of Computer Science and Engineering

We explore the intersection of robotics and physical games, introducing innovative methods to enhance robot capabilities in real-world scenarios. Recognizing games as potent tools for refining cognitive and sensory-motor skills, we leverage diverse perspectives, including benchmarking, human-robot interaction, and robot learning. Through these lenses, we demonstrate that physical games provide an optimal environment for robots to hone their problem-solving and manipulation capabilities. Our research delves into how puzzles, exemplified by the Rubik's Cube, can bolster robots' sensing and manipulation skills. Further, we delve into the development of human-robot interactions in competitive exercises, showcasing the potential benefits of a robot as a contender. We also present a game-theoretic automatic curriculum learning algorithm, aiming to enhance the learning efficiency of robots in competitive gaming contexts. Finally, we advocate for the application of game-based methods to real-world robotic tasks, particularly object handling and rearranging within warehouse storage units.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to my advisors, Josh Smith and Byron Boots, for their exceptional support and guidance throughout my Ph.D. journey.

Josh, your remarkable care and support have been invaluable. When I was just starting out in my research career, with more passion than skills in robotics, I am deeply thankful for your trust in me, and for providing the resources and opportunities to pursue the risky ideas I proposed. I also greatly appreciate your efforts in establishing and nurturing our lab's culture, which offers ample academic freedom. This environment has been instrumental in allowing each lab member, including myself, to develop as an independent researcher. You are my role model for what a visionary should be.

Byron, I am grateful for the hands-on advising, encouragement, and insistence on regular stand-up updates. These have been incredibly motivating, driving my research progress, particularly during times when intricate challenges tempted me to feel lazy. Your discussions have always been inspiring, both within and outside academia. You are my role model for what a great leader should be.

I would also like to express my gratitude to my other committee members, Abhishek Gupta, Sawyer Fuller, and Vikram Iyer, for their valuable feedback. Your insights were crucial in shaping the scope of my Ph.D. dissertation. Special thanks to Abhishek for dedicating many hours to project discussions, refining research ideas, and enhancing my understanding of reinforcement learning.

I feel privileged to have been a part of the Sensor Systems Lab, the Robot Learning Lab, and the Robotics group at UW CSE. My gratitude goes to the members of these groups for their mentorship, collaborative discussions, and the much-needed breaks from research that

contributed to making these labs excellent work environments. I also extend my thanks to the undergraduate and master's students I had the chance to mentor. Their contributions to my work have been invaluable, and I am grateful for the opportunity to guide their academic and research journeys. Special thanks to all members of the Sensor Systems Lab for their incredible support. I deeply appreciate their prompt assistance in experiments, hardware manufacturing, and various other aspects whenever I urgently needed help. I am immensely grateful to Patrick Lancaster, who devoted countless hours to mentoring me and significantly shaping my development into the researcher I am today.

I extend heartfelt gratitude to my family for their unwavering love and support throughout this journey. A special thank you goes to my mother, whose inspiration and reassurance have been my stronghold during the most stressful times. Finally, my deepest appreciation to my wife, for being my steadfast companion since the onset of this endeavor. This achievement would not have been possible without you.

DEDICATION

to my parents, Suqun Yang and Lufei Li, my wife, Siqi Huang, and my soon-to-arrive daughter, Hannah

Chapter 1

INTRODUCTION

Games are deliberately crafted to test players in multiple domains, such as problem-solving, spatial reasoning, and manipulation aptitudes. These games often come with a well-defined scoring mechanism, making them ideal tools for assessing intelligence in humans and specific animals. As a result, engaging in games emerges as a fundamental avenue for enhancing cognitive abilities and bolstering overall intelligence. Take, for instance, the widely-enjoyed game Jenga. This game features a tower constructed of rectangular blocks, meticulously arranged for stability. During gameplay, participants strategically extract a block from the structure and place it atop the tower, striving not to destabilize it. Such active engagement in Jenga sharpens a player's focus, observational prowess, and coordination between vision and motion.

Research has demonstrated the efficacy of physical games in facilitating sensory-motor learning [97, 197, 167]. Sensory-motor learning can be described as the journey an organism undertakes to master specific movements and behaviors based on the sensory stimuli from their surroundings. Central to this learning process is the integration of various sensory feedback mechanisms—like visual cues, auditory signals, and proprioceptive responses—with motor commands. This integration aids in the evolution and finesse of motor skills over continuous practice and interaction. For infants and young children, who are incessantly navigating and engaging with their world, sensory-motor learning plays a pivotal role in their cognitive progression and understanding of their environment.

As a crucial embodiment of machine intelligence, robots are designed to engage with the physical realm and liaise with other smart entities to fulfill designated roles. These roles range from foundational actions like grasping objects and locomotion, to more sophisticated

undertakings such as complex manipulation and human-robot interaction (HRI). The ability of a robot to accurately discern ambiguities in its environment and then implement strategies with precision stands at the heart of its operational success. Engaging in physical games provides robots with a distinctive platform to hone their analytical abilities, perception acuity, and manipulation prowess by navigating and resolving complex physical challenges. Although there has been preliminary research delving into the role of robots in physical games, this domain remains largely uncharted, signaling an inviting arena for deeper investigation and scholarly pursuits.

Driven by these considerations, this thesis aims to draw a distinct correlation between robotics challenges and physical games. By framing challenges across different sub-fields of robotics as physical games, we introduced innovative methodologies that enhance robots' capabilities in four major areas: sequential manipulation, human-robot interaction, robot learning, and robot manipulation in warehouse settings.

1.1 Solving The Rubik's Cube: Sequential Manipulation

The Rubik's Cube, a 3D puzzle invented in 1974 by Hungarian Ernő Rubik, comprises six faces, each showcasing nine squares of a single color. The objective is to shuffle the jumbled cube until each face reflects just one color. This iconic cube has fostered a community dedicated to speedcubing competitions. Serving as a benchmark for spatial intelligence, memory, dexterity, and fine motor skills, the puzzle challenges human players. Speedcubers, individuals who specialize in quickly solving the Rubik's Cube, display a range of exceptional qualities. They possess strong spatial intelligence, allowing them to intuitively understand the cube's configuration and the effects of their moves. Their acute memory facilitates the memorization of numerous algorithms for cube resolution, often leaning on muscle memory for swift maneuvers. The sport demands excellent dexterity and fine motor skills for fast and precise cube manipulations. Furthermore, their problem-solving skills help them determine the best algorithm to apply for a particular scramble.

As robots increasingly move from functioning in controlled, meticulously designed set-

tings to human-centric, unstructured environments, the qualities needed to solve a Rubik's Cube, such as dexterity and motor skills, become ever more crucial. In this thesis, we utilize the Rubik's Cube as a test bed to assess a robot's proficiency in sequential manipulation tasks. We suggest Rubik's cube manipulation as a benchmark for gauging both precise and sequential manipulation performance. The intricate design of the Rubik's cube demands exact positioning from the robot's end effectors. Simultaneously, its highly adaptable configuration facilitates tasks that necessitate the robot to handle pose uncertainty across extended sequences of actions. We introduce a protocol to quantitatively evaluate the precision and speed of Rubik's cube manipulation. This methodology can be applied by any general-purpose manipulator and solely requires a standard 3x3 Rubik's cube and a flat surface for the cube's initial placement.

Inspired by the challenge of empowering a general-purpose robot to solve a Rubik's cube, we devised a new optical time-of-flight pre-touch sensor using cost-effective components. Subsequently, we demonstrated that this pre-touch sensor enables the robot to accurately re-estimate the pose of the Rubik's cube. This equips the robot with the requisite dexterity to solve the cube robustly. To broaden the applicability of this pre-touch sensor to everyday objects, we introduce a unique framework that synergizes pre-touch sensing with deep learning to optimize pose estimation efficiency. The integration of pre-touch sensing permits direct object localization concerning the robot's end effector, effectively bypassing errors from arm miscalibration. Rather than scanning the entire object with its pre-touch sensor, our system employs a deep neural network to identify areas of the object with pronounced geometric features. By zeroing in on these specific regions using pre-touch sensing, the robot can swiftly collect the requisite data to refine its initial pose estimation.

1.2 Competitive Games for Human-robot Interaction

Competition is a prevalent phenomenon, evident both in nature [24, 70] and human societies [32, 37, 60]. However, in the realm of Human-Robot Interaction, the spotlight has predominantly been on *cooperative interactions*—like collaborative manipulation, mobility

support, feeding, and more [11, 13, 22, 33, 101]. This bias towards cooperative endeavors isn't wholly unexpected. The underlying sentiment is simple: while people readily seek robots for assistance, who would desire a machine acting counter to their objectives? Yet, one cannot deny the constructive and enriching nature of human-human competition, especially in structured environments such as sports. Drawing inspiration from this, our paper delves into the potential benefits of competitive human-robot interactions.

The subsequent segment of this thesis is driven by an ambition to pioneer research in competitive human-robot interactions. Our vision is to design a robotic adversary adept at challenging humans in particular contexts like physical workouts and gaming. Steered by this vision, we present the Fencing Game—a platform for human-robot competition intended to assess the prowess of the robotic contestant as well as the user experience. We sculpted our robot competitor employing iterative multi-agent reinforcement learning and demonstrated its commendable proficiency against human opponents. Our user evaluation showed the system's ability to consistently craft engaging and stimulating encounters, notably amplifying participants' heart rates. The majority of participants considered the system for its entertainment value and potential to elevate the intensity of their exercises.

1.3 Game-theoretic Autocurricula

From our physical human-robot competition project mentioned in the preceding section, we discovered that, within a simulated environment, a robot can acquire highly proficient skills from a competitively co-evolving agent. Notably, these skills demonstrate resilience against both random and adversarial disturbances. From a machine-learning perspective, this particular paradigm of policy training is within the scope of autocurricula training. In the robotics community, autocurricula has experimented with physically grounded problems, such as robust control and interactive manipulation tasks. However, the asymmetric nature of these tasks makes the generation of sophisticated policies challenging. Indeed, the asymmetry in the environment may implicitly or explicitly provide an advantage to a subset of agents which could, in turn, lead to a low-quality equilibrium. This paper proposes a novel game-

theoretic MARL algorithm, Stackelberg Multi-Agent Deep Deterministic Policy Gradient (ST-MADDPG), which formulates a two-player MARL problem as a Stackelberg game with one player as the 'leader' and the other as the 'follower' in a hierarchical interaction structure wherein the leader has an advantage. In three asymmetric competitive robotics environments, we exploit the leader's advantage from ST-MADDPG to improve the quality of autocurricula training and result in more sophisticated and complex autonomous agents.

1.4 Gamifying a Warehouse Manipulation Problem

In previous sections, we demonstrated that physical games serve as an effective test bed for benchmarking specific robotic capabilities. Additionally, we illustrated how robots can acquire sophisticated real-world skills through learning in competitive games. In the final section of this thesis, we detail our efforts to gamify a real-world warehouse robot manipulation challenge—a practical issue that Amazon seeks to automate using robots. In our research, we first outline the warehouse manipulation task and then formulate it as a robot learning problem within a general sum game setup. Subsequently, we identify and implement the necessary software infrastructure for this gamification process. Finally, we will provide a brief overview of our future work to conclude this project.

In Chapters 4 and 5, we demonstrated that while the competitive autocurricula learning paradigm can lead to effective robot skills in real-world scenarios, it has a limitation: the need for a large number of training samples. Given the labor-intensive and resource-consuming nature of generating real-world data, there is a significant reliance on reliable simulations for realistic data generation, whether for training or pre-training purposes.

In Chapter 6, we delve into the warehouse manipulation challenge of object picking. In this task, a robot employs a vacuum suction cup gripper to grasp and extract a target object from a pile housed within a container featuring a side opening. By gamifying this task, our objective is to design a system where one robot sets up challenges for a picking robot, thereby facilitating automated data collection that continuously stows or rearranges items within a container. Utilizing the autocurricular learning framework, the system can

autonomously assign and retrieve target objects, minimizing the need for human intervention. This methodology allows robots to autonomously generate real-world data and hone their manipulation skills. Importantly, the data generation process is steered by the robot's existing capabilities, ensuring the data generated is skewed towards areas where the robot needs further refinement and training.

One primary challenges of this task arise from the movement of objects during the manipulation process and the realistic simulation of the suction cup's physical properties. To overcome these challenges and successfully gamify this task, the majority of our research effort is focused on developing a realistic physical simulation that faithfully emulates object dynamics and the physics of suction grasping. We present a novel solution to the challenge of suction grasp point detection. By harnessing the strengths of both physics-based simulation and data-driven modeling, our method considers object dynamics during the grasping process, significantly improving the robot's ability to manage previously unseen objects in real-world scenarios. We evaluate DYNAMO-GRASP against established methods in both simulated and real-world environments. Notably, it exhibits enhanced and consistent grasping performance across these settings. In real-world tests involving challenging scenarios, our approach boasts a success rate improvement of up to 48% over state-of-the-art (SOTA) methods. Showcasing a robust adaptability to intricate and unforeseen object dynamics, our method ensures robust generalization to real-world challenges. These findings pave the way for more dependable robotic manipulation in complex real-world situations.

1.5 Contributions

In this thesis, we delve deeply into the utilization of physical games as a means to assess robotic capabilities. This exploration not only inspires innovative methodologies that amplify real-world robotic performance but also unveils previously uncharted applications that remain unattained by other robots. Our primary contributions are as follows:

- **Benchmarking robot manipulation with the Rubik's Cube:** Use the Rubik's

Cube, a physical and mechanical puzzle game, as a test-bed to evaluate a robot's ability in sequential manipulation tasks. Inspired by the task of Rubik's cube solving, we developed a novel robot finger-tip sensing method that drastically increases robot manipulation precision.

- **Competitive games for human-robot interaction:** Investigate the potential for a competitive relationship between robots and humans by developing a robot that can challenge human users in physical exercises and games. Perform a user study to test whether our competitive robot is capable of creating challenging and enjoyable interactions.

- **Robot learning in competitive games:** Develop a game-theoretic Automatic Curriculum Learning (ACL) algorithm that allows robots to learn sophisticated emergent skills with improved sample efficiency and applicability to physically grounded tasks.

- **Gamification of a warehouse manipulation problem:** We've created a simulation environment that accurately reflects the physical properties of a suction cup and the dynamics of objects during the suction grasping process. This environment is designed to facilitate the advancement of ACL for mastering sophisticated manipulation skills in this warehouse setting.

In each of these contributions, we undertook comprehensive experimentation to fully comprehend how specific techniques or advancements propel the frontier of robotic capabilities. This deep dive illuminated the pathways to unlock novel applications and provided a critical understanding of the inherent challenges and limitations. Importantly, to ensure the practical relevance and robustness of our findings, all the methodologies we proposed were rigorously tested on actual robotic systems.

1.6 Publication Note

Most of Chapter 3 appear in three papers [185, 184, 87], and these are joint work with Patrick Lancaster. Most of Chapter 4 appeared in our paper "Motivating Physical Activity via Competitive Human-robot Interaction"[188]. Chapter 5 was adapted from our paper "Stackelberg Games for Learning Emergent Behaviors During Competitive Autocurricula"[190]. Most of Chapter 6 appeared in our paper "DYNAMO-GRASP: DYNAMics-aware Optimization for GRASP Point Detection in Suction Grippers" [189].

Chapter 2

RELATED WORK

In this section, we first review research on robots participating in physical games. Next, we explore game-centric studies in manipulation benchmarks, Human-Robot Interaction (HRI), and automatic curriculum learning - areas where we apply robotic physical gameplay. Additional works related to these topics, although not directly game-focused, will be discussed in their respective sections below.

2.1 Robots for Physical Games

Games have been employed to assess various capabilities of robots. Fazeli et al. [36] employed Jenga to devise a methodology that simulates hierarchical reasoning and multisensory fusion in robots. Chess has seen its fair share of implementation in robotics research. Juang [75] developed an effective visual control strategy within the context of chess playing, while Kolosowski et. al. [80] investigated the use of collaborative robots for chess, exemplifying robotic intelligence in complex problem-solving tasks traditionally performed by humans. Robotic soccer has seen significant advancements, particularly due to the RoboCup competitions, with progress in hardware design, sophisticated machine learning algorithms for decision-making, and improved teamwork strategies. Notable contributions include MacAlpine et al.'s approach to skill learning, Liu et al.'s architecture focusing on path planning and cooperative behavior, etc [104, 176, 3, 56]. Socially interactive robots utilize games to foster learning and education, facilitate therapy and rehabilitation, aid in skill development, promote social interaction and bonding, and serve as tools for human-robot interaction research [81, 14, 29, 26, 122].

2.2 Manipulation Benchmark with Rubik's Cube

Although several studies have explored various benchmarks for robot manipulation [179, 193, 169, 8], few have explored using Rubik's cube manipulation for the same purpose. Rubik's cube manipulation has mainly been used to showcase the capabilities of new algorithms and hardware, as demonstrated by Zieliski et. al. [202], OpenAI et al.[1], and Higo et. al. [64]. These studies provide detailed descriptions of the algorithms and systems used but stop short of proposing a standardized evaluation process for Rubik's cube manipulation. Another notable approach by Yang et. al. [184] analyzes the utility of pre-touch sensors in reducing object pose uncertainty during manipulation, setting a foundation for the baselines reported in this work. Lancaster et. al. [87] extend this method to general object geometries. Although the idea of Rubik's cube manipulation as a community-wide benchmark is supported by certain works [17, 201], they do not provide a standard process for evaluation. The first project in this article aims to develop a generalizable protocol and establish baseline scores for the broader research community.

2.2.1 Robot Perception for Precise Sequential Manipulation

During the process of enabling our PR2 robot to solve Rubik's cube puzzles successfully, we discovered that numerous uncertainties exist within the kinematic and perception systems of most general-purpose robots, which are nearly impossible to eliminate through meticulous calibration. This inspired our development of a novel fingertip sensing method (i.e. pre-touch sensing). This section will describe related works for robot perception for precise and sequential manipulation and fingertip sensing.

Robotic manipulation commonly employs cameras to localize target objects. Maitin-Shepard [107] introduced a vision-based algorithm for detecting cloth corners, enabling autonomous towel folding. Similarly, Chang [20] developed a perception module for singulating and manipulating object groups using image data. Cameras have also been integral in closed-feedback loops for visual servoing; Vahrenkamp [170] showed a humanoid robot using visual

servoing to grasp varied objects. For a comprehensive study on this, see [82].

While cameras facilitate distant object sensing, tactile sensors are vital for manipulation upon contact. These sensors, attached to a robot's end-effector, offer greater maneuverability and can sense areas often unseen by cameras due to their fixed positions. Li [92] employed GelSight technology in a tactile sensor to generate detailed height maps for part localization and intricate tasks like USB insertion. Petrovskaya and Khatib [135] used tactile measurements for object pose estimation, enhancing a robot's ability to locate and grasp items with a particle-based approach.

Pre-touch sensors fill the intermediate range between tactile and vision sensors, combining the advantages of both. Mounted on the end-effector, they are less occluded than cameras and can precisely measure without contact—unlike tactile sensors, which risk displacing objects upon touch.

Optical pre-touch sensors are widely used in grasping for their precise measurements across various materials. Hsiao [67] paired such sensors with a probabilistic model for a reactive grasp controller. Maldonado [108] reconstructed object shapes for grasping using an optical sensor inspired by computer mouse technology, aiding in surface classification and slip detection. Guo [55] demonstrated the efficacy of break-beam sensors for challenging specular objects. Unlike reflective-based sensors, our time-of-flight sensor measures distance directly, providing a long-range capability with high accuracy without the need for reflectivity calibration.

Electric field sensing, often used in pre-touch applications, detects conductive objects by transmitting an AC signal and measuring displacement currents. This technology has been used to localize objects for stable grasping and co-manipulation with humans, as shown by Wistort [180] and Mayton et al.[115]. M"uhlbacher-Karre[121] incorporated it into a robotic bartending system to gauge beverage levels.

Acoustic sensing is an emerging pre-touch technique. Jiang and Smith [72] developed a "seashell effect" sensor to detect objects difficult for optical sensors and applied it to object localization and grasping.

Previous works on pre-touch sensing focused on estimating local surface pose for single-shot grasping. Our research takes a novel direction by using ICP algorithms for full object pose estimation, critical for complex manipulations. This is a first in applying sparse pre-touch scans for complete pose determination.

Lastly, to bridge the gap between head-mounted cameras and pre-touch sensors, some researchers have mounted cameras on the robot's wrist [89] [76]. While this reduces the need for actuation, it can limit mobility due to the cameras' relative size.

2.3 Competitive Human-robot Interaction

The development of competitive human-robot gameplay hinges on advancements across various fields of robotics and artificial intelligence. This section delves into the foundational work that has paved the way for robots to engage in dynamic and strategic interactions within competitive environments. Reinforcement Learning (RL) in competitive games has emerged as a core methodology for training agents to perform complex tasks with a degree of autonomy that closely mimics human strategic thinking. The interplay between humans and robots in competitive settings is a nuanced area that explores not just the technical capabilities of robots, but also the psychological and social dimensions of human-robot interaction. Additionally, the utilization of robots in physical training represents an application of robotics where the physicality and adaptability of robots are leveraged to enhance human skill and fitness. The related work reviewed here provides the necessary backdrop for understanding the current landscape of competitive human-robot gameplay, highlighting the successes, limitations, and potential directions for future research.

2.3.1 Reinforcement Learning in Competitive Games

Competitive games serve as benchmarks for evaluating the capability of algorithms to train agents in making strategic decisions [153, 18, 159]. Multi-agent reinforcement learning (RL) enables agents to learn complex behaviors through interaction and co-evolution [65, 161, 59, 166, 160, 44]. Recent advancements have seen multi-agent RL used to develop continu-

ous control policies for complex tasks. Bansal et al. [5] designed policies for humanoid and quadrupedal robots in competitive games like soccer and wrestling. Lowe et al. [100] expanded the Deep Deterministic Policy Gradient (DDPG) [94] to a multi-agent context with a centralized action-value function.

2.3.2 Human-Robot Competition

Research on human-robot interaction within competitive games is less common. Kshirsagar et al. [84] examined the impact of robotic "co-workers" on human performance in a competitive environment with monetary incentives, finding that a high-performing robot slightly discouraged human competitors. Conversely, humans showed a positive response to a less capable robot. Mutlu et al. [125] observed that male participants were more engaged in competitive video games with an ASIMO robot but preferred cooperative modes of play. Furthermore, Short et al. [158] discovered increased social and mental engagement from human participants when the robot cheated in a game of "rock-paper-scissors".

2.3.3 Robots in Physical Training

Robots have also been utilized as assistants in physical training. Fasola and Mataric [35] developed a robot that provided real-time coaching for seated arm exercises. For indoor cycling, Süssenbach et al. [165] introduced a motivational robot that adapted its communication techniques to the user's condition, enhancing workout efficiency and intensity. In volleyball training, Sato et al. [152] created a system that replicated the movements and strategies of top-level blockers.

While insightful, these studies are confined to specific competitive scenarios involving simple, repetitive motions. Our research aims to expand this domain by applying reinforcement learning to develop a robotic system capable of engaging in a variety of physically demanding games against human opponents.

2.4 Robot Learning from Competitive Games

Multi-agent Reinforcement Learning (MARL) tackles the complex problem where multiple autonomous agents make decisions in a shared environment, each aiming to maximize its individual cumulative reward. Such environments can be competitive, forming a distinct subset within MARL often modeled as zero-sum games between two players through competitive Markov decision processes [39]. Significant strides have been made using autocurricular approaches to address these scenarios, particularly in symmetric games like chess and certain video games, where agents evolve by continuously generating and solving new challenges for each other [159, 10, 173]. The limitations of this approach become apparent when agents plateau in ability, signaling an equilibrium point due to insufficient adaptive challenges.

The robotics field has recently adopted competitive autocurricular approaches to tackle physically-grounded tasks including robust control and complex behavior learning [30, 4, 186, 187]. However, these robotic applications frequently involve asymmetric scenarios where agents have unequal tasks and capabilities, potentially leading to dominant behavior that stifles co-evolution and traps the system in suboptimal equilibria. For instance, in a simulated boxing match, an agent with a significantly stronger punch may continually win by knockout, preventing its opponent from discovering effective counter-strategies.

To circumvent these issues, some research has resorted to population-based techniques and expansive sample generation [4, 5]. Though effective, these methods demand high computational resources and often necessitate large-scale infrastructure. Alternative policy initialization approaches, such as reward shaping and imitation learning, have been proposed to set the agents on a more balanced starting point [186, 182]. Strategies like minimax regret, goal-conditioned policies, and intrinsic motivation have also been explored to prevent stronger agents from monopolizing the competition in adversarial settings [30, 131, 164]. Nonetheless, these strategies might not address the fundamental asymmetry of certain environments, and traditional MARL approaches with simultaneous gradient updates can suffer from convergence issues [44, 142, 200]. Although asymmetric gradient-based algorithms have

been explored in theoretical research for normal-form games [95, 38], there is a gap in applying this knowledge practically, which this paper seeks to address.

2.4.1 Learning for Suction Manipulation

The latter part of this thesis focuses on developing simulation software infrastructure that facilitates the gamification of robot manipulation tasks in a warehouse setting. In this scenario, the robot utilizes visual inputs and a suction cup manipulator to handle items within sideways open shelf containers. Accordingly, we discuss related work pertaining to suction-based and visual manipulation.

Robot manipulators employing suction technology have become increasingly prevalent across various fields. Manufacturing processes, for example, have integrated suction-based techniques for material handling [199, 191, 129], while storage and retrieval systems in warehouses also leverage this technology [58, 155]. In more specialized environments, like underwater settings, suction is used to handle objects with precision [163, 85]. The food industry, too, employs suction for handling delicate items, such as produce [25, 119, 48, 12]. An intriguing application lies in the augmentation of end-effectors, where novel designs such as multifunctional suction cups enhance grasping and manipulation capabilities [71, 69, 117, 126].

Analytic Models. For traditional suction cup grippers, precise modeling of the cups' physical properties is essential for evaluating grasp efficacy. Often constructed from flexible materials like rubber or silicone, these cups are frequently represented using spring-mass models to simulate deformation [105, 19, 143]. Once engaged, the cups are assumed to be inflexible for the analysis, where forces, including normal, frictional, and suction, are considered [79]. A sophisticated model introduced by Mahler et al. [105] combines torsional friction and contact forces into a singular, compliant contact model, proving valuable for grasp quality prediction and data annotation [19, 106].

Learning Suction Grasps. Optimizing suction grasp selection through machine learning has seen vigorous exploration for complex robotic manipulation [73, 120]. These systems have advanced to pick novel objects, sort them, and extract them from containers us-

ing training data derived from expert input or simulated environments [105, 19, 73, 156]. While research like DexNet3.0 [105] focuses on suction point efficacy and wrench resistance, other studies have used RGB-D imagery to predict potential grasp points in cluttered spaces [19, 156, 196]. Jiang et al. [73] took into account both the quality of the grasp and the accessibility by the robot for efficient bin picking. Our study contributes by examining the impact of object displacement during the grasp, an often-overlooked factor that can influence task success.

Visual Pushing. This research also intersects with studies on object movement during manipulation, particularly non-prehensile tactics, to aid in grasping operations [112, 102]. Visual-based methods have achieved notable progress in predicting object displacement, as seen with the Transporter algorithm and its extensions [195, 183], which offer a data-efficient approach linking sight to action. However, these rely on the assumption of consistent object movement, which may not hold in more complex environments. Visual foresight models have developed frameworks that predict future states from current actions and observations [40, 68], although these models can be computationally intensive when searching for optimal actions. While some studies have focused on the dynamics of side-on robotic manipulation [178, 34, 62], they typically do not detail the nuanced dynamics that occur during robot-object interactions, an area our work aims to elaborate on.

Chapter 3

BENCHMARKING ROBOT MANIPULATION WITH THE RUBIK'S CUBE

Although researchers agree that quantitative evaluations are crucial for tracking advancements in the realm of robotic manipulation, universally accepted benchmarks have been elusive. In related fields such as object recognition, object tracking, and natural language processing, standardized datasets serve as representative proxies, used to estimate algorithm performance in real-world scenarios. However, developing benchmarks that adequately represent these fields is challenging, and this issue is magnified in robotic manipulation. Unlike in other fields where observation of the environment suffices, robots must interact with their surroundings to accomplish tasks, adding another layer of complexity. Consequently, effective robotic benchmarks should possess clear, quantifiable measures, reflect aspects of real-world manipulation, and be accessible to a diverse range of robotic platforms.

In this section, we suggest using the Rubik's cube, a popular 3D combination puzzle, as a benchmark strenuously measure the robot's ability to simultaneously perform precise manipulation and sequential manipulation. This benchmark not only fulfills all the above-mentioned criteria, but the complexity involved in enabling a robot to manipulate a Rubik's cube also inspired us to develop a novel sensing method. This method permits robots to execute precise, long-sequence manipulations that can be generalized for everyday object manipulation.

3.1 The Challenges of Rubik's Cube Manipulation

The precision needed to manipulate a Rubik's cube stems from its unique structure. Each of its six faces is comprised of nine smaller cubes, arranged in a grid of three rows and

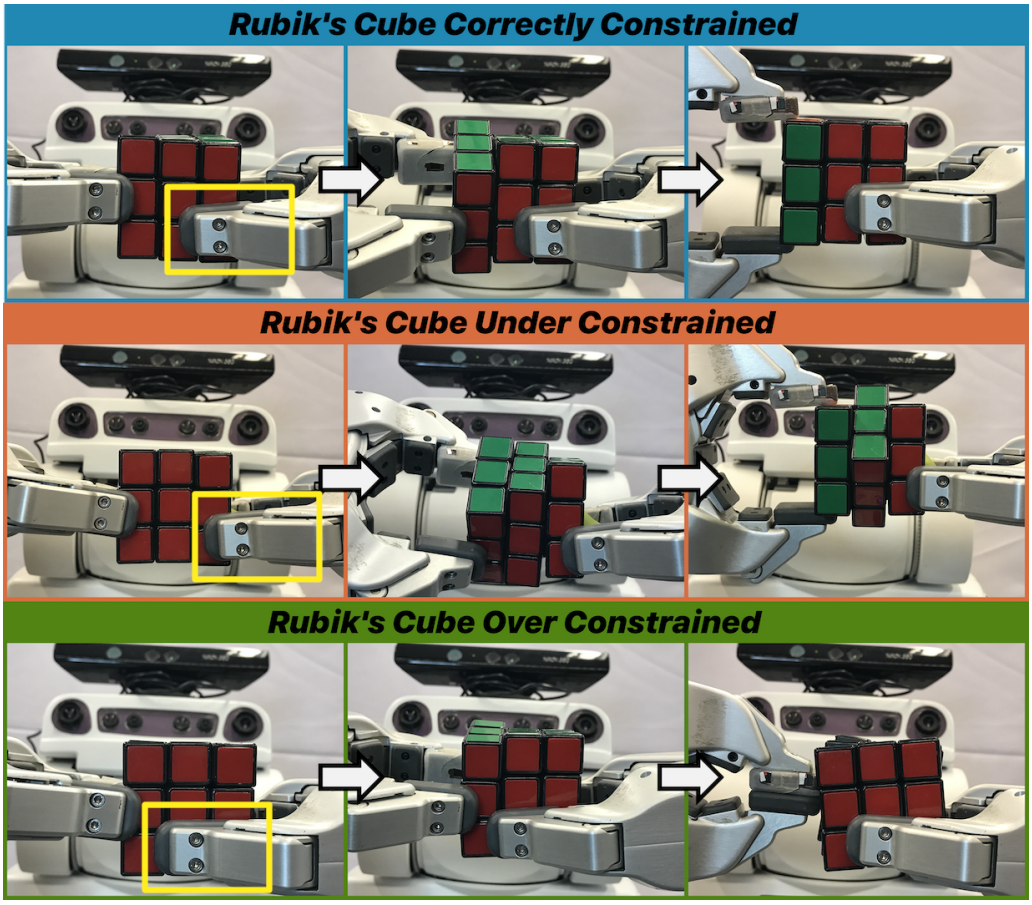

Figure 3.1: The robot must precisely position its grippers to rotate the left column of the Rubik's cube while constraining the middle and right columns in place. **Top Row:** The robot correctly positions its grippers: it is constraining the two right columns of the cube. The yellow box highlights the position of the constraining gripper. **Middle Row:** The robot only touches one column of the Rubik's cube and therefore fails to constrain its middle column. **Bottom Row:** The right gripper is touching all three columns of the cube; this prevents the left gripper from rotating the left column of the cube.

three columns. The cube's state can be altered by rotating a row or a column around an axis parallel to the respective columns or rows. However, when a particular row or column is rotated, the remaining ones on the same face must remain stationary. Given that each

smaller cube measures only 1.9 cm, this requires the robot to position its end-effectors with sub-centimeter precision.

Maintaining this level of accuracy is challenging due to the inherent uncertainty in the Rubik's cube's position relative to the end-effectors. This uncertainty, which can arise from various sources such as imperfect calibration of high-degree freedom arms, flawed actuators, and others, is a common issue for all general-purpose robots. As illustrated in Fig. 3.1, a lack of sufficient precision can lead to the unintentional rotation of *under-constrained* sections of the cube, or it may hinder the intended rotations of *over-constrained* sections.

Moreover, manipulating a Rubik's cube involves more than just executing a single precise rotation. For instance, solving the cube can require up to twenty rotations, according to Rokicki's study [149]. Each manipulation could introduce some degree of discrepancy between the robot's intended action and its actual execution, leading to errors in estimating the cube's pose. If the robot fails to utilize sensor feedback, take measures to control the cube's pose, or employ alternative methods to mitigate or address these errors, they will accumulate over time and ultimately result in a manipulation failure.

3.2 Protocol for Rubik's Cube Manipulation

We propose a two-dimensional protocol to evaluate both the speed and accuracy of Rubik's cube manipulation by a robot. This protocol comprises multiple tiers denoted as Rubiks-M-N, with M consecutive trials of N rotations each. Prerequisites include a standard 3x3 Rubik's cube and a flat platform. Initially, the cube should be placed at the center of the robot's workspace, and the robot should not be in contact with the cube. The procedure involves twelve specific tiers, such as Rubiks-1-5, Rubiks-1-10, up to Rubiks-5-200. For each tier, the robot must execute a specific number of consecutive manipulations as fast as possible. The final score for each tier is determined by the average time taken and its standard deviation. The capability and robustness of the system are assessed by the successful completion of higher tiers (larger N and M values) and the speed of completion, respectively. Please refer to our publication [185] for more details about the benchmarking protocol.

3.2.1 Protocol Prerequisites

The protocol necessitates a standard 3x3 Rubik's cube with a dimension of 5.7 centimeters for each edge and a flat platform for positioning the cube. We advise using the affordably priced and readily available 3x3 Hasbro Gaming Rubik's Cube, item number A9312. To minimize uncertainty or enhance manipulability, researchers or the robot can opt to utilize the flat platform. The Rubik's cube should be initially placed atop the table's surface so that its center aligns with the robot's workspace center in both x and y directions (relative to the robot's base frame). The cube's top face should be parallel to the ground, with its back face perpendicular to the robot's sagittal plane. Initially, the robot's manipulator(s) must avoid contact with the cube. Once the benchmarking process is initiated by the robot, human intervention is strictly prohibited.

To maintain benchmark consistency across different research groups and to assist in the validation of achieved scores, we offer dedicated software. The first module generates pseudo-random sequences of Rubik's cube rotations using a fixed seed, described in the standard Rubik's cube notation. The second module, after receiving the initial state of the Rubik's cube and a rotation sequence, provides the cube's anticipated final state. You can access the source code and usage instructions at https://gitlab.cs.washington.edu/bolingy/rubiks-cube-benchmark.

3.2.2 Protocol Details

This protocol gauges the speed at which a robot can execute a sequence of Rubik's cube manipulations. We introduce twelve distinct tiers for experimenters to tackle: Rubiks-1-5, Rubiks-1-10, Rubiks-1-20, Rubiks-1-50, Rubiks-1-100, Rubiks-1-200, Rubiks-5-5, Rubiks-5-10, Rubiks-5-20, Rubiks-5-50, Rubiks-5-100, and Rubiks-5-200. For instance, in the Rubiks-5-100 tier, the robot must lift the Rubik's cube and complete a 100-rotation sequence of manipulations five consecutive times. The first half of these tiers offers an optimistic assessment of the robot's capabilities, while the latter half assesses the consistency and reliability of

its performance. It's noteworthy that if a robot successfully completes the Rubiks-1-20 tier, it demonstrates adequate manipulation accuracy to solve any Rubik's cube configuration.

The protocol consists of the following steps:

1. Experimenter or the robot decides which tier to attempt
2. Robot acquires manipulation sequence from the provided software
3. Experimenter places the Rubik's cube on the surface in front of the robot as defined in Sub-section 3.2.1
4. Robot picks up cube and begins to execute the manipulation sequence
5. Robot terminates manipulation
6. Experimenter validates that the final cube state is correct using the provided software
7. Return to step two if there are remaining trials to be completed

For each trial, if the Rubik's cube reaches the desired final state, the system's score is the time taken from the robot's initial contact with the cube until the manipulation concludes. The final score for the tier is determined by the average trial time and its standard deviation. Experimenters are expected to report any tiers they complete, the associated speed scores, and provide a clear video recording of these accomplishments. For a specific value of M, completing tiers with larger N values suggests superior manipulation accuracy. Conversely, finishing a given tier in a shorter time indicates improved manipulation speed. For a specific N value, completing a tier with a larger M signifies enhanced system robustness.

3.3 Pre-touch Sensing for Sequential Manipulation

During the creation of our benchmark protocol and while collecting baseline data, we observed that a general-purpose robotics research platform, like the PR2 robot, has an approximate manipulation error of 1 cm. This implies that when the robot aims to move its

hand to a designated pose, the hand could deviate by up to 1 cm from the desired position. This level of uncertainty makes it challenging for the robot to solve a thoroughly scrambled Rubik's cube, as it typically makes a critical error after around nine rotations. In this study, we illustrate that accumulated manipulation errors can be mitigated using proximity sensors attached to the robot's end-effectors, termed 'pre-touch' sensors. We first prove that pre-touch sensing enhances a robot's ability to complete specific sequential manipulation tasks, such as solving a Rubik's cube. Further, to show the versatility of pre-touch sensing across various objects requiring sequential manipulation, we demonstrate that even a rudimentary pre-touch scanning approach enables the robot to determine the pose of several everyday objects.

In this section, we underscore the efficacy of pre-touch sensing in enhancing a robot's capability to execute sequential manipulations. We posit that pre-touch scanning empowers a robot to acquire essential geometric data, facilitating an accurate estimation of an object's pose. This, in turn, allows the robot to undertake actions pivotal to sequential manipulation, such as re-grasping. The distinct contributions of this study include:

1. We introduce a novel form of pre-touch sensing that utilizes optical time-of-flight measurements and seamlessly integrate this sensor into the PR2 system.

2. We apply and evaluate pre-touch sensing for robot manipulation specifically in the context of solving the Rubik's cube. To our knowledge, this is the first application of pre-touch sensing to Rubik's cube solving. Additionally, we posit that solving the Rubik's cube represents the most intricate sequential task that pre-touch sensing has been employed for to date.

3. The deployment of the Iterative Closest Point (ICP) algorithm to align a 1D pre-touch scan with a reference point cloud, facilitating object pose estimation.

4. A comparative analysis of the optical time-of-flight pre-touch sensor (integrated into

one finger) and our previously developed electric field pre-touch sensor (fitted into another finger), revealing that these two sensing methods complement each other.

3.3.1 Sensor Hardware

In this study, we examine the application of two distinct pre-touch sensors to aid in sequential manipulation. The first sensor is an electric field variant, adapted from the research by [115]. The second is an innovative optical time-of-flight sensor, which will be elaborated upon in the subsequent sections of this chapter. Both sensors were encased to align with the PR2's fingertip dimensions. On the robot's parallel jaw grippers, the left fingertip was equipped with the optical sensor, while the right fingertip incorporated the electric field sensor. When attempting to solve the Rubik's cube, only the optical sensor was utilized, as the electric field sensor lacks the capability to effectively detect the plastic material of the Rubik's cube.

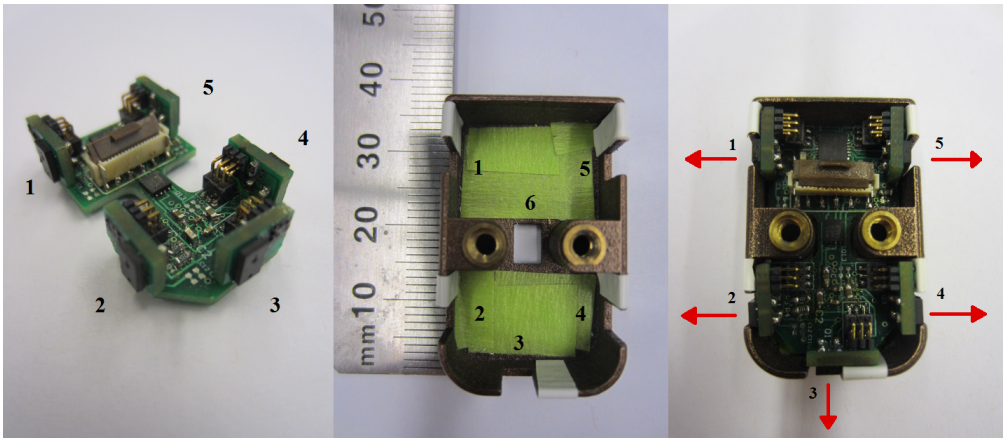

Figure 3.2: Left: The printed circuit boards that compose the sensor. Middle: The 3D printed sensor casing. The hole in the middle of the case is for a sensing module soldered to the bottom of the main PCB. Right: The assembled sensor with arrows denoting the directions of five out of six sensor modules' infrared beams.

The optical pre-touch sensor gauges the proximity to a surface employing the VL6180x optical time-of-flight module developed by ST MicroElectronics. This sensor can accurately

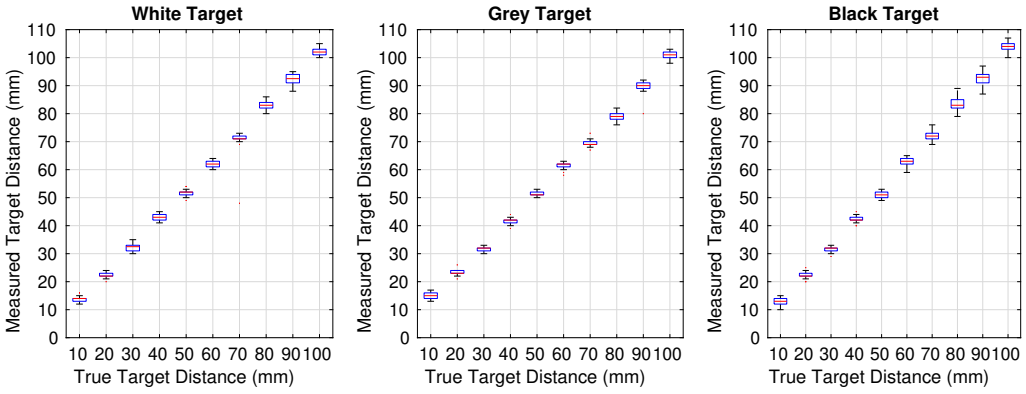

Figure 3.3: Boxplots of sensor measurements over the specified range for white, grey, and black target objects. Each box consists of 30 measurements.

determine the distance to an object with millimeter precision, within a range of 1cm to 10cm. Additionally, certain objects, especially those of lighter shades, can be detected up to a range of 25.5cm. The sensor's performance across different colored targets is illustrated in Fig. 3.3.

The sensor accommodates up to six VL6180x sensing modules: one at the fingertip's tip, two on each side, and one on the finger's pad. Each module's location is depicted in Fig. 3.2 and will be referred to accordingly throughout this work. The sensor's architecture comprises a primary board (29x16.5mm) and a secondary board (8.25x5.75mm). The main board is equipped with an ATMega168PA microcontroller, which communicates with the VL6180x sensing modules via I2C. The module on the fingertip's pad is soldered directly to the main board. All other modules are connected to the main board through 1mm pitch headers, bridging the main and secondary boards.

The robot can gather data from each of the six sensing modules at a rate of 30Hz. These measurements are transmitted from the sensor's microcontroller to the robot via an SPI interface integrated into the gripper and subsequently published to a ROS topic. The sensor's casing is equipped with two metal screw inserts, allowing it to be securely attached to the robot's fingertip.

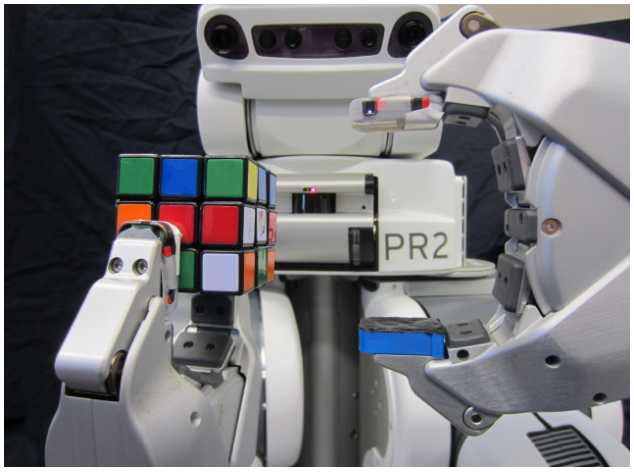

Figure 3.4: The robot is able to precisely manipulate the Rubik's cube using the equipped pre-touch sensors.

3.3.2 Method

Throughout this study, the robot employs basic scanning techniques to determine the pose of various objects. The fundamental nature of these scans indicates their potential applicability to a diverse set of objects. Subsequent subsections will elaborate on the scanning methods utilized, while Section 5 will showcase their efficacy. In this document, unless otherwise specified, any coordinate references pertain to the coordinate frame of the gripper currently holding the object. Here, the y-axis aligns with the gripper's opening and closing direction, the x-axis projects in the direction of the robot's fingertips, and the z-axis is perpendicular to both x and y-axes, following a right-handed coordinate system.

Optical Pre-touch Scanning for Rubik's Cube

In this study, pre-touch sensing not only helps marginalize positional errors for subsequent grasps but also rectifies any errors that emerge from one manipulation to another. As the robot endeavors to solve the Rubik's cube, it needs to transfer the cube between hands and modify its grasping approach. Before every re-grasp, the robot deploys pre-touch scanning

to enhance its pose estimation of the cube relative to the gripper holding it. While the robot assumes the cube's upper and lower faces to be approximately parallel to the ground, this assumption, albeit not always accurate, proves efficient in practice. Moreover, because the cube is nestled between the robot's fingers, a reliable approximation of the cube's center position in the y-direction and its y-axis rotation is present. However, unintended shifts in the x and/or z directions might arise from errors in prior re-grasps. Numerous pre-touch scanning strategies exist for position estimation in these directions. Our adopted strategy, designed to reduce actuation requirements, is as follows:

1. If not already open, the gripper not holding the cube is opened.

2. The optical pre-touch sensor on this gripper's fingertip is aligned so that the beam from sensing module 3 is perpendicular to one of the cube's faces, which in turn is orthogonal to the xz-plane, as depicted in Fig. 3.4. The gripper's position ensures the cube does not interrupt the beam.

3. The gripper starts to close, driving the sensor in the y-direction. Significant sensed distance alterations during closure pinpoint the cube's edge location, enabling the robot to deduce the cube's position along one of the uncertain axes.

4. After the gripper's closure, the robot harnesses the sensor's distance measurements at the current location to gauge the cube's position along the other uncertain axis.

This pre-touch scanning technique is amalgamated into a foundational system that utilizes a computer vision module for Rubik's cube face color recognition, an iterative deepening A* search [149] [148] to identify essential cube rotations, and a finite-state machine-driven motion planner to execute the requisite trajectories for cube solution.

Pre-touch Scanning for Common Objects

Pre-touch scanning proves versatile, applicable even to objects of intricate geometry. Our objective is to illustrate that a rudimentary 1D scan of a generic object can encompass distinctive features sufficient for pose estimation when aligned with a reference model. Such an estimation could be invaluable during the initial stages of object pickup or preceding a re-grasp.

There exists a plethora of potential scanning strategies. One approach might involve executing random trajectories until the robot confidently discerns the pose. Alternatively, trajectories grounded in heuristics or learning models can be pursued. The genesis of such trajectories will be a subject of our future studies. For the present study, the experimenter handpicked a single trajectory for each object, ensuring the trajectory's potential to highlight distinctive features. Every selected trajectory had the scanning gripper travel linearly with a set orientation. Throughout each trajectory's span, the robot methodically sampled the object at discrete intervals. The distance between these sampling points was gauged based on the object's dimensions, aiming for an approximate collection of 50 samples. The sampling methodology diverged slightly contingent on the pre-touch sensor in operation.

With the electric field pre-touch sensor, the robot aligned the forefront of its fingertip perpendicular to the trajectory, facing the object. The object influences the sensor readings by diverting the displacement current from the electrode situated at the fingertip sensor's forefront. At every sampling instance, the robot either advances or retracts its gripper relative to the object, altering the quantum of current diversion and, by extension, the deviation from the baseline measurement (the metric recorded when the object is distant from the sensor). The robot determines a distance measurement by adjusting its gripper until the variance from the baseline measurement aligns closely with a pre-set threshold. Documented prior to scanning and specific for each object, this threshold signifies a 1.5cm distance from the object. This methodology presumes the object's volume proximate to the fingertip remains constant throughout the trajectory. Although this presumption often fails

in reality, experience has shown that a reasonably accurate point cloud can be derived using this approach, as elucidated in Section 5.

When employing the optical pre-touch sensor, the robot, yet again, aligns the forefront of its fingertip perpendicular to the trajectory, facing the object. At each sample point, the robot's fingertip-located sensor module gauges the distance to the object. Contrasting with the electric field sensor, this sensor mandates no additional adjustments besides trajectory movement, as the optical sensor directly reports the object's distance at every sample point.

3.3.3 Experiment – ToF Sensor Properties

The subsequent two experiments were conducted to evaluate the efficacy of pre-touch sensing in assisting robots with sequential manipulation tasks. In both experiments, 1D pre-touch scans were employed to estimate the pose of an object by juxtaposing the gathered data with a reference model. By recalibrating the estimation of the object's pose, the robot can rectify errors from previous manipulations, paving the way for enhanced precision in subsequent tasks.

Rubik's Cube Manipulation Evaluation

To assess the utility of pre-touch scanning for solving the Rubik's cube, we devised a system, briefly outlined at the conclusion of Section 4a, for cube manipulation. This system could operate with or without the pre-touch sensing enabled. Without pre-touch sensing, the system establishes a benchmark representing achievable outcomes. Rather than scanning the cube post every re-grasp, the benchmark system operates under the presumption that the robot has flawlessly re-grasped the cube at the exact intended location.

Setup: We created 10 random cube configurations, each necessitating 20 to 23 rotations for resolution. Both the benchmark and the pre-touch enabled variants of the system were tested against these configurations. Alongside reporting the success and failure rates, we also analyzed the robot's perception of the cube's position throughout each trial. Prior to any re-grasp, the anticipated position of the cube, as determined by the robot, was documented.

All pose predictions were referenced to the frame of the gripper actively holding the cube. For accurate position measurements of the cube (referred to as "ground truth"), we affixed an AR tag to every face of the cube. To ascertain the cube's position, a Kinect, positioned on the robot's head, was employed in tandem with other external cameras. Each robot gripper was also labeled with an AR tag, placed at a consistent distance from the gripper's coordinate frame. This setup enabled us to compute the pose estimate with pre-touch enabled (along with the corresponding ground truth) without relying on the robot's coordinate transformations. Notwithstanding our meticulous recalibration of the robot before initiating this research, substantial discrepancies remained in the coordinate transformations— a commonplace issue for high-degree-of-freedom robots. It's pivotal to note that, while we utilized AR tag detection for ground truth estimates of cube pose, we are not suggesting its superiority over pre-touch sensing. This methodology inherently carries potential for errors based on the efficiency of tag detection. Moreover, it necessitates the attachment of one or more tags to any object for which a pose estimate is sought.

Table 3.1: End-to-end Rubik's Cube Solving

Method \ Result	Success	Fail	Avg. Rotations Completed
Baseline	0	10	9.6
Pre-touch	8	2	20.1

End-to-end Results: The experiment underscored the considerable advantages of pre-touch sensing in enhancing the robot's capability to resolve the Rubik's cube, as detailed in Table 3.1. With the incorporation of pre-touch sensing, the robot was able to successfully decipher 8 out of the 10 cube configurations. In the two unsuccessful attempts, the robot managed 14 and 19 rotations out of the required 21 and 20 respectively, before facing chal-

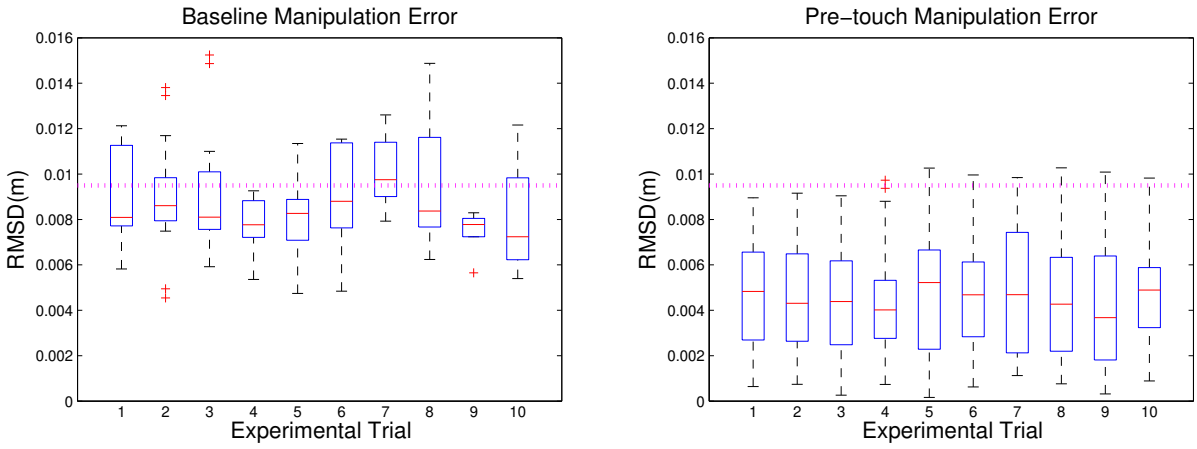

Figure 3.5: Box plots plots of positional error for the baseline (left), and corrected pre-touch (right) methods. Each box corresponds to one of the 10 trials and consists of all cube pose RMSD errors observed during that trial. The RMSD error is recorded prior to each re-grasp. The horizontal line across each plot denotes half of the dimension of a sub-cube, demonstrating that the increased dexterity provided by pre-touch sensing is significant for this task.

lenges in completing a rotation. In stark contrast, the baseline approach failed to solve even a single puzzle. Across the 10 trials, the highest number of successful rotations achieved was 17, with the lowest being 3 and an average of 9.6. Every failed rotation was attributed to the robot's inability to re-grasp the cube—either during the transfer from one gripper to the other or during an attempt to grip the cube for a face rotation. These findings underscore that while re-grasping actions are susceptible to positional errors, pre-touch sensing enables the robot to adeptly rectify these discrepancies.

Intermediate Pose Estimation: A comparison of the robot's positional estimates for the Rubik's cube, using both methods against the ground truth, revealed insightful observations. Before each re-grasp, the robot recalibrated the cube's pose. The discrepancy for each prediction was gauged by determining the root-mean-square deviation (RMSD)

between the x and z axes of the estimate compared to the ground truth. The aggregated RMSD for all trials using both methods is illustrated as box plots in Fig. 3.5.

The data indicates that pre-touch sensing significantly optimized the accuracy of the robot's pose estimates for the Rubik's cube. The robot's finger width is approximately equivalent to a Rubik's cube's sub-cube. Considering the ideal grasp point is situated between two sub-cubes, the acceptable error margin equates to around half the width of a sub-cube. Errors exceeding this margin risk the robot inadvertently blocking one of the cube faces, thereby leading to subsequent manipulation failures. Furthermore, surpassing this error threshold might result in the robot not grasping the cube at all. The left-hand visual in Fig. 3.5 reveals that in the majority of trials, the baseline method's pose estimates often had an RMSD exceeding half a sub-cube's dimension. Conversely, the right-hand visual emphasizes that the RMSD for pre-touch assisted estimates seldom crossed this mark. Indeed, with the pre-touch enabled technique, a significant majority of pose estimates recorded an RMSD under 0.8cm. This suggests that leveraging pre-touch scanning has enabled the robot to consistently reduce pose estimation errors, enhancing its cube-solving proficiency.

Expanding Pre-touch Scanning to Everyday Objects

Although pre-touch sensing proved highly effective in manipulating the Rubik's cube, our subsequent experiment delves into its applicability to more generic manipulation tasks. In this setup, we employ the Iterative Closest Point (ICP) algorithm to align a singular, straightforward pre-touch scan with a corresponding reference model. Future manipulation systems could leverage the transformation deduced between the pre-touch scan and reference model to determine the object's orientation and position. Such a prediction can be instrumental, both when initially grasping the object and during subsequent re-grasps throughout the manipulation process.

Experimental Setup: We undertook an assessment of pre-touch scanning across seven routinely encountered items that might be involved in sequential manipulations: a metal bowl, banana, lemon, coffee can, hammer, bell pepper, and a glass soda bottle. Each item

underwent separate scans using two distinct pre-touch sensors: an electric field sensor and the optical described above. Each sensor possesses its optimal working conditions. Specifically, the optical pre-touch sensor exhibits proficiency in detecting non-transparent items. In contrast, the electric field sensor is primarily effective in discerning objects with a dielectric constant notably distinct from air. Furthermore, while the optical sensor boasts a lengthier, more slender sensing area, the electric field sensor features a more compact, broader sensing zone. It's plausible that future systems might synergize data from both sensors, harnessing their combined capabilities to yield more precise estimations across a broader spectrum of objects than when relying on either sensor in isolation. For the scope of this study, however, we focus on evaluating the singular capabilities of each sensor in furnishing geometrically distinctive features that aid in pose estimation via pre-touch scanning.

Table 3.2: ICP Fitness Score

Sensor \ Object	1	2	3	4	5	6	7
E-field(10^{-6})	6	89	7	17	9	15	24
Optical(10^{-6})	23	24	6	24	286	33	210

Discussion and Implications: The investigations into ICP matching have shed significant light on the efficacy of pre-touch scanning in determining the pose of everyday objects. Figure3.6 visualizes the results of applying pre-touch scanning and ICP to seven common objects. The first column shows each of the seven objects and the region that was scanned in green. The second column shows the raw point cloud obtained by the electric field sensor for each object, and the third column shows how the ICP algorithm matched the raw point cloud to the reference model. The raw point cloud is colored green if the ICP algorithm found a correct partial or full match, and red if it failed. The fourth and fifth columns display the analogous results for the optical pre-touch scans. Some takeaways from the findings are:

1. **Material Composition Impact:** The electric field sensor's performance with certain objects, specifically the bowl and the hammer, underscores the sensor's ability to efficiently detect metal objects. This is consistent with the principles behind how electric fields interact with metal surfaces.

2. **Limitations of Optical Sensor:** Optical sensors have inherent limitations when it comes to dealing with reflective surfaces or complex geometries. As noted with the hammer, despite capturing the basic form, the complexities, especially at the claw end, affected the matching process. Such challenges are not limited to metal objects but extend to transparent ones as well, as evidenced by the results with the soda bottle.

3. **Varied Results with Organic Objects:** The analysis of fruits highlighted the intricacies in capturing detailed features. The bell pepper's hollow nature posed challenges for the electric field sensor, emphasizing the sensor's sensitivity to the internal constitution of objects.

4. **Potential for Combined Approach:** One of the key insights from this experiment is the complementary nature of the sensors. While one sensor might fail to obtain pertinent geometric data for an object, the other might succeed, hinting at the potential benefits of a combined sensing approach.

5. **Towards Customized Matching Techniques:** Off-the-shelf ICP, while powerful, may not be the most efficient choice for 1D scans. The possibility of designing specialized matching techniques tailored to these scans could further enhance pose estimation results.

In conclusion, pre-touch scanning, while still in its early stages, holds great promise for robustly identifying the pose of a diverse array of objects. By judiciously integrating data from varied sensors and possibly developing specialized matching techniques, this approach could find wide applications in robotics, particularly in scenarios involving object manipulation.

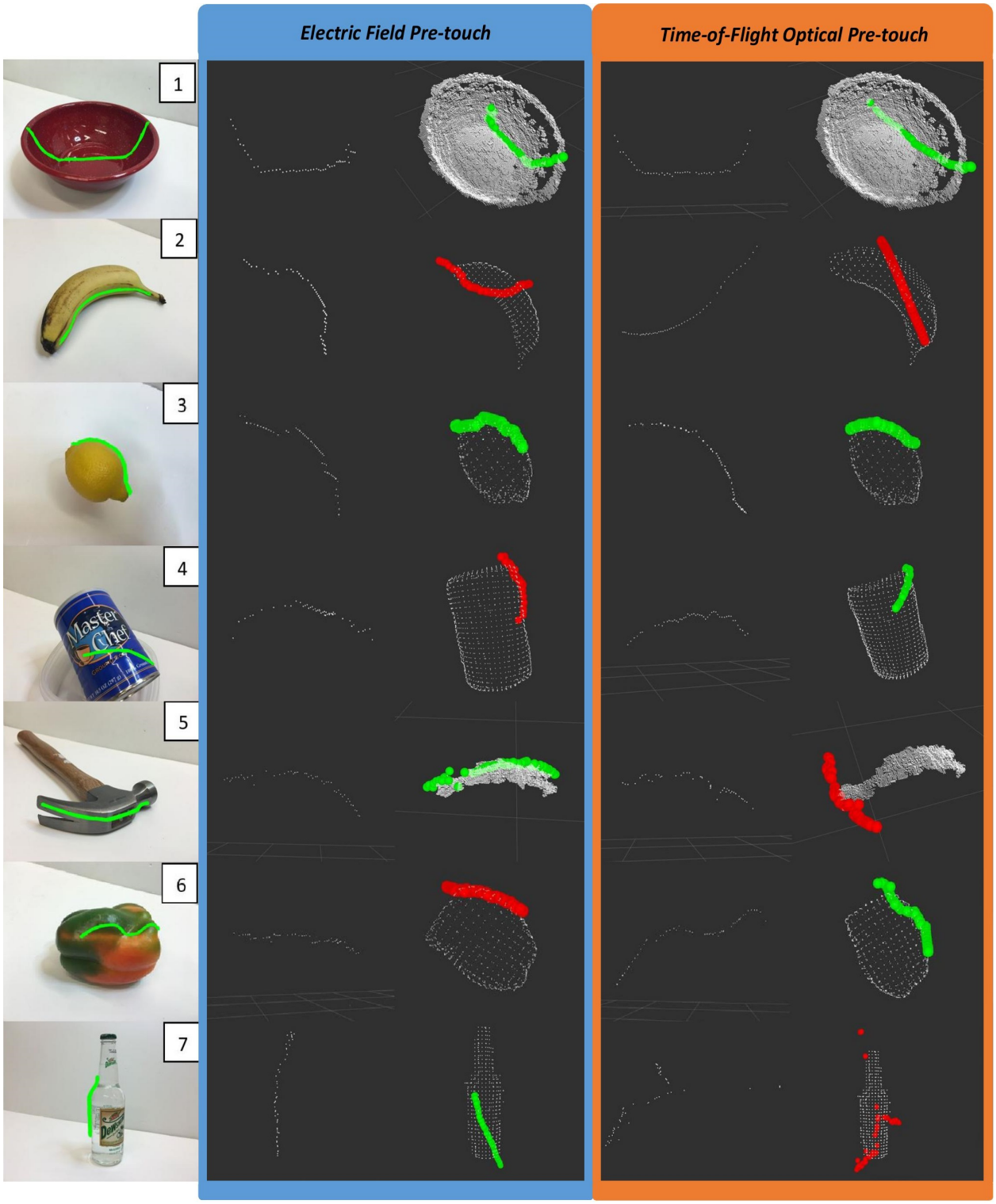

Figure 3.6: The results of applying pre-touch scanning and ICP to seven common objects.

3.4 Baselines for Rubik's Cube Manipulation

Based on our research in Section 3.3, we conducted extensive experiments using the Rubik's cube benchmark protocol. We have established baseline data for the robotics community, which can be used for comparisons in future research. Similar to the experimental setup in Section 3.3, in the first baseline, a PR2 robot localizes the Rubik's cube using its head-mounted camera and then attempts to execute sequences of manipulations without any additional sensor feedback. For the second baseline, while the initial pose of the cube is also determined through the head-mounted camera, the PR2 robot's end-effectors are equipped with our optical pre-touch sensors. This enables the robot to re-localize the Rubik's cube immediately before each re-grasp. Subsequent sub-sections will delve deeper into the details of each baseline and present the benchmark scores achieved with them.

3.4.1 Baseline Results

The dead reckoning baseline finished Rubiks-1-20 in 463.45 seconds but couldn't tackle any of the more challenging single trial tiers. While the system demonstrates adequate accuracy to solve a Rubik's cube over several attempts, it only managed to complete the lowest tier in five consecutive trials. It averaged a manipulation time of 113.35 seconds with a standard deviation of 17.16 seconds. Without continuous object state estimation, the system struggles with uncertainties that accumulate during sequential manipulation. After just 5 rotations of the Rubik's cube, the robot's pose estimation of the object becomes largely inaccurate, hindering its ability to achieve 10 or more rotations across 5 consecutive trials.

In contrast, the fingertip sensor-aided baseline displayed superior performance in terms of both speed and accuracy during single trial Rubik's Cube manipulation. It completed Rubiks-1-20 in just 447.96 seconds and managed Rubiks-1-100 in 2207.97 seconds. This configuration showcased greater robustness, even completing Rubiks-5-20. However, when juxtaposing the Rubiks-5-5 outcomes of both baselines, the introduction of sensing actions in the enhanced baseline led to a more varied speed protocol score. A detailed breakdown of

Baseline / Tier	Dead Reckoning PR2 (Avg/Std Dev)(s)	Sensor Aided PR2 (Avg/Std Dev)(s)
Rubiks-1-5	139.07 / -	126.67 / -
Rubiks-1-10	248.43 / -	215.37 / -
Rubiks-1-20	463.45 / -	447.96 / -
Rubiks-1-50	–	1123.04 / -
Rubiks-1-100	–	2207.97 / -
Rubiks-1-200	–	–
Rubiks-5-5	113.35 / 17.16	113.81 / 18.35
Rubiks-5-10	–	214.71 / 30.39
Rubiks-5-20	–	416.61 / 23.10
Rubiks-5-50	–	–
Rubiks-5-100	–	–
Rubiks-5-200	–	–

Table 3.3: The scores (mean and standard deviation of manipulation time) achieved by our baseline approaches on each of the tiers. Each row corresponds to a separate tier, Rubiks-M-N. Rubiks-M-N consists of M consecutive trials, where in each trial the robot must pick the Rubik's cube up off of the table and complete N rotations.

all completed tiers by our baselines and their respective scores can be found in Table 3.3.

3.4.2 Discussion

We have introduced a benchmark tailored to assess precise, sequential manipulation across an array of robotic platforms. Nonetheless, our design choices might not resonate with everyone. Specifically, while our benchmark gauges the overall speed and accuracy of Rubik's cube manipulation, it doesn't shed light on the performance of individual manipulative

actions in the sequence. This omission was driven by pragmatism. Manually recording such statistics might interrupt the manipulation process. Additionally, leveraging external systems for capturing ground truth, like motion capture, presents challenges such as occlusion and standardization discrepancies among research groups.

Our benchmark offers a versatile foundation for exploring various research avenues beyond the baselines presented. For instance, basic visual servoing techniques could encounter significant occlusion during manipulation. Our benchmark stands as a tool for assessing visual servoing approaches that specifically tackle occlusion challenges during manipulation [109, 96, 157].

In another dimension, planning techniques that factor in uncertainty might behave less restrictively than our fingertip sensor-assisted baseline. These planners, being cognizant of uncertainty, can more astutely ascertain the moments when uncertainty reduction is paramount [130, 15, 171]. The performance enhancements triggered by such algorithms can be quantified through our benchmark. Lastly, echoing the sentiments of Ma and Okamura [103, 128], robots possessing versatile kinematics or pronounced dynamic abilities are crucial for achieving genuine dexterity in manipulation. Notably, superior in-hand dexterity has been realized via external force application [28], optimal control methodologies [86], and deep reinforcement learning techniques [2]. Our benchmark can be employed to gauge the efficiency, robustness, and versatility of these strategies in comparison to one another.

3.5 Improving Generalization for Pre-touch Sensing via Deep Learning

The adoption of pre-touch sensors addresses the challenges associated with completely eliminating calibration discrepancies between a robot's head-mounted sensors and its end-effectors. Notably, before this study, our robot underwent calibration using established procedures [141]. Figure 3.7 depicts the typical extent of mis-calibration, emphasizing how pre-touch sensing can provide a more precise estimation of an object's pose. In practice, while the robot's fingertip's inner surface aligns with the bowl's edge, the data from the head-mounted Kinect suggests otherwise. However, readings from the optical pre-touch sensor align much more

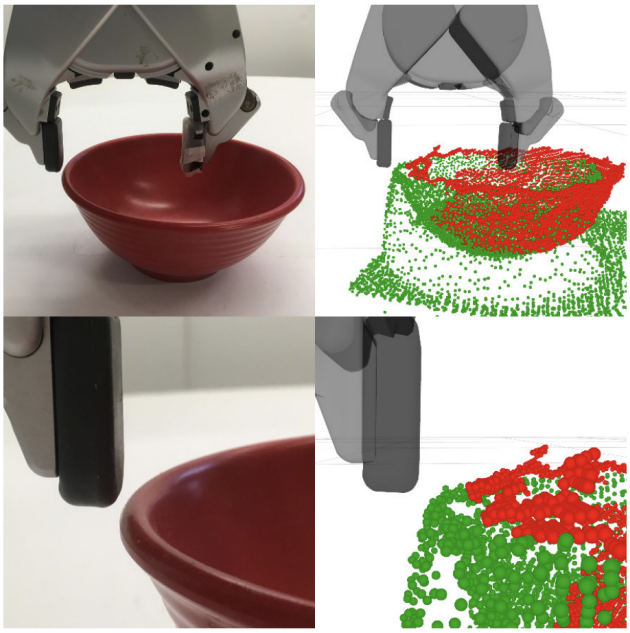

Figure 3.7: Left: A far and close view of the surface of the robot's fingertip aligned with the edge of the bowl. Right: Pre-touch measurements (green) and Kinect measurements (red) with respect to the fingertip.

closely with the actual scenario.

As illustrated in previous sections, accurate object pose estimation is crucial for certain robot manipulation tasks. Nonetheless, head-mounted sensors might not deliver the required accuracy due to imperfect calibration between the sensor and the robot's end effector. We've demonstrated that utilizing an optical pre-touch sensor mounted on the robot's end effector to gather geometric information about the object can circumvent the aforementioned calibration error. Subsequently, we employ the iterative closest point (ICP) algorithm to derive a more precise pose. In this section, we further refine this pose estimation framework, enhancing its performance for everyday objects using deep learning.

3.5.1 Data Registration

Given a point cloud of an object from a head-mounted Kinect, and a more accurate point cloud from a pre-touch sensor scan, it's essential to identify correspondences between these two data sets to determine the difference between them. The centroids of these point clouds are anticipated to have a discrepancy of a few centimeters, with both clouds maintaining similar overall orientations. The ICP algorithm is apt for aligning two point clouds with such a misalignment, rather than a complete registration approach that would be necessary for significantly misaligned pairs. By using ICP to align the two clouds, the algorithm provides a spatial transformation between the robot's previous, error-prone belief (from the Kinect cloud) and its updated belief (from the pre-touch sensor cloud).

While this strategy enables the robot to re-evaluate an object's pose, performing a complete pre-touch scan of the object can be time-consuming in terms of actuation. A more efficient approach would involve the robot re-estimating the pose by scanning only a section of the object and then aligning it with the corresponding segment from the original Kinect cloud. The challenge here is that as the amount of pre-touch data decreases, the scan becomes less distinctive, making the alignment more vulnerable to errors from noise and differing sensor attributes. However, by selecting specific areas rich in unique geometric features for scanning, the robot can acquire data effectively while still achieving accurate pose re-estimation. The method to identify such areas for scanning is elaborated upon in the subsequent section.

In this study, we employ the Point Cloud Library's (PCL) ICP implementation [150][66]. In addition to PCL's default ICP approach, we utilize a correspondence rejector to mitigate the impact of outliers [136].

3.5.2 Learning Where to Sense

We suggest leveraging a deep neural network to pinpoint object regions conducive to pre-touch sensing. These regions will be endowed with unique geometric features, enabling the

robot to precisely re-estimate an object's pose. These areas are parameterized as rectangles on the image plane. For this purpose, we adapt a cutting-edge object detection neural network. To produce training samples for the network, we construct an extensive set of random candidate regions, simulate pre-touch measurements within these regions, and select those that demonstrate resilience against random noise and misalignments when correlated with the Kinect cloud's corresponding region.

Region Specification

The use of rotated rectangles provides a versatile yet straightforward method for defining regions. Jiang et al. [74] were pioneers in endorsing this modality for robot grasping, an approach that subsequent studies have embraced [90][137][146]. In our research, a delineated rectangle signifies a region designated for pre-touch sensing. We specifically conduct the pre-touch scan around the region's boundary. The parameters for each rectangle are given by:

$$[x, y, w, h, \theta]$$

Here, x and y represent the rectangle's central coordinates in pixels, w and h denote its width and height, and θ signifies the rotation around the rectangle's center in the image plane. Building on [146], we conventionally employ the cosine and sine of twice the angle to depict rotation, attributed to the rectangle's inherent symmetry.

Neural Network Architecture

As indicated in [146], opting for rectangles to designate areas of interest essentially morphs the challenge into an object detection task, albeit with the inclusion of a rotation parameter. This transformation permits us to harness the potent tools previously formulated in that domain. A prevalent detection method involves classifying image sub-patches and subsequently identifying the most confident positive classifications as detections. A straightforward tactic

would be to individually classify every sub-patch of an image, but this proves to be exceedingly time-consuming. To counter this inefficiency, Lenz et al. [90] employ a compact neural network to sift through a vast array of candidates, only forwarding the most promising ones to a more extensive network. In contrast, our approach utilizes the Faster R-CNN architecture [147], which relies on a Region Proposal sub-network to guide the detection layers towards sub-patches most likely to enclose a region of interest. Although the initial Faster R-CNN version [50] was crafted for object detection, we have tailored it to predict rotated rectangles. Achieving this necessitated a mechanism for efficiently calculating the overlap between two such rectangles, a feat accomplished using the General Polygon Clipper library [124]. The convolutional layers of our neural model are initialized using the 'CNN_M' pre-trained architecture from [21]. When provided with a depth image, our network returns a series of detected regions, each complemented by a confidence score.

Data Collection

Generating samples of regions suitable for pre-touch sensing is intricate. The objective is to pinpoint regions that, when subjected to a pre-touch sensor, produce data that aligns consistently with the corresponding sections of the original Kinect cloud, even in the face of random noise and calibration discrepancies. While a human could potentially identify some of these regions, the human intuition may not always align with what is computationally optimal. Moreover, automating data collection can facilitate scalability, ensuring ample data to combat overfitting. Thus, we generate a plethora of random candidate regions and curate labels based on regions that offer the most stable ICP matching outcomes. The assessment of numerous candidates is conducted by simulating the pre-touch measurements of each region.

The procedure for label generation is delineated in Algorithm 1. We initiate with a candidate set comprising one thousand rectangles ($n_R = 1000$). Each rectangle is conditioned to have an intersection-over-union (IoU) of no less than fifteen percent with the object's bounding rectangle. Additionally, their area must range between ten to fifty percent of the bounding rectangle's area. The algorithm, in its loop as stated in line 5, constructs a target

cloud for every rectangle. This is achieved by retaining those points from K that, when projected onto the image plane, are proximate to, or lie on the rectangle's perimeter. The algorithm subsequently engenders ten random offsets ($n_{trials} = 10$). These offsets in x, y, and z translations range between -2.0 cm and 2.0 cm, while the roll, pitch, and yaw rotations span from -5.0 to 5.0.

For each of these offsets, the loop on line 10 modifies K to generate K'. This simulates the data a pre-touch sensor might capture when engaged with the entire object. Subsequently, the source cloud for every rectangle is derived by keeping those points from K' that, upon being projected onto the image plane, lie on the rectangle's perimeter. This emulates a pre-touch scan (sans noise) along the region's perimeter corresponding to the rectangle. Before utilizing ICP to match each source-target pair, gaussian noise ($\sigma = 0.15$cm) is added to the source. A score is allotted to the pair based on the average distance between the projected aligned cloud's points and the ground truth aligned cloud's points. The final score allocated to each rectangle (calculated in line 19) is the maximum score recorded across all trials for that rectangle. This scoring system ensures that only rectangles that consistently perform well across various simulations earn commendable scores. The rectangle with the most favorable score is retained as a label.

3.5.3 Experiments

To evaluate the proposed framework, we apply it to the pose estimation of eight objects that were not included in the dataset used to train the deep neural network. For each object, the PR2 robot acquires an initial point cloud from the Kinect. This is then utilized by our detection network to propose a set of pre-touch scanning regions. For each proposed region, the robot conducts a scan along the path corresponding to the rectangle's perimeter. After every scan with the pre-touch sensor, we match it with the appropriate region of the Kinect cloud. This matching region is determined in a manner akin to lines 6-8 of Algorithm 1. To obtain a ground-truth estimate of the pose, we match a separate pre-touch scan of the entire object with the Kinect cloud. We contend that this approach is valid because the data

Algorithm 1: Label Generation

Data: Kinect cloud K

Result: Filtered Rectangles

$R \leftarrow n_R$ random rectangles

$K2D \leftarrow$ projection of K into image plane

$T \leftarrow n_R$ empty target clouds

for $r \in 1$ **to** n_R **do**

  for $i \in 1$ **to** n_{K2D} **do**

    if $R^r.nearPerimeter(K2D_i)$ **then**

      T^r.addPoint(K_i)

$\Delta K \leftarrow n_{trials}$ random translations & rotations

for $j \in 1$ **to** n_{trials} **do**

  $K' = K + \Delta K^j$ ▷ `Offset cloud` K'

  $K'2D \leftarrow$ projection of K' into image plane

  $S \leftarrow n_R$ empty source clouds

  for $r \in 1$ **to** n_R **do**

    for $i \in 1$ **to** $n_{K'2D}$ **do**

      if $R^r.onPerimeter(K'2D_i)$ **then**

        S^r.addPoint(K'_i)

    $trialScores^{j,r} = \text{Score}(S^r + N(0, \sigma), T^r)$

for $r \in 1$ **to** n_R **do**

  $scores^r = \max_j(trialScores^{j,r})$

return FilterRecs($R, scores$)

Function *Score(P, Q)*

  $PAlign = icp(P, Q)$

  return $\frac{\sum |PAlign_i - PAlign_i^*|_2}{n_P}$

	Mean Error (cm)			Std. Dev. of Error (cm)		
	Random	NARF	Deep Pre-touch	Random	NARF	Deep Pre-touch
Air Freshener	1.31	2.21	**1.26**	0.52	1.50	**0.50**
Clock	2.21	2.97	**0.80**	1.40	2.16	**0.32**
Cleanser	1.93	**0.71**	1.16	1.46	**0.58**	0.60
Controller	2.77	2.91	**1.49**	1.33	1.96	**0.76**
Fruit Bowl	1.69	1.87	**0.83**	1.50	1.46	**0.29**
Gripper	2.24	1.48	**1.11**	1.39	0.95	**0.41**
Toy	2.55	1.94	**1.72**	1.13	0.66	**0.43**
Wallet	1.16	1.70	**0.89**	**0.63**	0.83	0.66

Table 3.4: Mean and standard deviation pose estimate error across ten individual scans for each of the eight different objects and each of the three methods. All values have centimeter units, and the best value across each of the three methods is bolded.

from the pre-touch sensor remains unaffected by potential mis-calibrations in the robot arm's coordinate frames. Additionally, as observed in Fig. **??**, data from the pre-touch sensor more accurately reflects reality compared to that from the Kinect. For simplicity, all scans were conducted with the sensing beam perpendicular to the supporting table. However, future systems could explore more intricate scanning strategies.

Individual Scan Comparison

Here's a revised version of the provided section:

We benchmarked our region proposal network against two baseline proposal techniques. The first baseline proposes random rectangles that maintain properties analogous to those employed in the label generation process, as delineated in Section 3.5.2. The second baseline, a variant of Normal Aligned Radial Features (NARF) [162], recommends regions characterized by significant changes in the object's surface. Initially, this method computes standard

NARF keypoints and their respective descriptors. For each keypoint, the associated rectangle is demarcated by two perpendicular vectors originating from the keypoint. The primary vector's direction aligns with the descriptor element that holds the maximum value, while the secondary vector's direction is determined by which of the two elements, perpendicular to the primary, exhibits a higher descriptor value. The vectors' magnitude is adjusted such that the rectangle's area equals half of the object-bounding rectangle. The rectangle's confidence score is the aggregate of the two descriptor elements corresponding to the selected directions.

For all three methodologies, we amassed scans corresponding to the initial ten proposed rectangles for each object. We then individually aligned them with relevant portions of the Kinect cloud, as outlined earlier, and computed the mean distance between aligned point pairs in the estimated cloud and the ground truth cloud. Table 3.4 displays the mean and standard deviation of these pose estimation errors. The standard deviations

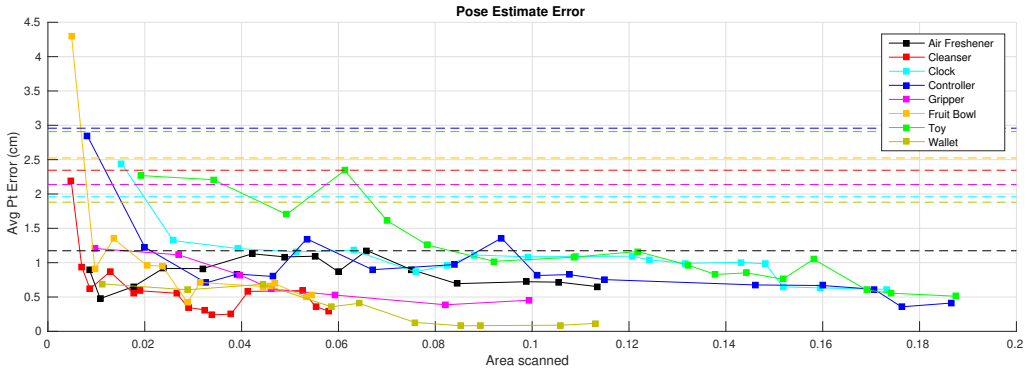

Figure 3.8: The pose estimate error after each pre-touch scan. The y-axis is the average distance between the ground truth alignment's points and the estimated alignment's points, and the x-axis denotes the percentage of the object that has been scanned. Each object corresponds to a different color, and each square represents a scan. The correspondingly colored dashed lines for each object represent the average distance between the points of the entire ground truth alignment and the points of the whole original Kinect point cloud.

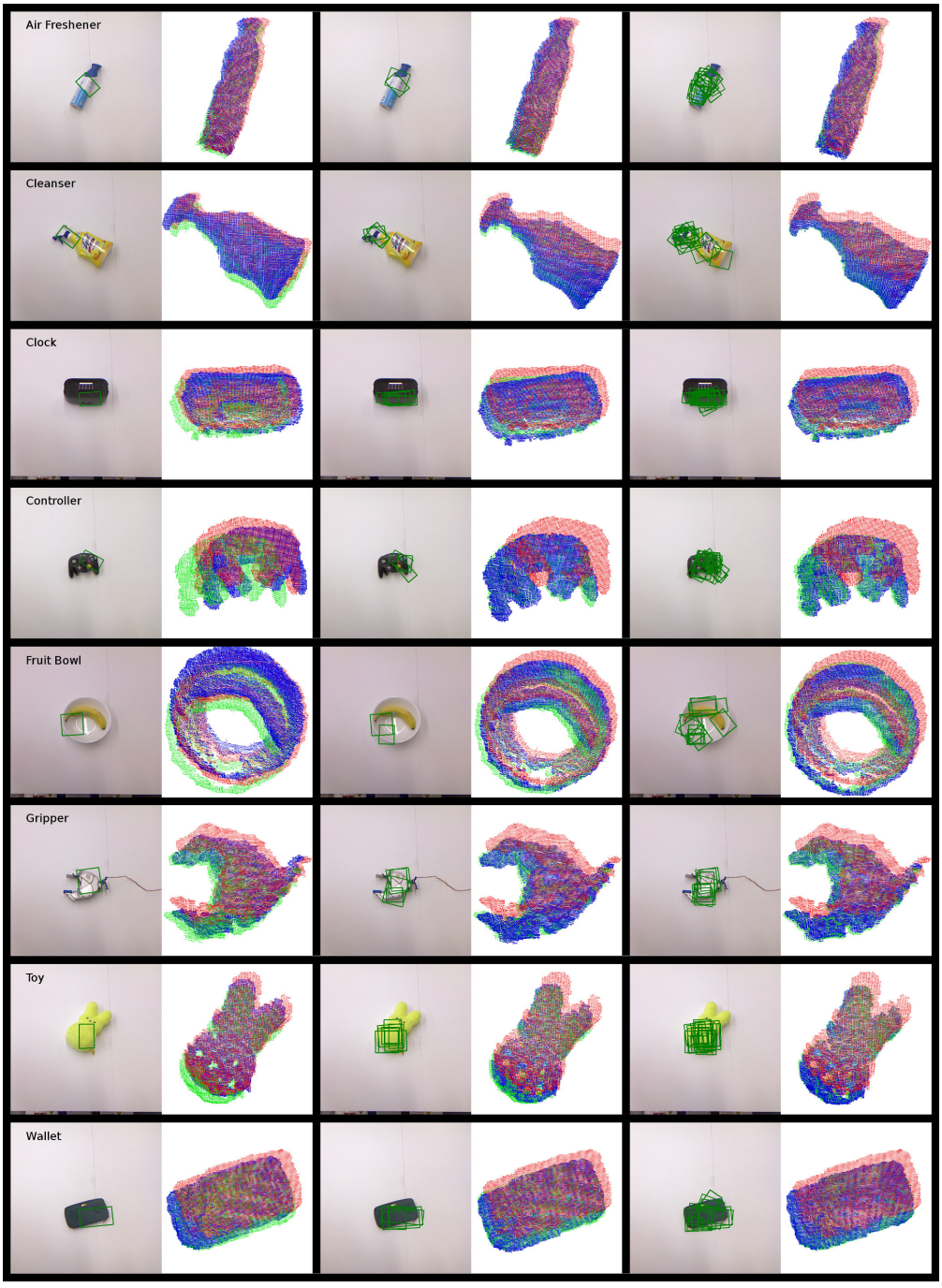

Figure 3.9: The matching results throughout the sequence of regions scanned by the pre-touch sensor for each object. For each pair, the left image indicates the regions to be scanned with green rectangles, while the right image displays the result of performing the scans. Three clouds are shown. Red represents the original Kinect data, blue represents the alignment estimated using the scans up to that point, and green represents the ground-truth alignment. The left-most pair corresponds to a single scan, the middle pair corresponds to a few scans, and the right-most pair corresponds to the point at which further scans provided

serve as an indicator of each method's reliability, with a smaller value signifying greater consistency. Barring one item, our region detection network consistently demonstrated the lowest mean—often by a considerable margin. Moreover, our network exhibited the lowest standard deviation for the majority of items, though it was marginally surpassed by the other methods for two items.

Deep Pre-touch Sequential Scanning

The previous section illustrated the efficacy of individual scans proposed by the region detection network. However, in scenarios demanding a more refined margin of error, integrating multiple scans might offer enhanced precision. We posit that superior pose estimation can be realized by merging successive scans. To achieve this, we arrange the regions for scanning based on their respective confidence scores. Subsequent to each scan with the pre-touch sensor, we amalgamate it with the prior ones before aligning it with the associated segments of the Kinect cloud.

Fig. 3.8 and Fig. 3.9 present the pose estimation outcomes for every object. As discerned in Fig. 3.8, the pose estimation generally becomes more accurate as the scanned area enlarges. This figure also reveals that a majority of the objects attained a localization precision of roughly 0.5 cm after scanning under a fifth of the total object area. Certain objects, like the gripper, wallet, and cleanser, achieved this precision with less than a tenth of the object scanned. Notably, the pose estimation for the majority of the objects surpasses the Kinect-based estimate after just a singular scan. Fig. 3.9 showcases the sequence of region detections and pose estimations for each object. The regions suggested by the network appear intuitively sound. A notable observation is the cleanser's first proposed region by the network, which is the distinct dispersal end, highlighting the geometric uniqueness of the object. Furthermore, Fig. 3.9 elucidates the robot's strategic flexibility in allocating scanning time. If constrained by actuation time, a solitary scan might suffice to enhance the pose estimate. Conversely, with more time on hand, the robot could potentially realize a precise pose using only a handful of scans. However, in cases where utmost accuracy is non-

negotiable, the robot could opt for multiple meticulous scans, circumventing a comprehensive scan of the entire object.

3.6 Discussion

Drawing from the insights of the three presented research endeavors, this chapter has elucidated several advancements and methodologies related to robotic manipulation. Firstly, the establishment of a versatile benchmark offers a means to assess precise, sequential robot manipulations across a gamut of platforms. While this benchmark is centered on evaluating Rubik's cube manipulations, it concedes the challenge of quantifying individual manipulative actions and the hurdles of obtaining unobtrusive ground-truth data. The benchmark, nonetheless, carves out promising avenues for its application, especially in scrutinizing visual servoing methods that navigate occlusions during manipulation [109, 96, 157].

Secondly, the chapter underscored the significance of pre-touch scanning as an enabling mechanism for robots to undertake sequential manipulations. A novel optical time-of-flight pre-touch sensor has been crafted from affordable components, enhancing the robot's capacity to accurately decipher the Rubik's cube's pose. This technique's extension to various objects reinforces the assertion that even rudimentary scans can accrue substantial geometric information requisite for competent pose estimation.

Lastly, we converged the realms of deep learning and pre-touch sensing to augment object pose estimation, achieving noteworthy results. Our deep learning model, trained on an unbiased dataset apt for any pre-touch sensor, has displayed superiority in estimating object poses, often refining the pose estimation with minimal scans. This has sparked an interest in enhancing the model's accuracy and incorporating it into comprehensive manipulation systems, potentially exploring the potential of memory-centric models like recurrent neural networks.

Collectively, these revelations lay the groundwork for a more dexterous and intelligent robotic future, where manipulation is nuanced, informed, and adept. Whether through benchmarking, sensor innovation, or leveraging deep learning, each research thread con-

tributes uniquely to this vision, signaling exciting trajectories ahead in robotic research.

The path forward in our research on robotic manipulation is paved with diverse and exciting prospects that aim to expand upon the findings presented in this chapter. A primary direction to explore is planning under uncertainty. Algorithms designed to judiciously decide when uncertainty reduction is imperative can potentially outpace the efficiency of our current fingertip sensor-aided baseline [130, 15, 171]. Moreover, our benchmark stands as a promising tool to evaluate the robustness and agility of robots endowed with high in-hand dexterity achieved through techniques like external force, optimal control, and deep reinforcement learning [103, 128, 28, 86, 2].

Building on the foundation of pre-touch scanning, the chapter prompts the research community to delve deeper into how robots can optimize scanning trajectories for maximum geometric information retrieval and how electric field sensors can be employed for electric field imagery. An interesting corollary to this would be the development of robust metrics that evaluate the quality and efficacy of scans. The very notion of combining various modalities of pre-touch scanning into a cohesive system for robotic manipulation is an enticing avenue worth investigating.

Lastly, in the realm of deep learning, there is a promising horizon in refining the accuracy of our methods, integrating them into expansive manipulation systems, and exploring models like recurrent neural networks that may offer better performance in predicting well-suited regions for pre-touch scanning. A deeper analysis of the features our model learns, and juxtaposing them with features from other robotic tasks like object detection, provides fertile ground for further studies.

In essence, the future beckons a multidimensional approach to robotic manipulation, leveraging the best of planning algorithms, sensor technology, and deep learning to create machines that are more adept, intelligent, and dexterous.

Chapter 4

COMPETITIVE GAME FOR HUMAN-ROBOT INTERACTION

Our second project investigates whether a robot can partake in physically competitive games against human players, focusing on the Fencing Game, a human-robot competition. The goal is to boost research in competitive human-robot interaction by developing a robot that challenges humans, particularly in physical exercise and games. We employ iterative multi-agent reinforcement learning to train the robot, demonstrating its competitiveness against human players.

Although competition is ubiquitous in nature and human society, it's been relatively unexplored in Human-Robot Interaction (HRI), which typically focuses on cooperative interactions [24, 70, 32]. We examine the benefits of competitive HRI in physical exercise scenarios—like athletic practice, fitness training, and physical therapy—where competition can foster adherence often limited by motivation deficits [113, 140, 172]. Given that enjoyment, competition, and challenges are key motivators [49, 140], we aim to design a robot that embodies these through adversarial behaviors.

Our key contributions include: 1. **Motivating Competitive-HRI Research:** We discuss how competitive interaction can positively impact individuals and the challenges posed by competitive-HRI tasks. This leads to our proposal of the Fencing Game, a physically interactive competition for competitive-HRI algorithms and user studies. 2. **System Design and Implementation:** We create a highly adaptable robotic system using a multi-agent RL method to train a robot in competitive games. 3. **Two User Studies:** Over 80% of participants found our robot enjoyable and engaging. It provided challenging gameplay that increased players' heart rates. Participants who explored different strategies achieved higher rewards. A subsequent study showed an RL-trained policy made quantitative improvement

harder for participants compared to a heuristic baseline policy. The RL-trained agent was perceived as more intelligent.

Figure 4.1: Competitive fencing games between a PR2 robot and human subjects. The detailed game rules are described in Sec. 4.1. Please refer to this link for example gameplay videos.

4.1 Competitive Interaction

In this section, we delve into the importance of competitive interaction and advocate for its heightened focus within the robotics community. First, we'll explore the psychological aspects and understand how competition can positively impact individuals. Subsequently, we'll address the technical challenges associated with competitive-HRI tasks and introduce the Fencing Game as the central task for this project.

4.1.1 Positive Influences of Competitive Interaction

Competition among players can be a motivating factor, often leading to enhanced performance in a specific task. Plass et al. [139] studied competitive versus cooperative interactions within an educational mathematics video game. The findings indicated that participants performed significantly better in a competitive setting than when working individually. Specifically, competitive players showcased superior problem-solving skills compared to their non-competitive counterparts. This research also found that participants derived more enjoyment from the game, evidenced by their increased engagement [151]. Moreover, they

exhibited higher situational interest, *i.e.*, they were more attentive and interactive during the game [63]. Viru et al. [174] revealed that competitive exercises could enhance athletic performance, as seen in a treadmill running test where participants' average running duration improved by 4.2

Building upon these studies, we believe that a personal robot can serve as a competitive partner, augmenting enjoyment, boosting motivation, and fostering improvement in tasks, especially in physical exercises. Thus, our competitive-HRI research begins with the development of a physical exercise companion.

4.1.2 Technical Challenges

Designing a robot capable of physically competing with a human presents significant challenges. The robot must continuously interpret the human's intentions through their actions and strategically maneuver its high degree-of-freedom body to counter the adversary's tactics and optimize its performance. Consequently, a substantial portion of our competitive-HRI research is dedicated to addressing these technical hurdles, aiming to develop a robotic system with real-time decision-making skills and agility comparable to that of humans.

4.1.3 The Fencing Game

Given the anticipated technical challenges, we devised a two-player zero-sum physically interactive game known as "The Fencing Game." This attack and defense game pits a human player, the antagonist, against a robot, the protagonist. The antagonist seeks to maximize their score, while the robot aims to minimize it. Figure.4.1 depicts human participants engaged in the game. Additionally, Algorithm.2 details the game's scoring mechanism.

Central to the game is an orange spherical target area positioned between the two players. The antagonist, on the right, accumulates 1 point every 0.01 seconds their bat remains inside the target zone without touching the robot's bat. Conversely, the antagonist loses 10 points if their bat, while inside the target zone, contacts the robot's bat. Furthermore, the antagonist is awarded 10 points if the robot's bat lingers inside the target area for over 2

seconds, signaling the robot's defensive stance. Each round of the game lasts 20 seconds. The observation metrics for both players encompass the Cartesian positions and velocities of both bats and the elapsed game time. Although both agents in this iteration are stationary for simplicity, future versions can seamlessly incorporate mobility.

Algo. 2 summarizes the scoring mechanism for the Fencing Game described in Sec. 4.1 [187].

Algorithm 2: The Fencing Game Scoring Mechanism

Initialize: Game score $s = 0$; Timestep = 0.01 Sec; Game horizon = 20 Sec

bat_a $\rightarrow$ Antagonist's bat

bat_p $\rightarrow$ Protagonist's bat

target $\rightarrow$ Target Area

for *every timestep in this game* **do**

  if *bat_a in target* **then**

    if *bat_a contacts bat_p* **then**

      s -= 10

    else

      s += 1

    end

  if *bat_p in target for more than 200 consecutive timesteps* **then**

    s += 10

end

4.2 Modeling Competitive-HRI as Games

In this research, we focus on 2-player zero-sum game scenarios where a single human user competitively interacts with a robot. To formalize the representation of the subset of competitive-HRI problems of interest, we frame them as multi-agent Markov games [98]. A Markov game between a human and a robot is described as a partially observable Markov decision process, defined by the tuple $(\mathcal{S}, \mathcal{O}^h, \mathcal{O}^r, \mathcal{A}^h, \mathcal{A}^r, \mathcal{T}, \mathcal{R})$. Here, $\mathcal{S}$ represents the set of states that describe the game's state. $\mathcal{O}^h$, $\mathcal{O}^r$, $\mathcal{A}^h$, and $\mathcal{A}^r$ denote the sets of observations and actions for the human and robot, respectively. The transition function $\mathcal{T} : S \times \mathcal{A}^h \times \mathcal{A}^r \rightarrow \mathcal{S}$

maps the current state and actions to a subsequent state. In the zero-sum game setting, if r_t^r and r_t^h are the instantaneous rewards at time t for the robot and human respectively, then $r_t^h = -r_t^r$. The human player aims to maximize their long-term reward $\mathcal{R}$ over a finite time horizon T, defined as $\mathcal{R} = \sum_{t=0}^{t=T-1} \gamma^t r_t^h$, while the robot seeks to minimize this reward.

RL Algorithms for Markov Games

In the context of multi-agent Markov Games, this paper delves into existing solutions and assesses their relevance to competitive-HRI. The task of constructing a transition model for human players using real human demonstration data is challenging. Such models are resource-intensive to create, and often, they aren't easily transferable across tasks. As model-based or supervised learning approaches necessitate a comprehensive state-action transition model, they aren't optimal for training the robotic player. Consequently, our focus will shift to exploring reinforcement learning (RL) methods that don't demand extensive real-world datasets.

The realm of Multi-agent Markov games is richly explored. With games like Chess, Checkers, and Go offering an interactive milieu replete with a definitive scoring mechanism, these environments have been instrumental for researchers to evaluate the efficacy of algorithms in training an agent for reasoning, learning, and planning [153, 18, 159]. Predominantly, algorithms addressing multi-agent Markov games are structured around a multi-agent reinforcement learning paradigm. Herein, agents exhibit emergent, intricate behaviors by interacting and co-evolving. Hillis [65] pioneered experiments with competitive co-evolution, while Stanley and Miikkulainen [161] dabbled with evolutionary strategies within a rudimentary 2D competitive landscape. Additionally, He et al. [59] employed a deep Q neural network to encapsulate both the transition function and the adversary's policy in competitive games. On another note, Tampuu et al. [166] leveraged deep-Q learning for training agents to navigate the Pong game, in both its cooperative and competitive versions. Remarkably, Silver et al. [160] surpassed human prowess in Go, Chess, and Shogi using reinforcement learning via competitive self-play. The LOLA [44] and LOLA-DiCE [43] algorithms employ

policy gradients to update policies, drawing insights from the parameter space of both the agent in training and its adversaries. These algorithms notably elevate learning stability in competitive settings. Yet, they presuppose that every agent's parameter space dimensions are congruent, and all agents are privy to the parameters of their counterparts, which curtails their generalizability.

Despite the commendable performance of the above methods in specific situations, their trials predominantly span board games, video games, 2D particles, and analogous environments. Such domains often possess a more straightforward state transition mechanism compared to the intricate competitive-HRI scenarios we're investigating. The kinematic and dynamic intricacies of robots and human bodies result in profoundly non-linear transition functions. Moreover, steering a robot mandates mastering intricate motor skills in the face of multifaceted dynamics. Recent works have made strides in harnessing game RL to imbibe such skills. Pinto et al. [138] introduced an adversarial training technique adaptable to many existing RL methods, relying on a straightforward iterative training schema, though it sometimes grapples with convergence and stability issues inherent to multi-agent learning [118, 116]. Using PPO and substantial training batch sizes, Bansal et al. [5] spawned humanoid and quadruped agents adept at simulated physical competitions like soccer and wrestling. Lowe et al. [100] proposed a centralized action-value function that absorbs the actions of all agents to stabilize the DDPG algorithm [94] in multi-agent contexts. These methods, unified by their neural network-backed modeling of environmental intricacies and policy updating rooted in the actor-critic framework, will be critiqued in our paper. Specifically, we will juxtapose two methods within our competitive-HRI setting. However, it's pivotal to acknowledge that a conventional game representation doesn't encapsulate the challenges unique to embodied agents, a subject we will broach in the succeeding section.

4.3 System Design and Implementation

Humans excel at recognizing patterns and acquiring skills from a limited set of examples [88]. Recent studies also demonstrate that human participants can rapidly adapt to robots, en-

hancing their performance in a handful of trials during physical HRI tasks [77, 127]. However, games against robotic opponents become less challenging if human players can readily anticipate the robot's actions and swiftly devise a counter-strategy. Consequently, we theorized that while human players can swiftly optimize their performance against a consistent robot policy, altering the robot's gameplay style could disrupt this learning, maintaining the game's challenge. A gameplay style is defined by patterns observed in the agent's end-effector motion trajectories. For instance, one antagonist might favor pronounced stabbing motions, while another could lean towards slashing movements.

Thus, two primary objectives guide our robot control policy creation. The first objective ensures the robot can competently engage in the Fencing Game, making the experience intense and captivating for human participants. The second aims to derive three distinct gameplay styles for our user study. We employed a multi-agent reinforcement method to formulate the robot control policies used in the study. The physical system, based on the PR2 robot, was designed and realized by us.

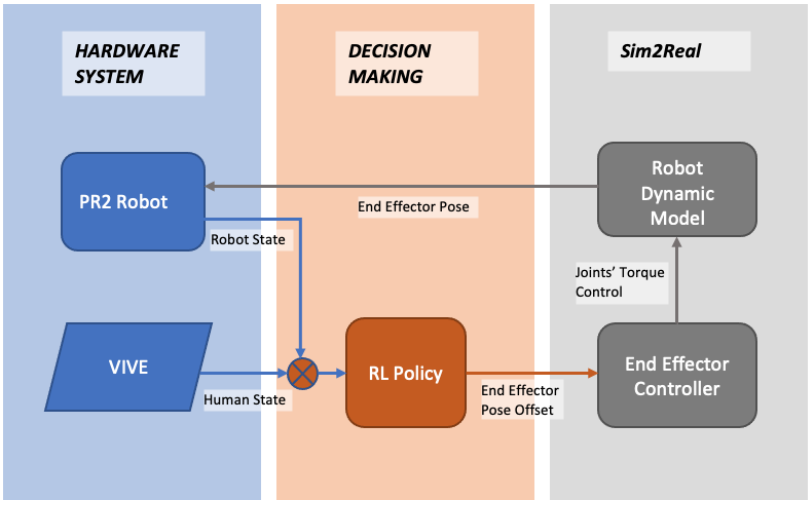

Figure 4.2: A block diagram demonstrating the pipeline of the proposed robotic system. Human motion tracking is achieved via a HTC VIVE VR system.

4.3.1 Hardware Details Details

The PR2 robot is a widely recognized general-purpose robotic platform featuring two 7-degree-of-freedom(DoF) arms. Its overall form-factor is akin to that of an adult human, making it referenced in numerous studies [185, 184, 126]. Due to its body size and arm flexibility, it's on par with a human player in competitive games. The human player's bat is equipped with an infrared photo-diode array tracker, and the robot senses its position and orientation using two tracking base stations. We've also implemented an audible scoring feedback system to provide real-time scoring updates during each game. Participants hear a high-frequency sound (440 Hz) when they score and a lower frequency (300 Hz) sound when penalized by the robot. The detailed system implementation process can be viewed in Fig. 4.2.

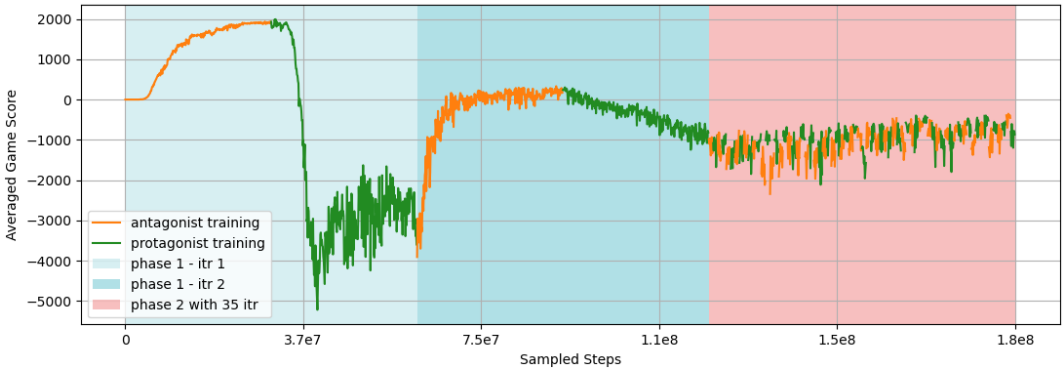

Figure 4.3: The antagonist's average game score during one complete round of phase one and two training. The two iterations of phase one training enabled two agents to interact according to the game rules. The phase two training performed 35 small updates resulting in random characteristics for both agents.

4.3.2 Learning to Compete

To craft robot control policies that fulfill the previously described objectives, we model the Fencing Game as a multi-agent Markov game problem [98]. In the Mujoco simulation environment, both the antagonist and protagonist agents are represented by a PR2 robot model. They undergo training through a co-evolutionary process, competing against each other. We streamline the multi-agent proximal policy optimization method, as proposed by Bansal et al. [5], into two distinct training procedures. This refinement reduces the computational demands, enabling the efficient generation of multiple agent pairs with satisfactory performance. Consequently, all sampling and training processes could be executed on a standard desktop computer (CPU: $1 \times i7$; GPU: $1 \times GTX970$). We introduce our tailored version of the multi-agent PPO, termed the two-phase iterative co-evolution algorithm, in Algo. 3.

Learning to Move and Play

The initial phase of training functions as a pre-training process, focusing on equipping both agents with the motor skills essential for joint control and familiarizing them with the game's rules. During each timestep, agents receive rewards based on a combination of a continuous reward and the game score. The continuous reward promotes exploration within the task space.

The antagonist's policy, denoted as μ with parameters θ_i^{μ}, undergoes its first round of training by gathering trajectories resulting from matches against the protagonist using its latest policy. This procedure persists until either a timeout occurs or μ stabilizes. Subsequently, the protagonist's policy, ν, characterized by the parameter θ_i^{ν}, is trained by competing against the antagonist's most recent policy. Remarkably, by executing this training sequence merely twice ($N_{iter} = 2$), we secured robot policies demonstrating proficient, albeit not flawless, gameplay. The policies derived from this inaugural phase are termed the warm-start policies.

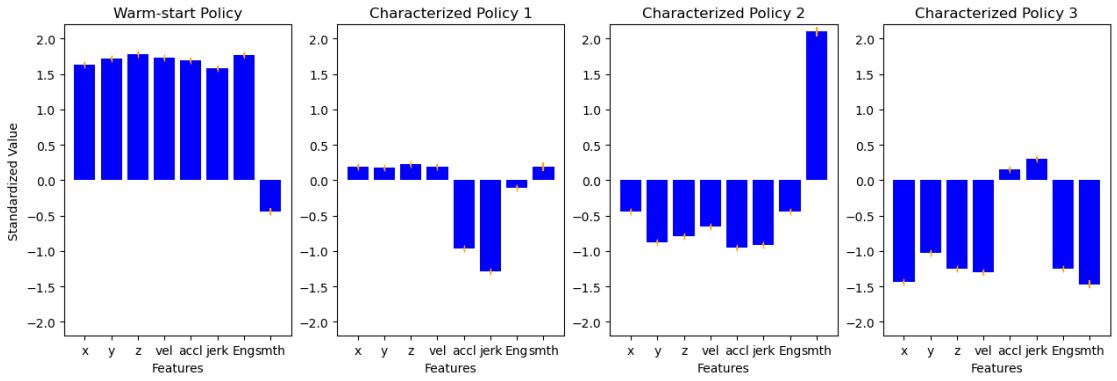

Figure 4.4: Visualization of quantified gameplay style of the four policies used in the user study. Error bars indicate the standard deviation of the feature values among the population.

Creating Characterized Policies

Research has demonstrated that the use of varying random seeds to steer policy optimization can lead agents to explore different behaviors and strategies for the same task [61, 160]. The inherent variability of multi-agent systems amplifies this stochastic effect. Within these systems, agents tend to develop highly diverse strategies and exhibit emergent behaviors when continuously competing against each other [5, 4, 99, 111].

The second training phase seeks to produce a policy with random characteristics by leveraging this phenomenon. Both agents commence with their warm-start policies from the first phase and undergo training as detailed in Algo. 3. During this phase, rewards are solely based on game scores. A departure from the first phase is evident when training each agent: rather than competing against the opponent's latest policy, a random past version of the opponent's policy is selected for use in this phase.

As a result, we crafted a policy library comprising six pairs of policies, each with random characteristics. These emerged from six distinct iterations of the second-phase training. Fig. 4.3 captures the evolution of game scores throughout the bi-phase training journey.

Additional Algorithmic and Training Details

In Algo. 3, phase one and phase two training are distinguished by two significant differences. **(1)** During the first phase, agents utilize a continuous reward mechanism, aiding the development of rudimentary motor skills. In contrast, phase two centers rewards solely around game scores. **(2)** In the initial phase, agents consistently train against the opponent's most recent version. However, in the second phase, agents intermittently and unpredictably switch to training against an earlier version of the opponent from their history.

The continuous reward in phase one accelerates the robot's exploration of the task space, making this phase effective for initializing antagonist and protagonist policies swiftly. Yet, persisting with this iterative strategy could confine both agents to an inferior local equilibrium or lead them on an endless loop within the parameter space during extended training [118]. Phase two's reward and iteration dynamics, on the other hand, introduce a higher variance in the learning environment, countering the potential looping issue. Moreover, the variability inherent in phase two bolsters the chances of agents adopting emergent behaviors and refining their policies.

The data in Fig. 4.3 elucidates the learning progression of the agents during their training phases. In the inception of their training journey, specifically during phase 1 - itr 1, there's a clear bias in game scores towards the agent that's actively undergoing the learning process. The simplicity and naivety of both agents' policies during this embryonic stage make it rather uncomplicated for the opposing agent to discern and employ a counter strategy, granting it a significant advantage in the game. However, this dynamic changes post the warm-start training. By the culmination of phase 1 - itr 2, both agents have navigated to a region within their policy space where they effectively challenge one another, ensuring that neither gains an overwhelming upper hand in the game.

Algorithm 3: Iterative Co-evolution

Input: Environment $\mathcal{E}$; Stochastic policies μ and ν; Instantaneous Reward Function $r(\cdot)$

Initialize: Parameters θ_0^μ for μ and θ_0^ν for ν

for $i = 1, 2, ..N_{iter}$ **do**

  if *phase one training* **then**

    $\theta_i^\nu \leftarrow \theta_{i-1}^\nu$

  else

    $\theta_i^\nu \leftarrow \theta^\nu_{random\_from\_history}$

  end

  for $j = 1, 2, ..N_\mu$ **do**

    rollout $\leftarrow roll(\mathcal{E}, \mu_{\theta_i^\mu}, \nu_{\theta_{i-1}^\nu}, r(\cdot))$

    $\theta_i^\mu \leftarrow$ PPO_Update(rollout)

  end

  if *phase one training* **then**

    $\theta_i^\mu \leftarrow \theta_i^\mu$

  else

    $\theta_i^\mu \leftarrow \theta^\mu_{random\_from\_history}$

  end

  for $j = 1, 2, ..N_\nu$ **do**

    rollout $\leftarrow roll(\mathcal{E}, \mu_{\theta_i^\mu}, \nu_{\theta_i^\nu}, r(\cdot))$

    $\theta_i^\nu \leftarrow$ PPO_Update(rollout)

  end

end

4.3.3 Selecting Agents With Distinct Gameplay Styles

To pinpoint the most distinct protagonist policies from the library mentioned in the previous subsection, it is crucial to quantify and contrast each agent's gameplay style. We begin by organizing a tournament for all protagonist policies, as per [168], where each agent competes in 100 games against each of the six opponents.

We employ eight end effector trajectory features to encapsulate an agent's gameplay style: total displacement change on the x, y, and z axes; average velocity; average acceleration; average jerk; total kinetic energy; and trajectory smoothness. For each game participated in by the protagonists, these features are calculated. An agent's quantified style is derived from the mean of these features across all 600 games.

By leveraging their three most salient principal components—ensuring less than 2% information loss [181]—we contrast the features of all protagonists. Subsequently, the trio of policies deemed most distinguishable is chosen for the user study. The gameplay style of the warm-start policy, alongside the three distinct policies, is illustrated in Fig. 4.4.

4.3.4 Sim2Real

Due to the discrepancies in kinematics and dynamics between simulation and real-world settings, policies exclusively trained on simulated data may underperform in real scenarios. To address this issue, we employ a combined approach: a Jacobian Transpose end-effector controller coupled with a system identification (systemID) process. Rather than stipulating the torque values for individual joints, the policy indicates an offset from the current end-effector pose. This updated desired pose is subsequently executed by the end-effector controller. We utilize the CMA-ES algorithm to optimize an objective across the parameter space for both the controller and the robot model within the simulation.

$$(\theta_m^*, \theta_c^*) = \arg\min_{(\theta_m, \theta_c)} \sum_{t=0}^{T} (s_r^t - s_s^t)^2$$

In this context, θ_m denotes the parameters of the simulated robot model, which includes damping, armature, and friction loss. Meanwhile, θ_c symbolizes the proportional gains and derivative gains associated with the end-effector controller. T_r and T_s are trajectories derived from the real robotic system and the simulation, respectively, in response to an identical control sequence. The terms $s_r^t \in T_r$ and $s_s^t \in T_s$ represent the end-effector pose of the robot in reality and the simulation at a given time t, respectively. Consequently, any disparity in the end-effector dynamics between the simulated robot and the real-life PR2 robot is minimized. Furthermore, the controller output's maximum limit is set conservatively to

mitigate the risk of human injury.

4.4 Experiments and Analysis

In this work, we conducted two in-lab user studies. The first study broadly explored the concept of competitive-HRI within the context of a fencing game. We engaged sixteen participants, asking them to play five games with each of the four RL-trained policies derived from Sec. 4.3. We evaluated our system using participants' game scores, heart rates, arm movements, and their feedback based on a modified Technology Acceptance Model (TAM) [23]. The evaluation centered around three primary queries:

1. Do human users accept a competitive robot in the context of physical games and exercises?

2. Is our system capable of providing a challenging and intense gameplay experience?

3. Can the robot influence human learning by altering its gameplay style?

The second study juxtaposed characterized policy 1 against a meticulously crafted heuristic-based policy. The robot, leveraging its understanding of the game's rules, positioned its bat between the target area and the point on the human's bat nearest to that area. To infuse unpredictability, action noise was incorporated into the heuristic strategy. Ten participants played 10 games against each policy. The study compared the policies based on game scores, participants' TAM feedback, and their perceptions regarding difficulty, enjoyment, and the robot's perceived intelligence.

4.4.1 User Study One Result: A Broad Exploration

Our initial experiment revealed that participants greatly appreciated the incorporation of a competitive robot as an exercise companion. The majority found engaging in competitive games with our robot beneficial, enjoyable, appealing, and motivating. Our system succeeded

Table 4.1: Subjective Ratings of the Modified TAM Questions

	1-Strongly Disagree	2-Disagree	3-Neutral	4-Agree	5-Strongly Agree
Perceived Usefulness	0%	6.25%	25%	37.5%	31.25%
Perceived Ease of Use	0%	18.75%	18.75%	62.5%	0%
Attitude	0%	6.25%	37.5%	25%	31.25%
Intention to Use	0%	18.74%	18.75%	25%	37.5%
Perceived Enjoyment	0%	6.25%	6.25%	31.25%	56.25%
Desirability	0%	6.25%	12.5%	31.25%	50%
Increase Engagement	6.25%	6.25%	31.25%	18.75%	37.5%

in offering a challenging and intense interactive experience, notably elevating participants' heart rates. When facing off against our RL-trained robot, many participants found it challenging to markedly enhance their performance over time. However, a subset of participants who persistently experimented with diverse strategies ultimately achieved superior scores.

Demographics

For the initial human-subject studies, 16 participants (10 males and 6 females, with an average age of $M = 28.8$ years and a standard deviation of $SD = 5.56$) were recruited. Of these, nine participants reported engaging in more than three hours of physical exercise per week, while the remaining seven indicated they participated in less than three hours of exercise weekly. The most frequently reported forms of exercise among the participants were jogging, walking, cardio, and weight training. Out of these 16 participants, ten took part in the subsequent user study.

Experiment Procedure

Before the experiment, each participant was instructed to sit for 3 minutes and then walk for 1 minute to record two baseline average heart rate values. During the experiment, both the robot and the participant wielded a polystyrene bat to play the games. A safety line was marked on the floor, which all participants stood behind to avoid potential collisions. Since the robot remained stationary, participants were also asked to keep their feet firmly planted on the ground throughout the game. The target area was not directly visible to the participants. Instead, an auditory feedback system informed them when their bat was correctly positioned within the target area: a high-frequency sound indicated a score for the human player, while a low-frequency sound signified a penalty. Before the game commenced, participants were given 5 minutes to familiarize themselves with the target area and scoring mechanism using their bats. This was followed by two practice games with the robot to help participants become more acquainted with the system. During these practice games, the robot's movements were deliberately slowed, and no data was recorded.

User Study One: The experiment comprises four sections. In each section, a participant engages in five consecutive games with the robot, adhering to a fixed robot policy, and takes roughly a 30-second break between games. Each game lasts 20 seconds, during which the participant's heart rate is monitored. Due to the brevity of these games, our discussions in Sec. 4.4.1 use the peak heart rate as a representative statistic of the recorded data.

In the experiment's initial section, the robot employs the warm-start policy derived from the first phase of training. In subsequent sections, the robot switches to one of the three chosen characterized policies in a random sequence. This setup ensures participants first familiarize themselves with the robot's regular speed, and subsequently experience a new gameplay style in each section.

After concluding each section, participants provide feedback on the preceding five games by choosing one or more adjectives from the following list: 'Exciting', 'Joyful', 'Frustrating', 'Motivating', 'Amusing', 'Intimidating', 'Physically Demanding', 'Cognitively Demanding',

'Boring', and an option for 'Others' with space for elaboration.

Once all four sections are completed, participants respond to a concluding questionnaire evaluating their receptiveness towards the competitive robot and their subjective impressions of the games. Our questions, presented in Table. 4.2, modify the technology acceptance model (TAM) [23]. We incorporated two additional questions (specifically, DE and IE) to gauge participants' potential acceptance and preference for a future competitive robot companion, as well as to discern if such a robot could inspire them to exercise more regularly. Except for the open-ended query, all TAM questions utilized a 5-point scale, with 1 representing "Strongly Disagree," 3 signifying "Neutral," and 5 denoting "Strongly Agree".

Furthermore, after each gameplay section, participants are prompted to evaluate and juxtapose both the enjoyment and challenge levels of the games across the completed sections. They can rate one section as being equivalent, less, or more enjoyable and challenging than another. By the experiment's conclusion, this method offers participants the opportunity to rank the four sections based on their perceived enjoyment and difficulty.

Experiment Procedure for User Study Two: The experiment consists of two sections. In each section, participants are instructed to engage in 10 consecutive games using a consistent robot policy, with a roughly 15-second intermission between games. The sequence in which participants play against the baseline policy and the characterized RL policy is randomized.

Upon the conclusion of each section, participants are prompted to respond to the modified TAM questions. After completing the final section, they are also required to address short questions 2, 3, and 4, as detailed in Table. 4.2.

Baseline Heuristic Policy: We endeavored to design a robust baseline heuristic policy to foster an intense human-robot gameplay experience. Upon observing the environment, the robot adjusts its bat to be perpendicular to the human's bat, incorporating random angular offsets uniformly chosen from -25 to 25 degrees on the x, y, and z axes. To consistently ensure the robot maintains a competitive defensive stance, the policy directs the robot to position the center of its bat equidistant between the target area and the closest point on

the human's bat to that target area.

$$\bar{b_p} = \bar{tar} + (\bar{h_{close}} - \bar{tar}) \cdot uniform(0.5, 1)$$

$$\bar{h_{close}} = \bar{h_{low}} + ht \cdot (\bar{h_{up}} - \bar{h_{low}})$$

$$ht = \max\left(0, \min\left(1, (\bar{tar} - \bar{h_{low}}) \cdot (\bar{h_{up}} - \bar{h_{low}})/(2 \cdot L_{sword})\right)\right)$$

Let $\bar{b_p}$, $\bar{tar}$, $\bar{h_{up}}$, and $\bar{h_{low}}$ denote the positions of the robot's bat frame, the center of the target area, the upper end of the human's bat, and the lower end of the human's bat, respectively. The term $\bar{h_{close}}$ signifies the point on the human's bat closest to the center of the target area, while L_{sword} denotes the length of a bat. The function $uniform(0.5, 1)$ randomly determines the distance between the robot's bat and the human's bat. Moreover, there's a 50% probability that the robot will retain and execute the desired bat position from the previous time step, rather than updating to the most recent desired pose. Introducing these uncertainties adds randomness to the robot's behavior. This heuristic approach allows the robot to dominate the fencing game when its speed matches or exceeds that of its opponent. However, since human participants can move slightly faster than our PR2 robot, opportunities arise for them to devise counter-strategies.

User Acceptance

Table 4.1 presents the subjects' responses based on the technology acceptance model. The majority of the subjects, specifically 68.75%, believed that a competitive robot could enhance their physical exercise experience. Additionally, 87.5% found competitive human-robot interactive games enjoyable, while 81.25% envisioned a future where a competitive robot would be a sought-after exercise companion. Notably, the metrics for the intention to use (with 62.5% agreeing or strongly agreeing) and heightened engagement (56.25% agreeing or strongly agreeing) did not reach the overwhelming consensus evident in the areas of perceived enjoyment and desirability. A few subjects, who weren't particularly inclined to use our system, shed light on their reservations in the open-ended section. They mentioned their uncertainty about how competitive robots would fit into their standard exercises like

jogging or weight training. Conversely, a significant number of participants, amounting to 71% of those who exercise less than three hours a week, were in favor of a competitive robot companion, seeing it as a way to boost their engagement in physical activities. This suggests that our competitive robot might be particularly motivational for those less active. Future studies should delve into strategies for effectively integrating competitive-HRI into mainstream workouts.

Increased Heart Rate

Heart rate data was captured using a Polar OH1+ optical heart rate sensor. Fig. 4.5. b. contrasts the subjective descriptions across four gameplay section groups, each characterized by varying average human heart rate levels. For every section in the user study, the average peak heart rate over the five games within that section was computed. The heart rate level, denoted as l, for a section is determined by dividing the section's average peak heart rate by the user's walking baseline heart rate. This yields a percentage value that indicates how the average section heart rate compares to the baseline. The four groups — low, medium, high, and ultra-high heart rate — correspond to sections where $l \leq 100\%$, $100\% < l \leq 120\%$, $120\% < l \leq 140\%$, and $140\% < l$, respectively.

Playing against our competitive robot led to a noticeable increase in human subjects' heart rates. In over 99% of the games, subjects' peak heart rates exceeded their resting rates, and in 92.6% of the games, these peaks surpassed heart rates from their walking baseline. It's worth noting that participants were instructed to keep their feet stationary during the games, thus restricting their mobility. Yet, even with these constraints, in 29% of the games, subjects' peak heart rates were substantially elevated—ranging from 20% to 58%—above their walking baseline.

Our observations revealed that not only physical exertion but also cognitive effort and expressed emotions were linked to heart rate elevations. As illustrated in Figure 4.5. b, intervals with a higher average heart rate were often associated with periods when participants felt cognitively challenged or reported feeling motivated, frustrated, or intimidated by the

robot. In essence, engaging in competitive games with our robot posed significant cognitive demands and elicited strong emotional responses.

The Human Learning Effect

As detailed in Sec. 4.3, we aimed to determine if altering the robot's gameplay style would disrupt the human learning trajectory, ensuring sustained challenge throughout the entire experiment. We initially hypothesized that most human players could achieve substantial improvement within five games when up against a static robot policy. Contrary to our expectations, our user study data did not support this assumption. Fig. 4.5. c illustrates the average game scores, along with the standard deviation for all participants, over five consecutive games against each of the four robot policies. An analysis of variance (ANOVA) indicates no significant rise in performance across the five games for any given policy (Warm-start Policy: $p = 0.96$, Characterized Policy 1: $p = 0.74$, Characterized Policy 2: $p = 0.85$, Characterized Policy 3: $p = 0.43$). The red horizontal dashed line in each subplot of Fig. 4.5. c denotes the top average score achieved by a participant, acting as a proxy for the peak human performance. The collective performance of all participants consistently fell below—often significantly so—the peak human benchmarks, indicating ample room for further improvement. Our initial beliefs regarding human learning stemmed from results observed in noncompetitive HRI scenarios. However, our competitive framework introduces a more dynamic environment, elevating the learning challenge for human players. Given the absence of significant performance variations across the four policy segments (ANOVA $F = 1.51, p = 0.26$), our system effectively maintained its challenging aspect for participants throughout the entire experiment. Nevertheless, further research is warranted to delve deeper into how shifts in gameplay styles might influence human learning.

Human Performance

While we did not observe a significant learning effect across the entire group of participants, it was evident that certain individuals experimented with multiple strategies throughout the

experiment. This behavior can be viewed as a form of exploration within a learning context. Prompted by this observation, we decided to investigate the relationship between strategic exploration and performance. We first quantified this by measuring the variance in game scores and then by analyzing the characterized gameplay styles as outlined in Sec. 4.3. As depicted in Fig 4.5. a., a higher variance in scores is associated with better performance, both in terms of peak and average scores.

To further delve into this, we contrasted the variance in gameplay style between the top five performers (based on their highest scores) and the bottom five. For clarity, we limited our comparison to changes in end-effector displacement on the x, y, and z axes, along with the average velocity. Remarkably, the variance for each selected feature among top-performing participants was at least 29% greater than that of the lower-scoring individuals. This suggests that those with a higher score variance were more inclined to adopt a risk-taking approach by continually experimenting with different strategies, which ultimately led to more favorable outcomes over time.

An interesting avenue for future research could be to explore whether a robot can enhance human performance by either verbally or non-verbally prompting users to experiment with diverse strategies.

4.4.2 User Study Two Result: Baseline Comparison

This experiment compared an RL-trained policy (characterized policy 1) against a robust heuristic baseline policy over an extended gameplay sequence (10 games for each policy). The RL-trained policy demonstrated marginally superior game score performance compared to the baseline. Contrary to findings from the initial experiment, evidence of a human learning effect was apparent for both policies. Nevertheless, the RL-trained policy was notably more effective at hindering human subjects from progressing, even without varying its gameplay style throughout the study. While the responses to the TAM model remained largely consistent for both policies, participants perceived the RL policy as more astute due to its “defensive” and “varied” behavior.

Game Scores

When participants played against the baseline policy, their average game scores (**Baseline mean:** 383.5 vs. **RL mean:** 349.1), maximum game scores (**Baseline max:** 929.0 vs. **RL max:** 744.0), and minimum game scores (**Baseline min:** -291.0 vs. **RL min:** -320.0) were higher than those recorded against the RL policy. In this study, rather than alternating policies every five games, we extended the gameplay sequence to more thoroughly evaluate each policy. This approach enabled us to discern a positive correlation between game scores and the duration of gameplay experience against a particular policy. Linear regression analysis of the **Baseline** data (slope=30.0, coefficient=0.56, p-value=9.3e-10) revealed a steeper positive slope, a stronger correlation coefficient, and a more statistically significant p-value when compared to the **RL policy** (slope=15.7, coefficient=0.34, p-value=0.0005).

Subjective Responses

Subjects' responses to most of the modified TAM questions for the two policies are very similar as shown in Table 4.3. However, 70% of the population considered the **RL policy** to be more intelligent. In the responses for short questions 2, 3, and 4 in Table 4.2, the **Baseline** policy was described as "fast" by 2 subjects, "follows my movement" by 4 subjects, and has "repetitive/predictable" behavior by 4 subjects. Meanwhile, the **RL** policy was considered to be "defensive" by 5 subjects, "strategic" by 2 subjects, and to have "diverse behavior" by 2 subjects.

Perceived Ease of Use

Interestingly, while no significant performance improvement was observed either within or between sections, 62.5% of subjects believed that they could easily make progress against the robot. Moreover, some participants were driven by the ambition to outscore previous participants. This suggests that some might have been more focused on achieving what they perceived as a "high" score in a few games rather than consistently performing well across all

20 games. In a competitive game context, our understanding is still evolving regarding what it means for participants to feel that attaining a better score is achievable. In our study, it likely indicates that the majority viewed the robot adversary as a beatable opponent, given that only a small fraction described their gameplay experience as "Frustrating", "Intimidating", or "Boring". Future studies could delve deeper into how this perceived ease influences the level of effort participants put into competitive games.

Enjoyment and Difficulty

We discovered that while perceived enjoyment and difficulty remain fairly consistent as player experience grows, the amount of data used to train the associated policy indeed influences these perceptions. Fig. 4.6 provides a comparative view of the average rankings for both enjoyability and difficulty across various policies and experimental sections. Among the policies in use, we didn't detect any notable differences in the three characterized policies regarding enjoyability ($p = 0.40$) and difficulty ($p = 0.35$). Paired-T tests indicate that the warm-start policy is perceived as less enjoyable and challenging than the characterized policies ($p < 0.01$), with the sole exception being the comparison between the enjoyability of the warm-start policy and policy 2 ($p = 0.13$).

This trend also holds true temporally. The latter three sections, where characterized policies were used in a randomized order, showed no significant variance in either enjoyability ($p = 0.5$) or difficulty ($p = 0.42$). However, section one, dedicated solely to the warm-start policy, was rated as less enjoyable and less challenging than the subsequent sections ($p < 0.05$). Across all the sections, there exists a moderate positive correlation between perceived difficulty and enjoyment, evidenced by a coefficient of 0.6.

4.4.3 Discussion

This work spurred research in competitive Human-Robot Interaction (HRI) by outlining how competition can benefit people and detailing the technical challenges posed by competitive-HRI tasks. We discovered that human subjects are highly receptive to a competitive robot as

an exercise partner. Our competitive robot delivers engaging, challenging, and high-intensity gameplay experiences, significantly elevating participants' heart rates. Additionally, our findings revealed that our reinforcement learning trained policies were significantly more effective at suppressing the human learning effect, and were perceived as more intelligent by the human users. Importantly, these findings demonstrate that the skills a robot acquires from participating in competitive games are highly effective in real-world execution. The robot develops intelligent, robust strategies capable of countering unforeseen adversarial disturbances, thereby engaging human users in competitive games.

	Questions
Perceived Usefulness (PU)	Having a competitive robot companion would improve the quality of my physical exercise.
Perceived Ease of Use (PEOU)	Learning to earn higher score (make progress) in the games with a competitive robot would be easy for me.
Attitude (ATT)	Using a competitive robot exercise partner to improve my exercise quality is a good idea.
Intention to Use (ITU)	Assuming I had access to a competitive robot for exercise, I would intend to use it.
Perceived Enjoyment (PENJ)	I would find competitive human-robot gameplays are entertaining.
Desirability (DE)	Based on your experience today, future physical exercises and games with a competitive robot will be desirable.
Increased Engagement (IE)	Having a competitive robot companion would make me more likely to engage in physical exercise.
Short Question 1 (used in study one)	Are there anything you would like to change to improve the interaction experience?
Short Question 2 (used in study two)	Which robot (Section 1, Section 2, or equally) do you think is more challenging/difficult to play against? and why?
Short Question 3 (used in study two)	Which robot (Section 1, Section 2, or equally) do you think is more enjoyable/fun to play against? and why?
Short Question 4 (used in study two)	Which robot (Section 1, Section 2, or equally) do you think is more intelligent? and why?

Table 4.2: Modified Technology Acceptance Model and Open-ended Questions

	Baseline Median	RL Median	t-statistic	p-value
PU	3.6	3.6	0.0	1.0
PEOU	3.9	3.6	0.85	0.41
ATT	4.1	3.9	0.41	0.69
ITU	3.7	3.6	0.2	0.84
PENJ	3.9	3.9	0.0	1.0
DE	3.7	3.9	-0.45	0.65
IE	3.0	3.5	-0.9	0.37

Table 4.3: TAM Response Comparison Between Baseline Policy and RL Policy in Pair-T Tests

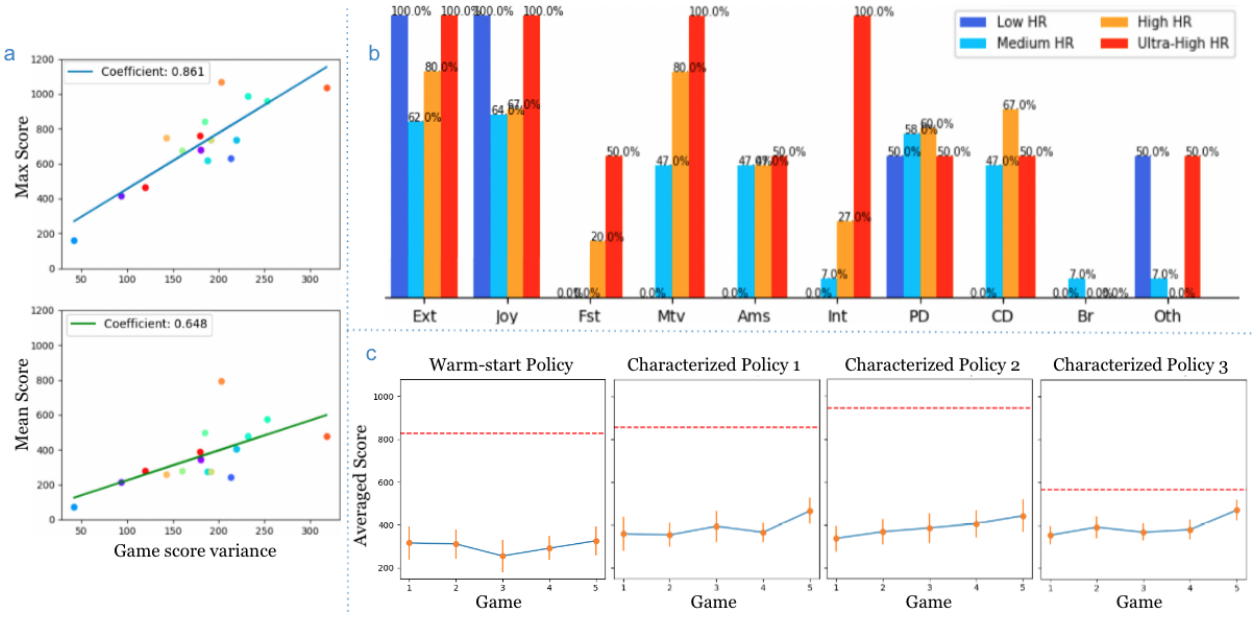

Figure 4.5: **a.** The 16 points in each subplot represent the 16 subjects. The variance of subjects' game scores is positively correlated to their achieved maximum and mean scores. **b.** Comparison of subjective descriptions between sections with low, medium, high, and ultra-high averaged heart rates. The definition of each group is detailed in Appendix **??**. The x-axis of each subplot shows the adjective describing each section of games (Exciting, Joyful, Frustrating, Motivating, Amusing, Intimidating, Physically Demanding, Cognitively Demanding, Boring, Others). The y-axis indicates the percentage of subjects in the corresponding group who selected the corresponding answer. **c.** Each subplot shows the average game scores of all subjects on the five games against each robot policy. The error bars indicate the standard errors over the samples. The red horizontal lines indicate the best mean score achieved by one of the subjects against the corresponding robot policy.

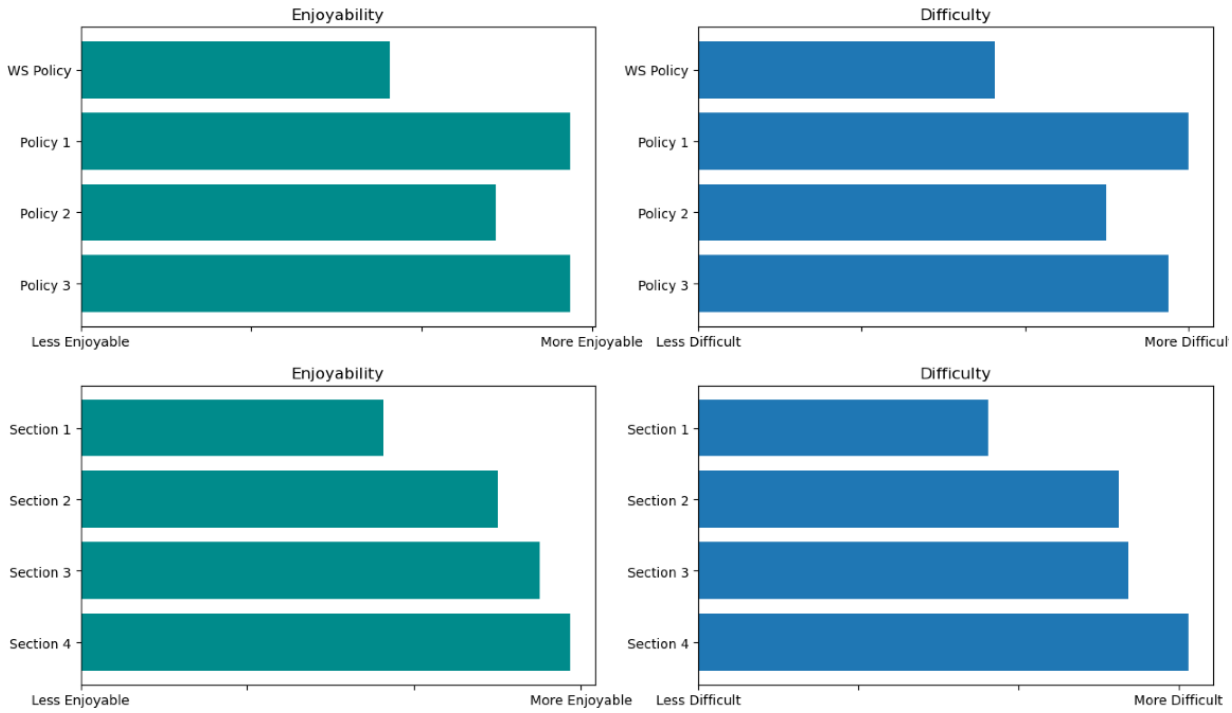

Figure 4.6: The average ranking comparison for enjoyability and difficulty across both policies and experiment sections.

Chapter 5

ROBOT LEARNING IN COMPETITIVE GAMES

In the previous section, we showcased how robots can derive highly effective policies through their participation in competitive games. Motivated by this observation, our next objective is to investigate the potential of framing other robotic challenges as competitive games. Here, an intelligent agent poses challenges to the robot, fostering a learning environment. Our discussion will encompass the challenges inherent in transforming robotic issues into competitive games and the methodologies we advocate for navigating these hurdles.

From the vantage point of machine learning, this learning journey of a robot through a competitive game is conceptualized as autocurriculum learning. This paradigm is part of the broader curriculum learning strategy. In this approach, the model commences with more straightforward tasks and progressively advances to intricate challenges. Such a structure resonates with human educational curricula, wherein complexity is phased in, paving the way for a logical learning progression.

The robotics community is increasingly adopting competitive autocurricular methods, especially for problems associated with adversarial learning for robust control and the learning of intricate robotic behaviors [30, 4, 186, 187]. However, a recurring challenge is the emergence of asymmetry in these scenarios. Often, one agent secures an early advantage, thereby monopolizing the game dynamics [138]. Such dominance can culminate in a less-than-ideal equilibrium, particularly when there's a constraint on computational resources, hindering extensive training with large batches.

Our research endeavors to rectify this asymmetry by altering the game dynamics via the Stackelberg game framework [175]. Within this two-player structure, the leader refines its strategy, keeping in mind the probable reactions of the follower. Concurrently, the fol-

lower optimizes its goals based on the leader's maneuvers. Such an arrangement grants the leader more favorable outcomes compared to what's achievable in conventional competitive games [6]. This framework proves beneficial when an agent holds primary significance or when there's a necessity to recalibrate power dynamics due to inherent or initial disadvantages, as our empirical analysis will further elucidate.

This work presents the following major contributions: **1. A Novel Autocurricular Algorithm:** We formulate the two-player competitive MARL problem as a Stackelberg game. By employing the total derivative Stackelberg learning update rule, we extend the state-of-the-art MARL algorithm, MADDPG [100], to a novel Stackelberg version, termed Stackelberg MADDPG (ST-MADDPG). **2. Re-balancing Asymmetry with ST-MADDPG:** We explicitly study how force exertion asymmetry affects agents' performance in the *competitive-cartpoles* environment (Fig.5.1.1), demonstrating how ST-MADDPG mitigates environmental asymmetry. **3. Enhanced Performance and Efficiency:** In a robust control problem (Fig 5.1.2), the leader advantage during adversarial training renders the resulting robot **3.19**× and **2.22**× more robust against adversarial and random intense disturbances, respectively, compared to the standard MARL setting. **4. Learning Complex Emergent Behaviors:** In a competitive fencing game (Fig 5.1.3), the leader's advantage in ST-MADDPG encourages certain attacking agents to act more aggressively and learn sophisticated strategies for superior performance. Notably, two of the top-performing attackers mastered the strategy of deceiving the opponent into less manipulable joint configurations, thereby temporarily limiting the opponent's maneuverability.

5.1 Preliminaries

In this section, we provide the requisite preliminary mathematical model and notation.

5.1.1 Two-player Competitive Markov Game

We consider a two-player zero-sum fully observable competitive Markov game (i.e., competitive MDP). A competitive Markov game is a tuple of $(\mathcal{S}, \mathcal{A}^1, \mathcal{A}^2, P, r)$, where $\mathcal{S}$ is the state

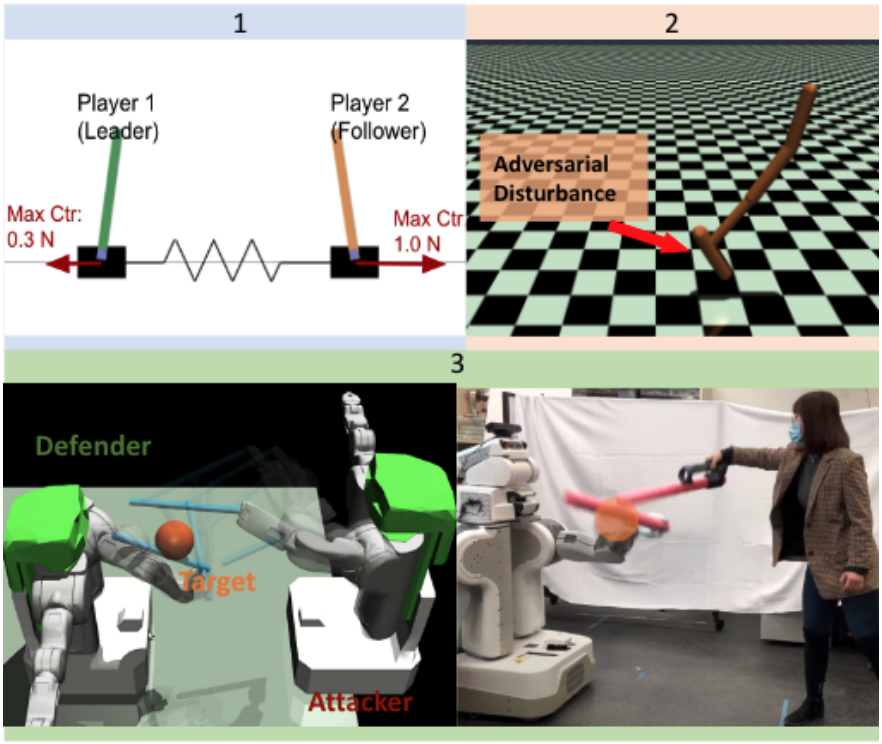

Figure 5.1: This work focuses on three competitive robotics tasks with physical interaction. 1. **Competitive-cartpoles** 2. **Hopper with adversarial disturbances** and 3. **The fencing game**. Although our experiment focuses on collecting a large number of gameplay samples from simulation to evaluate the algorithms, our simulation environment is tuned to represent the real-world challenges accurately. All the learned policies for the third environment support zero-shot transfer to our real PR2 robot. Video demonstration of the simulated and real robots' behaviors in various environments can be found on the project website - https://sites.google.com/view/stackelberg-autocurricula.

space, $s \in \mathcal{S}$ is a state, player $i \in \{1, 2\}$, A^i is the player i's action space with $a^i \in \mathcal{A}^i$. $P : \mathcal{S} \times \mathcal{A}^1 \times \mathcal{A}^2 \to \mathcal{S}$ is the transition kernel such that $P(s'|s, a^1, a^2)$ is the probability of transitioning to state s' given that the previous state was s and the agents took action (a^1, a^2) simultaneously in s. Reward $r : \mathcal{S} \times \mathcal{A}^1 \times \mathcal{A}^2 \to \mathbb{R}$ is the reward function of player

1 and by the zero-sum nature of the competitive setting, player 2 receives the negation of r as its own reward feedback. Each agent uses a stochastic policy π^i_θ, parameterized by θ^i.

A trajectory $\tau = (s_0, a^1_0, a^2_0, \ldots, s_T, a^1_T, a^2_T)$ gives the cumulative rewards or return defined as $R(\tau) = \sum_{t=0}^{T} \gamma^t r(s_t, a^1_t, a^2_t)$, where the discount factor $0 < \gamma \leq 1$ assigns weights to rewards received at different time steps. The expected return of $\pi = \{\pi^1, \pi^2\}$ after executing joint action profile (a^1_t, a^2_t) in state s_t can be expressed by the following Q^π function: $Q^\pi(s_t, a^1_t, a^2_t) = \mathbb{E}_{\tau\sim\pi}\big[\sum_{t'=t}^{T} \gamma^{t'-t} r(s_{t'}, a^1_{t'}, a^2_{t'}) | s_t, a^1_t, a^2_t\big]$, where $\tau \sim \pi$ is shorthand to indicate that the distribution over trajectories depends on $\pi : s_0 \sim \rho, a^1_t \sim \pi^1(\cdot|s_t), a^2_t \sim \pi^2(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a^1_t, a^2_t)$. ρ is the system's initial state distribution. The game objective is the expected return and is given by

$$\begin{aligned} J(\pi) &= \mathbb{E}_{\tau\sim\pi}\big[\sum_{t=0}^{T} \gamma^t r(s_t, a^1_t, a^2_t)\big] \\ &= \mathbb{E}_{s\sim\rho, a^1\sim\pi^1(\cdot|s), a^2\sim\pi^2(\cdot|s)}\big[Q^\pi(s, a^1, a^2)\big]. \end{aligned}$$

In a competitive Markov game, player 1 aims to find a policy maximizing the game objective, while player 2 aims to minimize it. That is, they solve for $\max_{\theta_1} J(\pi^1, \pi^2)$ and $\min_{\theta_2} J(\pi^1, \pi^2)$, respectively.

5.1.2 Stackelberg Game Preliminaries

A Stackelberg game is a game between two agents where one agent is deemed the leader and the other the follower. Each agent has an objective they want to optimize that depends on not only their own actions but also the actions of the other agent. Specifically, the leader optimizes its objective under the assumption that the follower will play the best response. Let $J_1(\theta_1, \theta_2)$ and $J_2(\theta_1, \theta_2)$ be the objective functions that the leader and follower want to minimize (in a competitive setting $J_2 = -J_1$), respectively, where $\theta_1 \in \Theta_1 \subseteq \mathbb{R}^{d_1}$ and $\theta_2 \in \Theta_2 \subseteq \mathbb{R}^{d_2}$ are their decision variables or strategies and $\theta = (\theta_1, \theta_2) \in \Theta_1 \times \Theta_2$ is their

joint strategy. The leader and follower aim to solve the following problems:

$$\max_{\theta_1 \in \Theta_1} \{J_1(\theta_1, \theta_2) \mid \theta_2 \in \arg\max_{\theta_2 \in \Theta_2} J_2(\theta_1, \theta_2)\}, \quad \text{(L)}$$

$$\max_{\theta_2 \in \Theta_2} J_2(\theta_1, \theta_2). \quad \text{(F)}$$

Since the leader assumes the follower chooses a best response $\theta_2^*(\theta_1) = \arg\max_\phi J_2(\theta_1, \theta_2)$, the follower's decision variables are implicitly a function of the leader's. In deriving sufficient conditions for the optimization problem in (L), the leader utilizes this information in computing the total derivative of its cost:

$$\nabla J_1(\theta_1, \theta_2^*(\theta_1)) = \nabla_{\theta_1} J_1(\theta) + (\nabla \theta_2^*(\theta_1))^\top \nabla_{\theta_2} J_1(\theta),$$

where $\nabla \theta_2^*(\theta_1) = -(\nabla^2_{\theta_2} J_2(\theta))^{-1} \nabla_{\theta_2 \theta_1} J_2(\theta)$[1] by the implicit function theorem [83].

A point $\theta = (\theta_1, \theta_2)$ is a local solution to (L) if $\nabla J_1(\theta_1, \theta_2^*(\theta_1)) = 0$ and $\nabla^2 J_1(\theta_1, \theta_2^*(\theta_1)) > 0$. For the follower's problem, sufficient conditions for optimality are $\nabla_{\theta_2} J_2(\theta_1, \theta_2) = 0$ and $\nabla^2_{\theta_2} J_2(\theta_1, \theta_2) > 0$. This gives rise to the following equilibrium concept which characterizes sufficient conditions for a local Stackelberg equilibrium.

Definition 1 (Differential Stackelberg Equilibrium [38]) *The joint strategy profile* $\theta^* = (\theta_1^*, \theta_2^*) \in \Theta_1 \times \Theta_2$ *is a differential Stackelberg equilibrium if* $\nabla J_1(\theta^*) = 0$, $\nabla_{\theta_2} J_2(\theta^*) = 0$, $\nabla^2 J_1(\theta^*) > 0$, *and* $\nabla^2_{\theta_2} J_2(\theta^*) > 0$.

The Stackelberg learning dynamics derive from the first-order gradient-based sufficient conditions and are given by $\theta_{1,k+1} = \theta_{1,k} - \alpha_1 \nabla J_1(\theta_{1,k}, \theta_{2,k})$, and $\theta_{2,k+1} = \theta_{2,k} - \alpha_2 \nabla_{\theta_2} J_2(\theta_{1,k}, \theta_{2,k})$, where α_i, $i = 1, 2$ are the leader and follower learning rates.

5.1.3 MADDPG

Lowe, et al.[100] showed that naïve policy gradient methods perform poorly in simple multi-agent continuous control tasks and proposed a more advanced MARL algorithm termed

[1]The partial derivative of $J(\theta_1, \theta_2)$ with respect to the θ_i is denoted by $\nabla_{\theta_i} J(\theta_1, \theta_2)$ and the total derivative of $J(\theta_1, h(\theta_1))$ for some function h, is denoted ∇J where $\nabla J(\theta_1, h(\theta_1)) = \nabla_{\theta_1} J(\theta_1, h(\theta_1)) + (\nabla h(\theta_1))^\top \nabla_{\theta_2} J(\theta_1, h(\theta_1))$.

MADDPG, which is one of the state-of-the-art multi-agent control algorithms. The idea of MADDPG is to adopt the framework of centralized training with decentralized execution. Specifically, they use a centralized critic network Q_w to approximate the Q^π function, and update the policy network π^i_θ of each agent using the global critic. Consider the deterministic policy setting where each player has policy μ_{θ_i} with parameter θ_i.[2] The game objective (for player 1) is $J(\theta_1, \theta_2) = \mathbb{E}_{\xi\sim\mathcal{D}}\big[Q_w(s, \mu_{\theta_1}(s), \mu_{\theta_2}(s))\big]$, where $\xi = (s, a^1, a^2, r, s')$, $\mathcal{D}$ is a replay buffer. The policy gradient of each player can be computed as $\nabla_{\theta_1} J(\theta_1, \theta_2) = \mathbb{E}_{\xi\sim\mathcal{D}}\big[\nabla_{\theta_1}\mu_{\theta_1}(s)\nabla_{a^1}Q_w(s, a^1, a^2)|_{a^1=\mu_{\theta_1}(s)}\big]$, and $\nabla_{\theta_2} J(\theta_1, \theta_2) = \mathbb{E}_{\xi\sim\mathcal{D}}\big[\nabla_{\theta_2}\mu_{\theta_2}(s)\nabla_{a^2}Q_w(s, a^1, a^2)|_{a^2=\mu_{\theta_2}(s)}\big]$.

The critic objective is defined as the mean square Bellman error:

$$L(w) = \mathbb{E}_{\xi\sim\mathcal{D}}[(Q_w(s, a^1, a^2) \\ -(r + \gamma Q_{w'}(s', \mu_{\theta'_1}(s'), \mu_{\theta'_2}(s')))^2]. \tag{5.1}$$

where $Q_{w'}$ and $\mu_{\theta'_1}, \mu_{\theta'_2}$ are target networks obtained by polyak averaging the Q_w and $\mu_{\theta_1}, \mu_{\theta_2}$ network parameters over the course of training.

With MADDPG in competitive settings, the centralized critic is updated by gradient descent and the two agents' policies are updated by simultaneous gradient ascent and descent $\theta_1 \leftarrow \theta_1 + \alpha^1\nabla_{\theta_1} J(\theta_1, \theta_2)$, $\theta_2 \leftarrow \theta_2 - \alpha^2\nabla_{\theta_2} J(\theta_1, \theta_2)$.

5.2 Stackelberg MADDPG Algorithm

In this section, we unveil our pioneering ST-MADDPG algorithm. A hallmark of ST-MADDPG is the leader agent's ability to leverage the understanding that the follower will react to its actions when formulating its gradient-based update. Specifically, the total derivative learning update offers the leader an edge by forecasting the follower's update during the learning process. This subsequently paves the way for convergence to the Stackelberg equilibrium in a myriad of applications, including generative adversarial networks and actor-critic networks [38, 200]. As outlined by Başar and Olsder [6, Chapter 4], within a two-player

[2]Following the setting and notation in origin DDPG algorithm [100], we use μ to represent deterministic policy to differentiate it from stochastic ones.

game framework where the follower's best responses are unique, the leader's payoff at the Stackelberg equilibrium surpasses that at the Nash equilibrium. This superior payoff is a sought-after attribute in numerous applications. The comprehensive ST-MADDPG algorithm can be found in Algorithm 4.

Setting player 1 to be the leader, the ST-MADDPG policy gradient update rules for both players are given by:

$$\theta_1 \leftarrow \theta_1 + \alpha^1 \nabla J(\theta_1, \theta_2), \tag{5.2}$$

$$\theta_2 \leftarrow \theta_2 - \alpha^2 \nabla_{\theta_2} J(\theta_1, \theta_2), \tag{5.3}$$

where the total derivative in the leader's update is given by

$$\nabla J(\theta_1, \theta_2) = \nabla_{\theta_1} J(\theta_1, \theta_2) - \nabla_{\theta_1 \theta_2} J(\theta_1, \theta_2)(\nabla^2_{\theta_2} J(\theta_1, \theta_2))^{-1} \nabla_{\theta_2} J(\theta_1, \theta_2). \tag{5.4}$$

The second order terms of the total derivative in (5.4) can be computed by applying the chain rule:

$$\begin{aligned}
\nabla_{\theta_1 \theta_2} J(\theta_1, \theta_2) &= \mathbb{E}_{\xi \sim \mathcal{D}}[\nabla_{\theta_1} \mu_{\theta_1}(s) \nabla_{a^1 a^2} Q_w(s, a^1, a^2) \\
&\qquad (\nabla_{\theta_2} \mu_{\theta_2}(s))^T |_{a^1 = \mu_{\theta_1}(s), a^2 = \mu_{\theta_2}(s)}], \\
\nabla^2_{\theta_2} J(\theta_1, \theta_2) &= \mathbb{E}_{\xi \sim \mathcal{D}}[\nabla^2_{\theta_2} \mu_{\theta_2}(s) \nabla_{a^2} Q_w(s, a^1, a^2) \\
&\quad |_{a^2 = \mu_{\theta_2}(s)}].
\end{aligned}$$

To estimate the total derivative $\nabla J(\theta_1, \theta_2)$, every component of (5.4) is determined by sampling from a replay buffer. The inverse-Hessian-vector product can be proficiently calculated using the conjugate gradient method [200].

In this study, we determine the implicit map regularization hyperparameter λ through a grid search for the first and second environments. Generally, given that the neural network can be highly non-convex, the Hessian inverse might become ill-conditioned. Incorporating a larger regularization restricts the gradient from exploding, leading to more consistent learning

dynamics. This observation aligns with findings in our experiments and other Stackelberg learning applications [38, 200]. Determining the optimal balance between Stackelberg and conventional gradient learning—whether by optimally selecting or adaptively adjusting the regularization—is a promising avenue for future research.

Algorithm 4: ST-MADDPG algorithm

for episodes $k = 1, 2, \ldots, K$ **do**
  receive initial state s_0;
  for $t = 1, 2, \ldots, T$ **do**
    for each agent i, select action $a^i = \mu^i_\theta(s)$ according to the current policy;
    execute actions (a^1, a^2) and observe reward r and new state s';
    store (s, a^1, a^2, r, s') in replay buffer $\mathcal{D}$;
    $s \leftarrow s'$;
    sample a random minibatch of N transitions $(s_i, a^1_i, a^2_i, r_i, s'_i)$ from $\mathcal{D}$;
    update the critic by minimizing the loss by (5.1);
    update the leader policy using the total gradient computed by (5.2) and (5.4);
    update the follower policy using the policy gradient by (5.3);
    update the target networks;
  end
end

5.2.1 Implicit Map Regularization

The computation of the total derivative in the Stackelberg gradient dynamics necessitates the inverse of the follower Hessian, denoted by $\nabla^2_{\theta_2} J(\theta_1, \theta_2)$. In real-world reinforcement learning (RL) scenarios, given that policy networks might be deeply non-convex, the inversion of $(\nabla^2_{\theta_2} J(\theta_1, \theta_2))$ can pose challenges due to its potential ill-conditioned nature.

As a remedy, rather than directly computing this term, we often resort to a regularized version: $(\nabla^2_{\theta_2} J(\theta_1, \theta_2) + \lambda I)^{-1}$. This regularization technique can be construed as the leader perceiving the follower as optimizing a regularized cost, given by $J(\theta_1, \theta_2) + \frac{\lambda}{2}\|\theta_2\|^2$. However,

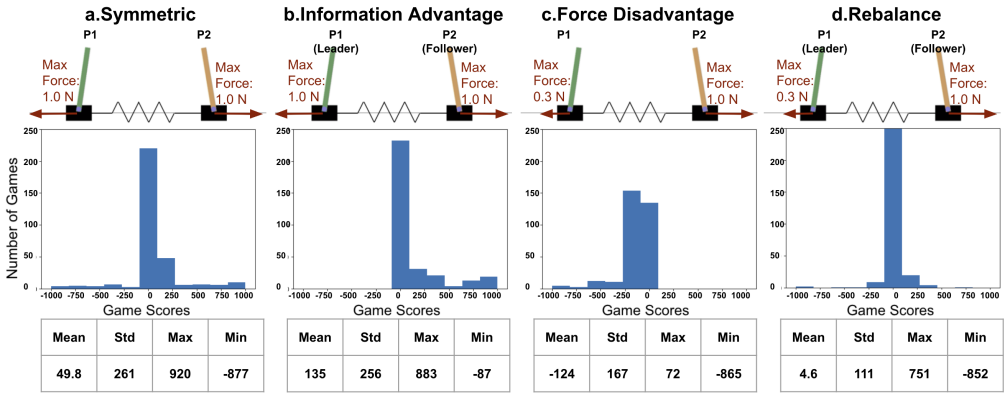

Figure 5.2: Statistical analysis of the learned policies' performance in four different variations of the competitive-cartpoles environment. The game scores refer to Player 1's scores, a game will have a positive score if player 1 wins, a negative score if player 2 wins, and zero if the two players are tied. ST-MADDPG can provide an advantage to the leader and improve its performance (i.e. column b). Given an asymmetric environment where one agent has a force exertion advantage over the other (i.e. column c), ST-MADDPG can be used to retain a balance in agents' performance (i.e. column d).

in reality, the follower is focusing on optimizing $J(\theta_1, \theta_2)$.

This regularization factor, λ, serves as a bridge between the Stackelberg and individual gradient updates for the leader.

5.2.2 Computational Complexity

Contrary to the standard policy gradient update, the leader's policy update necessitates the calculation of a Jacobian-vector product for the $\nabla_{\theta_1\theta_2} J(\theta_1, \theta_2)$ term, as well as an inverse-Hessian-vector product, $\nabla^2_{\theta_2} J(\theta_1, \theta_2)\nabla_{\theta_2} J(\theta_1, \theta_2)$, as seen in equation 5.4. The PyTorch automatic differentiation engine handles the computation of the Jacobian-vector product. The iterative conjugate gradient (CG) method, implemented within the same differentiation engine, computes the inverse-Hessian-vector term. As a result, all computations associated

with the Stackelberg gradient update are executed on the GPU. Each CG iteration computes a Hessian vector product, which is approximately 1.5 times the cost of a standard gradient [134]. Empirical evidence suggests that five CG iterations typically provide adequate numerical precision, making the leader update roughly 7.5 times more costly than a regular gradient [144, 38]. However, in practical applications, ST-MADDPG requires only about twice the time of MADDPG, primarily because the most time-consuming aspect of reinforcement learning is trajectory sampling, not gradient calculation.

5.3 Experiments

In this section, we delve into results from three experimental environments to address the subsequent queries:

- **(Q1)**: How do various asymmetries in the training environment influence agent performance and behavior?
- **(Q2)**: Can the advantage derived from ST-MADDPG for the leader compensate for the inherent disadvantages of a weaker agent?
- **(Q3)**: Does the total derivative update methodology of ST-MADDPG outperform alternative approximation methods?
- **(Q4)**: How can real-world robotic challenges be addressed by harnessing the Stackelberg informational structure?

It's important to highlight that in many competitive MARL environments, the trend of cumulative reward during learning doesn't display a consistent increase, as it typically does in well-tuned single-agent or cooperative MARL scenarios. To assess agent performance, we gather gameplay data by allowing the trained agents to engage in several matches against their training partner or a pre-designed benchmark opponent. More detailed procedures are delineated in Sections 5.4, 5.5, and 5.6.

Competitive-Cartpoles: To address **Q1** and **Q2**, we introduced a two-player zero-sum competitive game focused on a one-dimensional control task. Illustrated in Figure 5.2, the environment features two standard cartpole agents. A spring interconnects the dynamics of the two agents, with each end attached to an agent's body. Both agents earn a zero reward when they successfully maintain their respective poles in an upright position at the same time. Should one agent lose its balance, it incurs a reward of -1 for every ensuing time step until the game's conclusion. Conversely, the agent that remains balanced receives a reward of $+1$ for every time step, continuing until it, too, loses balance, concluding the game. Consequently, each agent's objective is twofold: firstly, to keep its pole upright, and secondly, to destabilize its opponent by exerting disruptive forces through the spring.

Hopper with Adversarial Disturbance: To address **Q3**, we delve into the development of a robust control policy for the classic hopper environment through adversarial training [31, 138]. In this scenario, the primary agent operates the traditional hopper robot, which comprises four rigid links and three actuated joints. Simultaneously, the secondary agent is trained to exert adversarial two-dimensional forces onto the hopper's foot.

The Fencing Game: To delve deeper into **Q3**, we explore a zero-sum competitive game introduced by Yang et al. [186]. This game encapsulates various real-world robotic challenges, including intricate robot kinematics, unpredictable transition dynamics, and a notably asymmetric game mechanism. In this two-player confrontation, the attacker strives to maximize its score by targeting a predefined area with a sword while avoiding contact with the defender's blade. Conversely, the defender seeks to diminish the attacker's score by safeguarding the target zone. Comprehensive game rules are elaborated upon in Section 4.1.3.

5.4 Learning Under Asymmetric Advantage

In this experiment, we assessed the equilibrium in two symmetric and two asymmetric competitive-cartpoles settings. Four pairs of agents were created with different random seeds and evaluated against unseen opponents in a tournament. For additional details, refer to the project website.

The competitive-cartpoles game training and experimentations were constrained to 1000 time steps as the maximum duration. Initiation of the training saw each agent executing 10^4 steps, leveraging uniform-random action selection to bolster exploration in line with the methodologies outlined by [94]. The replay buffer's capacity was set at 10^6, accompanying a learning rate of 10^{-3}, and every agent underwent updates using a batch size of 10^2.

Two distinct tournaments were orchestrated to accumulate evaluation data: one set of agents emerged from MADDPG and the other from ST-MADDPG. Within the framework of a tournament, every one of the quartet of player 1 agents—originating from the four respective random seeds—participated in 20 games against each of the four player 2 agents. This resulted in an accumulation of 320 game scores and their associated trajectories.

The regularization coefficient, denoted as λ, was universally fixed at one across all ST-MADDPG training sessions. This decision was instrumental in thwarting gradient explosion, while simultaneously ensuring the establishment of a robust leadership advantage during training. The choice of this evaluation methodology was strategic. It not only facilitated gameplay for an agent against its native co-evolving adversary but also permitted matchups against opponents trained with alternative random seeds. This ensured a holistic and comprehensive review for each set of training outcomes.

5.4.1 Leader Advantage

This experiment commenced with a symmetric competitive-cartpoles environment, where both agents shared identical action capabilities. To probe the influence of the leader advantage inherent in the Stackelberg game structure on the auto-curriculum process, we deployed

both MADDPG and ST-MADDPG methods in the competitive-cartpoles environment. The MADDPG training symbolizes a symmetric environment, while the ST-MADDPG training grants a leader's edge to player 1. For each method, we fashioned four agent pairs with unique random seeds. To juxtapose the performances between the two training strategies, we conducted a tournament yielding 320 game scores and trajectories for each method. The specifics of this tournament are elaborated in Appendix **??**. The first two columns of Figure 5.2 encapsulate the statistics of both tournaments. It's pertinent to mention that the tournament game scores in this segment correspond to player 1's scores. A positive score indicates player 1's victory, a negative one suggests player 2's triumph, and a zero denotes a tie.

In the symmetrical setup (i.e., MADDPG), the performances of players 1 and 2 are analogous. The tournament averaged a score of 49.8. Although a significant chunk of games scored between -90 and 90, the remaining spanned a vast score spectrum ranging from -877 to 920. This reveals that while both players generally display comparable prowess, either can sporadically overshadow the other substantially. Conversely, with the incorporation of a leader advantage during training (i.e., ST-MADDPG), player 1 clinched more victories, accruing a higher average score of 135. Player 2's most commendable win was limited to a score of -87, implying that the follower seldom posed a formidable challenge to the leader. This corroborates the leader's superior performance relative to the follower. Upon scrutinizing the agents' tactics by revisiting the accumulated trajectories, we discerned that agents emerging from the symmetric milieu predominantly engaged in fierce contests, exerting mutual force via the spring. While adept at maintaining their poles' equilibrium, they seldom destabilized their adversary sufficiently to secure a win. In contrast, ST-MADDPG-bred agents saw the leader master a strategy to tug the follower out of the playing field, thereby clinching the game. A visual showcase of these robotic maneuvers is accessible on the project website.

5.4.2 Re-balancing Asymmetric Environment

With the premise that ST-MADDPG confers a benefit that enhances the leader's performance, we were intrigued to ascertain if this edge could offset a disadvantage imposed on an agent due to an asymmetric environment. To this end, we crafted an asymmetric competitive-cartpoles setting where player 1 grappled with a force disadvantage. Specifically, player 1's maximum control effort was curtailed to merely 30% of that of player 2. Subsequent to this, agents were trained employing both MADDPG and ST-MADDPG methodologies (with player 1 designated as the leader) using four distinct random seeds, and evaluation data was collated via tournaments.

As depicted in Figure 5.2, when burdened with a pronounced force disadvantage, the prowess of player 1 was markedly diminished relative to player 2 in the aftermath of MADDPG training. Yet, when the Stackelberg gradient updates were used, the performances of the two contenders became indistinguishable. With a mean score of 4.57, a peak score of 751, and a trough at -852, it's evident that **the leader's advantage effectively counterbalances the force disadvantage**. Consequently, player 1 produced a score distribution akin to the symmetric environment delineated earlier.

5.5 Hopper Against Adversarial and Random Disturbance

This study zeroes in on the derivation and deployment of a MADDPG algorithm variant, leveraging the total derivative learning update to forge a Stackelberg informational structure. Notably, while other standalone learning algorithms have garnered traction in auto-curriculum studies [30, 4], their decentralized value network estimations through surrogate functions render direct total derivative computations impractical. A workaround is to approximate the Stackelberg information structure by differentially adjusting update steps for the leader and the follower within a standalone learning context [145]. This approximates the follower's optimal responsive update in a Stackelberg game by letting the follower execute markedly more update steps than the leader for each training data batch.

Adversarial Disturbance				
	ST-MADDPG	MADDPG	ST-PPO	PPO
Avg. Score	5113.6	1601.6	521.8	473.6
Std	3333.0	753.2	147.1	91.4
Statistical Significance	$p < 0.01$		$p > 0.05$	
No Random Disturbance				
	ST-MADDPG	MADDPG	ST-PPO	PPO
Avg. Score	5415.2	2436.8	510.0	495.7
Std	3730.8	2265.5	84.6	102.0
Statistical Significance	$p < 0.05$		$p > 0.05$	
0.1N Random Disturbance				
	ST-MADDPG	MADDPG	ST-PPO	PPO
Avg. Score	5463.3	2423.6	530.9	486.0
Std	3575.6	2178.2	101.3	75.7
Statistical Significance	$p < 0.05$		$p > 0.05$	
10 N Random Disturbance				
	ST-MADDPG	MADDPG	ST-PPO	PPO
Avg. Score	4928.3	1991.2	540.1	474.3
Std	3588.0	2084.6	151.3	80.8
Statistical Significance	$p < 0.05$		$p > 0.05$	

Table 5.1: Performance comparison between ST-MADDPG, MADDPG, ST-PPO, and PPO trained Hopper agents under three levels of random disturbances. Evaluation data between the Stackelberg and regular versions of both algorithms are compared via a statistic significant test (i.e. u-test)

Tackling this robust control challenge, our initial comparison pitted ST-MADDPG against MADDPG. Subsequently, we assessed the approximated Stackelberg update using two PPO-driven training configurations. The preliminary PPO strategy trained all agents using native PPO in a decentralized fashion, a typical approach in auto-curriculum literature [30, 4, 186]. We then innovated a PPO variant, dubbed ST-PPO, which endowed the follower (adversarial disturbance) with the capacity to take tenfold update steps compared to the leader (hopper).

Table 5.1 illustrates that ST-MADDPG-trained hopper agents markedly eclipsed their MADDPG-trained counterparts under both adversarial onslaughts (with ST-MADDPG registering an average reward of 5113.6 compared to MADDPG's 1601.6 — a staggering average **increase of 319.3%**) and diverse intensities of random disturbances (where ST-MADDPG's average rewards consistently outdid MADDPG's by a factor of at least **2.22**). In stark contrast, the approximated Stackelberg informational structure only eked out a modest uptick in performance for the leader during adversarial training, averaging a mere **11% boost**. Similarly, ST-PPO's hopper agents marginally outstripped standard PPO agents in the randomized disturbance tests, registering a maximum improvement factor of **1.14**. Consequently, **bestowing a leadership advantage on the robot during adversarial training can substantially amplify the robustness of a robotic control policy**. With comparable computational complexity (elaborated upon in 5.5.3), **the total derivative update proved superior to its approximated Stackelberg counterpart in constructing the leader's advantage**.

5.5.1 Experimental Details

In the specified training environment, game durations were intentionally limited to a maximum of 1000 time steps. This decision was motivated by the intent to expedite the training process. Nevertheless, when the phase shifted to evaluation experiments, the length of trials was extended to 3000 time steps. This modification was purposeful—it was to ensure a more discerning differentiation of agent performances.

Drawing parallels with the competitive-cartpole setup, every training phase in this envi-

ronment was initiated with 10^4 steps. Uniform-random action selection was the methodology of choice, aiming to augment exploration, in accordance with the techniques recommended by [94].

Further configuration details include:

- Replay buffer was scaled to accommodate 10^6 data points.
- The designated learning rate was determined at 8×10^{-5}.
- Each agent underwent periodic updates using batches of size 10^2.

In this specific training environment, the regularization parameter, λ, for all ST-MADDPG trainings was consistently set to 5000.

5.5.2 Extra Experimental Result

We utilized the MADDPG, ST-MADDPG, PPO, and ST-PPO algorithms to generate 20 pairs of hoppers and adversaries, each initialized with one of the 20 random seeds. Every agent pair underwent evaluation across 10 games, yielding a total of 200 game scores per method.

Adversarial Attack. Under adversarial attacks, hopper agents trained with ST-MADDPG consistently demonstrated longer survival durations, achieving an average reward of **5113.6**. This was notably superior to the **3333.0** avg. reward garnered by their MADDPG-trained counterparts. Comparatively, ST-PPO trained hopper agents (**522.0** avg. reward) exhibited a slight advantage over those trained with PPO (**473.6** avg. reward).

High Intensity Random Disturbance. The resilience and adaptability of hopper agents were scrutinized under three specific conditions: a neutral environment devoid of disturbances and two environments characterized by pronounced random disturbances (i.e., $0N$,

$0.1N$, and $10N$ respectively). Each agent was subjected to 100 trials within each environment, and the results are tabulated in Table 5.1. Remarkably, even though the training's adversarial strength was capped at $0.001N$, a significant number of agents (for instance, MADDPG: 50% and ST-MADDPG: 100%) still managed to attain an average reward exceeding 1000 when faced with the maximum disturbance strength of $10N$. In all tested disturbance intensities, agents trained with ST-MADDPG significantly surpassed the performance of those trained with MADDPG. Consequently, the evidence suggests that ST-MADDPG policies exhibit robustness, ensuring commendable performance even under unforeseen circumstances.

5.5.3 Total Derivative Stackelberg Update V.S. Approximated Stackelberg Update

As elucidated in section 5.2.2, the total derivative Stackelberg update step is approximately 7.5 times longer than a standard gradient step. In our experiments, the approximated Stackelberg update for the follower necessitated ten times the number of update steps compared to the leader. For instance, if a regular policy gradient method necessitates time t to update a single agent's policy for each epoch (excluding trajectory sampling time), the total derivative update method would require $7.5t$ for the leader update and t for the follower update. Conversely, the approximation update method takes t for the leader update and $10t$ for the follower update. Hence, in our experimental setup, the computational complexity of the total derivative update is marginally lower than that of the approximation update. Despite their comparable computational complexities, the total derivative update demonstrated a superior ability to establish the Stackelberg information structure. The efficacy of the approximation approach could be enhanced by further amplifying the number of update steps for the follower, albeit at a computational cost. As observed by Rajeswaran et al. [145], the approximation method's performance reaches a more satisfactory level when the follower undergoes $25\times$ the number of update steps compared to the leader.

5.6 Co-evolution Under Complex Environment

The above experiments are predicated on environments characterized by simplistic robot dynamics and game rules. However, real-world robotic scenarios can be substantially more intricate. To delve deeper into competitive co-evolution within the framework of a complex environment governed by the Stackelberg structure, we assessed various training configurations in the Fencing Game. This game encapsulates challenges more representative of tangible robotic issues than the preceding environments. The intricate kinematics of the two humanoid agents, each with seven degrees of freedom, coupled with the contact-intensive essence of the game, magnify the complexity and unpredictability of the environmental transition model. Additionally, the game's mechanics are inherently asymmetric. As the protector garners rewards solely from making contact with the attacker within a designated target area, the attacker takes the reins of the co-evolutionary trajectory, instigating attacking maneuvers. Should the attacker adopt a strategy of inaction to dodge penalties, the policy updates for both agents might wane in efficacy, ensnaring them in suboptimal exploration realms within the task space. Given its pivotal role, our attention in this experiment is honed in on assessing the attacker's behavior and prowess.

Our objective was to discern the impact of diverse Stackelberg configurations on the caliber of co-evolutionary progress. To this end, attackers stemming from MADDPG were juxtaposed against those from six distinctive ST-MADDPG paradigms. In these ST-MADDPG configurations, both the protector and the attacker were alternately designated as leaders, under the aegis of varying regularization values: small (λ=50), medium (λ=500), and large (λ=1000). This design facilitated the assessment of co-evolutionary trajectories under varying leadership strengths, ranging from strong to moderate to weak. For each of the seven configurations, ten agent pairs underwent training, guided by ten unique random seeds. Subsequent to training, the 70 polished attackers were put to the test in 100 games, pitted against a meticulously crafted heuristic-based protector policy. This policy, anchored in the game's fundamental tenets, positions the protector's sword as a barrier between the target

zone and the closest point on the attacker's sword, executing a formidable defensive stance. The introduction of this heuristic benchmark policy sets the stage for a level playing field among attackers. It stands to reason that superior co-evolutionary trajectories, culminating in a higher-grade equilibrium, would birth attacker strategies more resilient and adept at facing off against this formidable, previously unencountered protector.

5.6.1 Heuristic-based Protagonist Policy

We sought to design a robust baseline heuristic policy to facilitate an intense human-robot gameplay experience. Given a world observation, the robot aligns its bat perpendicular to the human's bat, introducing random angular offsets uniformly drawn from -25 to 25 degrees on the x, y, and z axes. To ensure that the robot consistently adopts a competitive defense, the policy directs the robot to position the center of its bat midway between the target area and the point on the human's bat nearest to that target area.

$$\bar{b_p} = \bar{tar} + (\bar{h_{close}} - \bar{tar}) \cdot uniform(0.5, 1)$$
$$\bar{h_{close}} = \bar{h_{low}} + ht \cdot (\bar{h_{up}} - \bar{h_{low}})$$
$$ht = \max\left(0, \min\left(1, (\bar{tar} - \bar{h_{low}}) \cdot (\bar{h_{up}} - \bar{h_{low}})/(2 \cdot L_{sword})\right)\right)$$

Let $\bar{b_p}$, $\bar{tar}$, $\bar{h_{up}}$, and $\bar{h_{low}}$ denote the position of the robot's bat frame, the center of the target area, the upper end of the human's bat, and the lower end of the human's bat, respectively. The point on the human's bat closest to the center of the target area is represented by $\bar{h_{close}}$, while L_{sword} denotes the length of a bat. The function $uniform(0.5, 1)$ randomly decides the distance between the robot's bat and the human's bat. Moreover, there's a 50% probability that the robot will maintain the bat position computed from the previous time step, instead of adopting the most recent desired pose. This added variability imparts a degree of unpredictability to the robot's actions. Such a heuristic enables the robot to excel in the fencing game when its movement speed matches or exceeds that of its opponent. In these experiments, both the antagonist and protagonist agents have identical physical capabilities.

5.6.2 Pre-training

The fencing environment is notably more intricate compared to the other two experimental environments. Hence, we opted to initiate the co-evolution process for all seven settings (i.e., LA0_Reg50, LA0_Reg500, LA0_Reg1000, Normal, LA1_Reg50, LA1_Reg500, and LA1_Reg1000 as depicted in Fig.4.1) using the same pair of pre-trained policies. These pre-trained policies were developed through iteratively executing two rounds of best response individual updates for each agent using MADDPG, mirroring the pre-training process described by Yang et al. [186]. Each iteration of the best response update begins with 10^4 steps of uniform-random action selection. We set the replay buffer size to 10^6, adopted a learning rate of 8×10^{-4}, and updated each agent using a batch size of 100.

5.6.3 Co-evolution

Both the protector (i.e., agent 0) and the attacker (i.e., agent 1) agents were initialized using the pre-trained policies across all seven training settings and for each of the 10 random seeds. Unlike prior settings, there was no initial phase of uniform-random action selection at the start of each co-evolution process. The replay buffer size was set at 10^6. A learning rate of 8×10^{-5} was employed, and updates for each agent were performed with a batch size of 1024. As depicted in Fig.5.3, the agents quickly deviated from the pre-trained policies. They underwent substantial policy updates and displayed notable performance volatility during the course of the co-evolution process. However, by the conclusion of this process, the agents seemed to have reached a new equilibrium.

5.6.4 Increase of Attackers' Aggressiveness

In all seven training configurations, a substantial subset of attackers gravitated towards adopting conservative strategies. These strategies typically favored fewer attacking actions and yielded game scores that sat relatively close to neutral, neither significantly positive (indicative of winning) nor negative (suggesting defeat). This inclination towards caution is

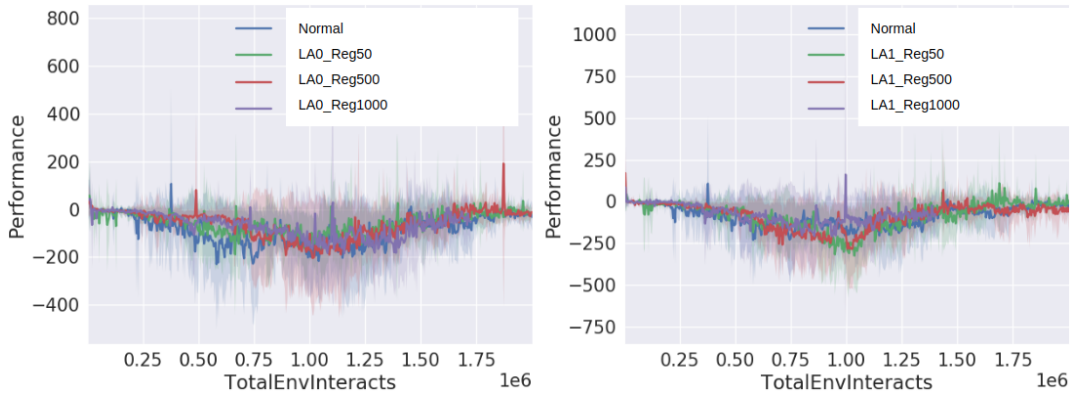

Figure 5.3: Both plots demonstrate the protector's (i.e. agent 0) average game score during the co-evolution process under the six different settings.

understandable when one considers that initiating an attacking move also opens the door to a considerable risk of incurring penalties from the protector. Yet, as illustrated in Fig.4.1.b., there is a noteworthy observation: as attackers accrued more advantages during training, a minority began to exhibit increasingly aggressive behavior, characterized by a higher frequency of successful strikes on the target area.

5.6.5 Emergent Complexity

Increased involvement in the competition often facilitates the development of superior attack strategies. As evidence, Fig.4.1.a. highlights that the top five performers among the attackers are predominantly positioned on the right half of the graph, where their co-evolving counterparts are not acting as leaders in a Stackelberg game. Intriguingly, the two most accomplished attackers emerged from ST-MADDPG configurations in which the attackers were designated as leaders, with regularization values of 500 and 50 respectively. Both managed to craft strategies that deceived the protector, luring them into less advantageous joint configurations. By doing so, these attackers were able to target the vulnerable area at a reduced risk while the protector found itself partially ensnared, consumed with extricating

itself from that compromising position. However, it's worth noting that not all aggressive policies guaranteed success. Some, despite their assertiveness, registered subpar performance, as exemplified by agent#5 from both LA1_Reg1000 and LA1_Reg500 in Fig.4.1.

5.7 Discussion

This study explores the application of MARL in asymmetric, physically based competitive games. We introduced the Stackelberg-MADDPG algorithm, which recasts a two-player MARL problem as a Stackelberg game, thereby conferring an advantage to one of the participating agents. Our findings highlight that an intrinsic advantage held by one agent can skew the training process towards undesirable equilibria. The Stackelberg-MADDPG algorithm aims to recalibrate these environments, thereby enhancing the efficacy of both agents. Additionally, our experiments reveal that certain problems might exhibit heightened sensitivity to random seeds, potentially diminishing the discernible impact of the proposed methodology. Among the limitations of our study is its restriction to two-player competitive scenarios and the requisite access to the opponent's parameters during the training phase.

Chapter 6

GAMIFYING WAREHOUSE MANIPULATION

The complete automation of industrial warehouses offers an array of benefits. Primarily, it bolsters efficiency and productivity through uninterrupted operations and optimized workflows. Such streamlining can translate into significant cost savings by reducing labor expenses and curtailing errors. Automation also enhances safety, reducing the risks often associated with manual operations. Moreover, automated systems allow for simpler scalability, adeptly adapting to fluctuating demands. The pressing need for automation is underscored by prevailing labor shortages and the ever-increasing variety of inventory in expansive e-commerce warehouses. However, realizing full-scale automation presents its challenges. Robots, in these contexts, grapple with the task of managing billions of diverse, unfamiliar items crammed onto shelves. The sheer density of the packing not only impedes the visual differentiation of items but also demands intricate robot manipulation techniques tailored to each unique shelf layout and item configuration. In this section, we pivot our focus towards harnessing game-inspired strategies to augment robotic capabilities in addressing a pertinent warehouse automation issue—robotic manipulation within tightly packed containers. Tackling warehouse automation through robotic manipulation is rife with complexities. We will outline the primary challenges associated with robotic manipulation in our warehouse context, clarify two central manipulation tasks fundamental to our research, and introduce a comprehensive strategy to surmount these challenges. The main research effort in this section concentrates on the design, implementation, and evaluation of a vital infrastructure: a novel physical suction grasping simulation essential for integrating game-inspired methodologies into warehouse tasks. Building on this infrastructure, we delineate a future research direction: a robotic learning system designed to master manipulation skills

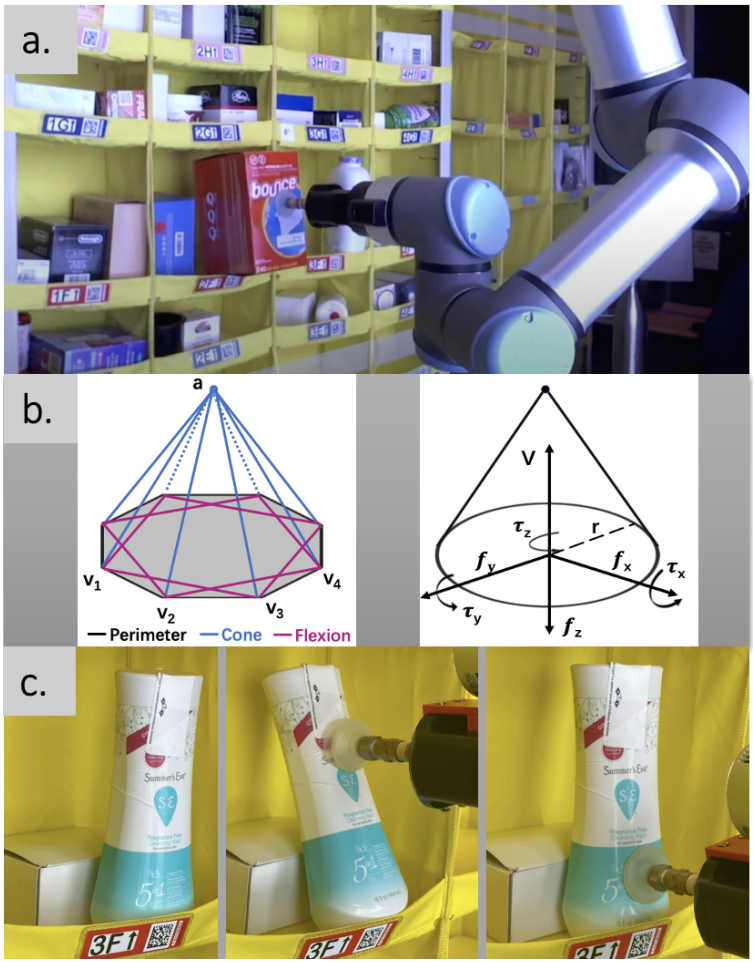

Figure 6.1: a. Suction grasping for real-world scenarios remains challenging due to limited analysis of object movements. b. SOTA methods only reason for object's surface properties. *Left*: The quasi-static spring model. *Right*: Wrench basis for the suction cup. [105] c. *Left*: A warehouse picking scenario. *Middle*: DexNet failing the grasp due to object toppling. *Right*: An effective grasp point that prevents unfavorable object movements. See the project website for experiment videos.

for this warehouse environment through both simulated and real-world interactive gameplay.

6.1 Problem Description

6.1.1 Warehouse Setup and Robotic System

Before delving into the specific challenges, let's first review the warehouse configuration and the robotic system utilized in this research. Our project zeroes in Amazon's 'pick-from-storage-pod' use case, which necessitates robotic picking of a diverse range of items from densely populated bins located on shelves (i.e. "pods"). A UR 16e robot, situated in front of a pod and outfitted with a Robotiq Epick vacuum suction gripper, comprises the core of our system. The software architecture is constituted by perception and control components, which are orchestrated by a state machine. The state machine adheres to the standard workflow of extracting items from a shelf. Furthermore, our system integrates a database and a web interface, both of which are instrumental in curating the inventory of the bins and in enabling a human operator to transmit pick requests to the state machine. In this proposal, we aim to enhance the robot's ability to pick up and reposition items within bins using a suction gripper, based on visual inputs.

6.1.2 Major Challenges and Project Goals

The warehouse environment in which we intend to operate presents multiple challenges. First, a robot needs to manage an **extensive range of unfamiliar objects** that can differ significantly in attributes such as material, weight, and shape. Additionally, the robot only gets a **partial view of the containers**, for instance, a 2.5D point cloud, rendering traditional planning methods insufficient. Most crucially, while we have opted for a suction gripper due to its ease of use and versatility, the majority of suction grasping methods assume minimal object movement during the operation, which aids in the deformation of the suction cup and the establishment of an air seal. However, in our case, as depicted in Figure.6.1.a, the container's side opening necessitates sideways manipulation. This is notably more difficult than top-down scenarios, as the robot's actions can trigger a series of object displacements, potentially causing items to shift or fall over. Furthermore, the warehouse

containers, made of fabric and reinforced with a plastic board and metal frame, are unstable during robot manipulation. The **complex object movements** arising from robot activity, object interaction, and container instability are notably difficult to simulate.

The challenge of diverse objects and partial observation introduces large uncertainties when robot observe and analyze the environment. For example, given a single view of the container, it is very hard for the robot to fully identify the shape, volume, and weight of each items in many cases. Moreover, factoring in the complex object movements during manipulation further complicates the robot's decision-making process regarding the appropriate actions to take.

As outlined in Chapters 4 and 5, the game-theoretic autocurricula learning paradigm offers a compelling strategy for enhancing a robot's ability to handle significant uncertainties with improved robustness. In line with this, our project aspires to construct a robotic system that learns through gamified interactions within the real world. Our approach involves training two robot control policies simultaneously within a general-sum game framework. The first policy is designed to enable the robot to generate complex container scenarios through basic swiping actions. The second policy, which is centered on non-prehensile manipulations, is focused on maneuvering the robot to uncover the graspable surface of an object to facilitate a successful grasp. One of the primary challenges associated with automated curriculum learning methods is their substantial requirement for numerous training samples. Given the resource-intensive nature of collecting real-world data, there is a pronounced dependence on accurate simulations to create realistic datasets for either training or pre-training objectives. Thus, the major amount of our research has been channeled into devising a simulation capable of faithfully mimicking suction grasping within a warehouse environment, which realistically models the dynamics between the suction cup and the objects. This simulation development is crucial as it addresses a complex virtual manipulation task, which involves identifying grasp points while considering the dynamic behavior of objects during the grasping sequence.

6.1.3 Problem Representation

In this warehouse manipulation task, the robot's goal is to identify one or more points on any objects in the container, using a single-view depth image observation, where executing actions will ultimately result in the successful pick-up of the target object. A point is defined by $[\mathbf{p}, \mathbf{v}]$. Here, $\mathbf{p} \in \mathbb{R}^3$ represents the center of the contact ring between the suction cup and the object, while $\mathbf{v} \in \mathbb{S}^2$ denotes the gripper's approach direction. To model object movements during the manipulation process, an object's state is denoted by its Cartesian pose and velocity in a workspace, represented as $s = (p, \delta p)$, the states of i objects in a container at time t can be represented as $\mathbf{s_t} = \{s_{t_0}, s_{t_1}, ..., s_{t_i}\}$. At each time step, a robot performs a pushing action $a_t = (f_t, \mathbf{p}, \mathbf{v})$, applying a force f_t to a specific point, $\mathbf{p}$, on the object's surface in the direction of $\mathbf{v}$. The state transition model $p(\mathbf{s_{t+1}}) = T(\mathbf{s_t}, a_t)$ provides a distribution over the potential movements of the objects during the picking process.

6.2 The Infrastructure for Gamification

By introducing gamification into the process, our aim is to establish a system in which one robot poses challenges for another robot designated for picking. This arrangement enables automated data collection as the robot consistently stows or reconfigures items inside a container. Adopting the autocurricular learning framework, the system can autonomously determine and fetch target objects, significantly curtailing human involvement. This approach permits robots to independently amass real-world data and sharpen their manipulation capabilities. Notably, the process of data generation is influenced by the robot's existing competencies, ensuring that the data produced aligns with areas needing additional enhancement and instruction. One of the main challenges associated with this task is the shifting of objects during the manipulation phase and realistically simulating the physical characteristics of the suction cup. To gamify this task effectively and overcome these hurdles, the crux of our research concentrates on devising a lifelike physical simulation that accurately represents object dynamics and the mechanics of suction-based grasping.

6.3 A Physical Simulation for Suction Grasping

Current methodologies for suction grasping predominantly follow a top-down approach. In this configuration, objects are initially situated on a steady, flat plane. The robot then attempts to grasp the item from above. This method is driven by the mechanics of the suction cup gripper, which demands that the robot exert a specific force to press the suction cup onto the object's surface. This pressure deforms the cup, producing an airtight seal, and culminating in a firm suction grasp. Therefore, any object to be grasped necessitates sturdy backing in the opposite direction to the robot's pressing motion. Absent this support, the object might shift undesirably, thwarting the suction cup's ability to form the needed seal. Nevertheless, there are myriad real-world situations where robots must seize objects lacking this steady backing—like picking from a side-opening container or an unstable stack of items. In these scenarios, objects may exhibit intricate dynamics due to displacement from the robot's maneuvers and interactions amongst the objects themselves. Leading-edge detection methods for suction grasping falter under these intricate object-picking situations because they disregard the mobility of objects during the manipulation stage. This shortfall dramatically constrains the applicability of suction grippers, barring them from realizing their utmost potential in tangible manipulation assignments. Fig.6.1.a showcases a tangible manipulation assignment.

In our research, we aspire to harness the full capabilities of suction grippers by formulating a grasp point detection model that not only contemplates quantitative metrics, such as the quality of suction, but crucially, factors in the object dynamics throughout the picking operation. This paper advances the following contributions: **1. Suction Grasping with Object Movement Awareness:** We delineate the challenge of intricate object movement during suction grasping—a facet overlooked by prevailing state-of-the-art techniques. **2. A Novel Open Source Suction Grasping Simulation:** In response to this challenge, we have fashioned a top-tier suction grasping simulation environment using Isaac Gym[**?**]. This simulation environment mirrors the influence of object dynamics on suction grasp efficacy

throughout the grasping procedure. **3. A Dataset and Refined Model:** By employing the simulation environment, we've curated a dataset comprising over a million simulated grasps and cultivated a grasp point detection model. This model integrates the interplay between object motion and the robot's kinematics in determining grasping success. **4. Evaluation in Tangible Warehouse Contexts:** We compared our model against two prominent grasp point detection methodologies. In both simulated and actual tests, our strategy outshined its counterparts in accuracy and consistency.

6.4 Modeling Suction Grasping and Object Movements

Our goal is to pinpoint optimal grasp locations on an object within a container filled with several items, based on insights from a single-view depth image. The grasp locations are intended to aid a robot in achieving a successful suction grasp, especially in scenarios where the object does not have firm support against the robot's applied force. Drawing from earlier works on suction grasp detection [105, 19, 106], we define a grasp location as a target point, denoted as $[\mathbf{p}, \mathbf{v}]$. Here, $\mathbf{p} \in \mathbb{R}^3$ symbolizes the intersection of the suction cup and the object, while $\mathbf{v} \in \mathbb{S}^2$ indicates the robot's approach vector. A grasp is labeled as 1 if a successful suction grasp is achieved, and 0 otherwise. This section elaborates on the essential criteria for a successful suction grasp.

6.4.1 Seal Quality and Wrench Resistance

The fundamental principle behind a suction cup's ability to lift objects is the air pressure differential across its membrane, created by a vacuum generator, pulling the object closer. A secure seal between the cup and the object is imperative for successful grasping. For seal assessment, we employ the quasi-static spring-based model from DexNet 3.0 [105]. As illustrated in Fig.6.1.b., this model integrates three spring systems to characterize the deformation of the suction cup. The perimeter springs gauge the deformation between successive vertices (v_i and v_{i+1}). The cone springs highlight the cup's structure deformation determined by the distance between vertex v_i and point a. Flexion springs, connecting v_i to v_{i+2}, resist

bending along the cup's surface.

After forming a seal with the object, the suction gripper must fend off disturbances like gravitational pulls. DexNet 3.0's suction ring contact model [105] encapsulates the forces a suction cup encounters during grasping. As shown in Fig.6.1.b., this model accounts for five forces: the actuated normal force (f_z) and vacuum force V - both are essential for gripping; frictional forces (f_x, f_y) and torsional friction (τ_z) emerge due to the normal force interaction; elastic restoring torques (τ_x, τ_y) result from the suction cup's innate elastic forces exerting torque on the object.

6.4.2 Object Dynamics

Traditional suction grasping methods often assume minimal object movement during grasping, ensuring the cup's deformation and airtight sealing. Yet, real-world scenarios might not offer sufficient support against the robot's force, causing unwanted object displacements and hampering the seal's formation. The dynamics become intricate when nearby objects influence the target. Our study captures these intricacies by considering object movements during the grasping phase, enriching the efficacy of suction grippers for real-world tasks. If we describe an object's state by its Cartesian pose and velocity in the workspace, symbolized as $s = (p, \delta p)$, the states of i objects in a container at time t are expressed as $\mathbf{s_t} = \{s_{t_0}, s_{t_1}, ..., s_{t_i}\}$. In every timestep, a suction-equipped robot performs an action $a_t = (f_t, \mathbf{p}, \mathbf{v})$, exerting force f_t on a specific object surface location, $\mathbf{p}$, following the direction $\mathbf{v}$. The state transition model $p(\mathbf{s_{t+1}}) = T(\mathbf{s_t}, a_t)$ details the possible object movements during grasping.

6.5 DYNAMO-GRASP

This section presents a robotic learning pipeline that develops a grasp point detection model. The model predicts suction grasp points by assimilating information related to object surface attributes and object motion during the picking phase. We introduced a novel simulation environment for suction grasping, which mirrors both the properties of a suction cup and

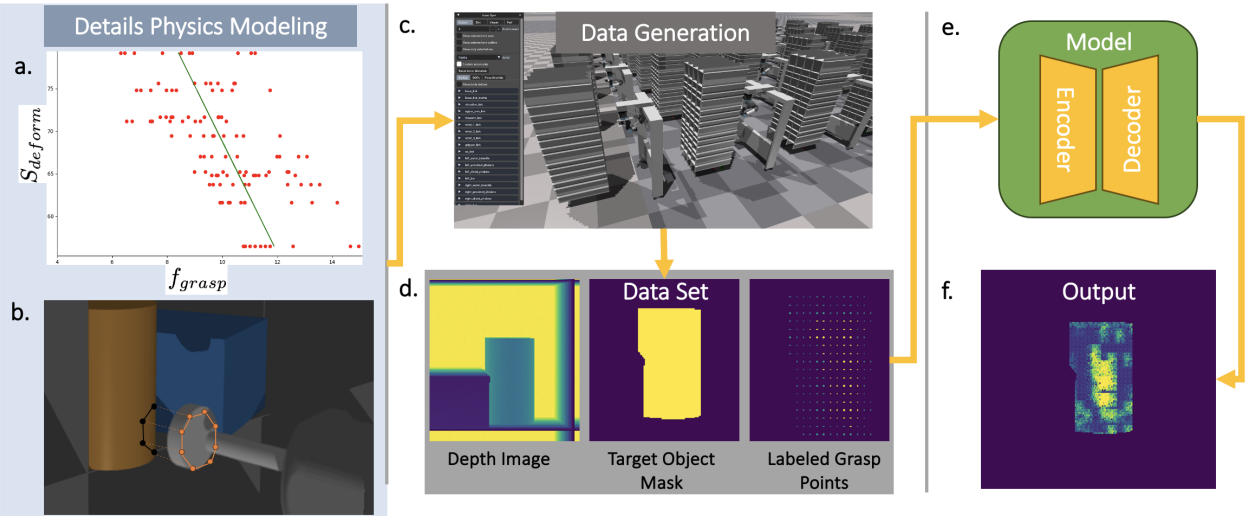

Figure 6.2: An overview of the proposed pipeline: **a.** We conducted system identification using 19 everyday objects of diverse shapes, weights, volumes, and materials to ascertain the function F discussed in Section 6.5. **b.** Calculation of deformation score at each simulation time step. **c.&d.** Generating dataset with our simulation environment. **e.&f.** Trained DYNAMO-GRASP model outputs an affordance map highlighting optimal grasp areas.

object movements influenced by robotic actions and inter-object interactions. Using depth imagery as input, a transformer-based model outputs an affordance map across the object's surface. This map reflects the success probability of a suction grasp upon a robotic push in different object regions. Importantly, our methodology prioritizes evaluating how physical interactions between robots and objects affect the efficacy of a suction grasp. During actual operations, we exclude grasp points with insufficient air seal and wrench resistance capabilities, leaning on DexNet's insights. The system's architecture can be observed in Fig.6.2.

6.5.1 Simulation Environment and Data Collection

Bypassing the need for costly real-world robotic data collection, we meticulously constructed a simulation environment. This environment faithfully reproduces the suction cup's physical properties, object motion from robotic grasping, and the robot's kinematics during picking. Our grasping simulation is rooted in Isaac Gym, facilitating GPU-accelerated computations. Although Isaac Gym doesn't provide comprehensive features simulating detailed suction grasping, our setup incorporates several bespoke functional modules. Our platform closely emulates the suction-picking sequence by considering factors like suction cup attributes, robot kinematic limitations, collision dynamics, control disturbances, and object behavior.

Suction Properties Modeling

Conventional physics simulations for robotics usually emulate suction grasping through rudimentary methods. These often involve directly connecting the object to the robot's end-effector or generating a magnetic force between them. Such methods overlook essential physical nuances. Specifically, to ensure a successful suction grasp, the suction cup needs to be adequately pressed and distorted so its rim attaches to the object's surface, forming an air-tight seal. Determining the force magnitude necessary for air seal creation is pivotal. This is particularly true when an object lacks a solid backing; applying sufficient force can displace it. Recognizing the force magnitude exerted on the object by the robot is key for accurate object dynamics simulation. To mimic the suction cup's deformation attributes, we employed the Perimeter Springs of the quasi-static spring system, as highlighted in Sec 6.4.1. Given a grasp point $\mathbf{p}$ on the object's surface and the incident angle $\mathbf{v}$, this model derives a suction deformation score S_{deform}. Using empirical data, we undertook system identification to deduce the function F which reveals the force required for successful grasping, contingent on a specific grasp point's deformation score.

We ran a system identification procedure to precisely determine the force the robot needs to exert to deform its suction cup, ensuring the cup's rim securely adheres to the object's

surface, creating an airtight seal. We selected 18 common objects, each showcasing varied surface geometries, in an effort to encompass a wide range of deformation profiles. With our UR16 robot, we performed ten suction grasps on each object. To ensure accurate measurements by reducing potential interference, the objects were securely held to prevent any movement during the grasp. The successful formation of a suction seal was identified by a notable reduction in the suction airflow and readings from the force-torque sensor positioned on the robot's wrist. Our findings revealed that the behavior of our suction cup varied considerably when comparing almost flat surfaces to those that were more curved or intricate. As a result, we decided to express the function $F(S_{deform}) \rightarrow f_{grasp}$ using a hybrid linear model:

$$F(S_{deform}) = \begin{cases} 7.66 - 0.06 * S_{deform} & \text{if } S_{deform} \leq 80 \\ 22.2 - 0.18 * S_{deform} & \text{otherwise} \end{cases}$$

Generally, the dimensions and rigidity of a suction cup determine its operational scope for handling objects of different sizes and weights. Nonetheless, these factors don't fundamentally change the essence of the grasping challenge. For example, when employing a smaller suction cup to handle lighter, compact objects, these objects often exhibit decreased friction with their container and diminished inertia, making them more susceptible to toppling. While we expect some level of adaptability to unfamiliar suction cups, we advise conducting the system identification procedure to ensure peak performance.

Grasping Physics Simulation

(1) Kinematics: Our simulation ingests a robot model and uses an end-effector controller to facilitate various suction grasps. This showcases how the robot's shape and kinematic attributes influence its grasp. *(2) Scenario Generation:* Our study is primarily anchored in warehouse lateral picking contexts. In the data collection phase, the simulator randomly picks one to three objects from our set, placing them in a container with varied positions and angles. We also employ domain randomization for observational noise, object weights, and control parameters. This ensures the dataset mirrors a plethora of physical properties

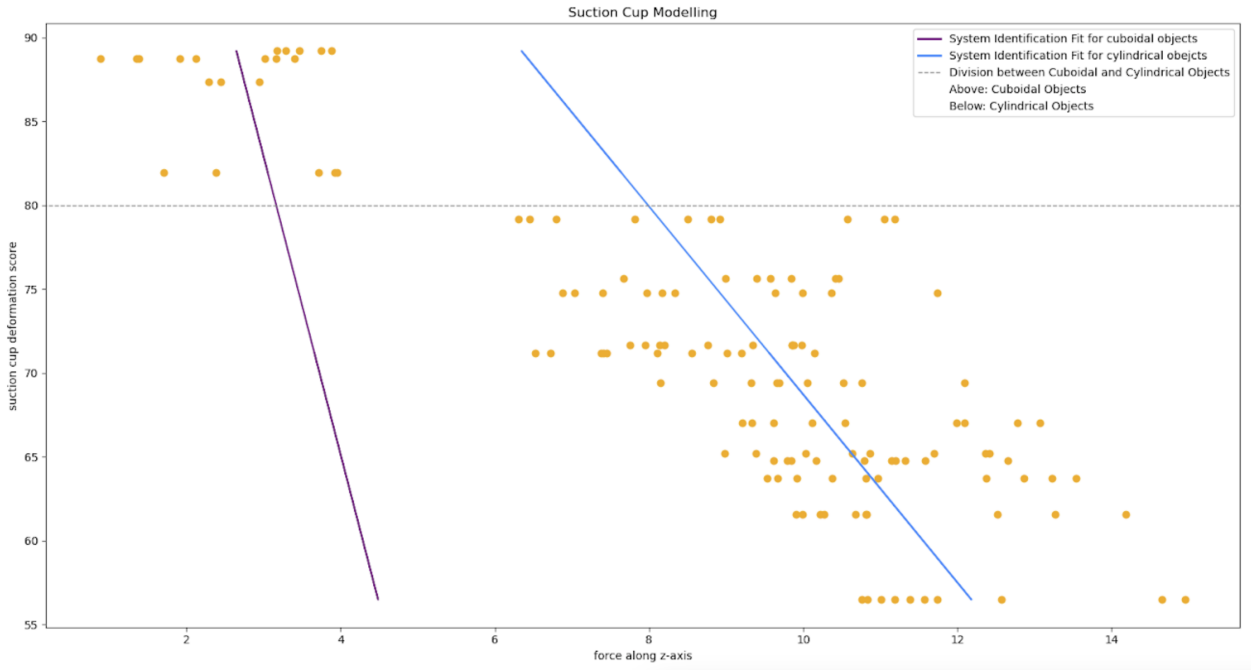

Figure 6.3: Force exerted on an object as a function of the suction deformation score. Solid lines represent system identification fits for cylindrical (blue-colored line) and cuboidal (violet-colored line) objects. The dotted line demarcates the distribution of data points between the two object types.

and robotic behaviors. One object in the container is randomly designated as the target for picking. *(3) Grasp Point Sampling:* For a given picking scenario, we sample two sets of potential grasp points from the target object's visible surface.

Data Labeling

Post grasp point sampling, our virtual robot "physically" enacts each candidate $\mathbf{p}$ by undertaking a sequence of pushing actions $\mathbf{A}$. The robot exerts consistent force $f_t = f_c$ if the target object is unstable. If the object achieves a position with ample push resistance, f_t gradually escalates until the suction cup is adequately distorted to form an air seal or the object begins moving once more. Calculating the suction cup's distortion and accurately

estimating suction grasp involves a real-time evaluation of S_{deform} and f_{grasp} during each interval. This assessment gauges the suction cup's present position concerning the target object, as illustrated in Fig.6.2.b. A sensor on the end-effector constantly tracks f_t. A grasp is deemed successful if $f_t \geq f_{grasp}$. Any deviations from this benchmark lead to the grasp point being labeled as unsuccessful. For triumphant grasps, a penalization term p_{move} is integrated into the label to counteract unwarranted object movements.

A grasp point that doesn't achieve a stable suction grip is given a definite score of zero. Furthermore, the label is marked as a 'failure' if the robot arm doesn't align and pick up the object following the calculated angle of incidence based on the surface normals, ensuring the grasp matches the determined optimal orientation. It's also vital that the arm avoids any unintentional contact with surrounding objects before touching the target. Such unintended collisions can jeopardize the quality of the grasp, potentially leading to errors or even damage. In contrast, successful grasp points are evaluated using the formula $s = 1 - p_{move}$, where,

$$p_{move} = \max(0, \min(obj\_movement, 0.3))$$

$$obj\_movement = \sum_{t=0}^{T-1} \left(||\text{tran}_{t+1} - \text{trans}_t|| + (1 - |\text{quat}_{t+1} \cdot \text{quat}_t|) \right)$$

The term p_{move} acts as a penalty to deter unwanted movement of the target object during grasping. The metric $obj\_movement$ measures the cumulative movement of the target throughout the grasping procedure. The picking duration T is divided into regular intervals, with t denoting a specific timestep within this horizon, T. The variables tran and quat correspond to the translation and orientation of the target object at any given timestep, respectively.

6.6 A New Grasp Point Detection Model

To train a model for determining optimal grasp points, we utilized the dataset generated from the suction grasping simulation, as previously described. This dataset emulates a warehouse setting in which a suction-equipped robot is responsible for retrieving a target item from a cluttered container filled with various objects.

Given the dataset, the objective of our model is to accept specific inputs and produce a corresponding affordance map. The inputs are a point cloud representing a single-view snapshot of the container's contents and a segmentation mask pinpointing each object's location and boundaries within the container. The model's output, the affordance map, estimates the likelihood of a successful grasp for every feasible grasp point on the designated target object. In this affordance map, the highest value signifies the most desirable grasp point, represented as $(\mathbf{p}^*, \mathbf{v}^*)$, pertinent to the present scenario.

Our model's architecture, as depicted in Fig.6.2.e, integrates an auto-encoder framework. This design merges a transformer encoder for grasping information capture with a deconvolutional decoder to generate the affordance map. It's noteworthy that our data generation methodology is meticulously crafted to encapsulate the myriad uncertainties associated with real-world robotic suction grasping. Such uncertainties emerge from diverse factors, including variations in objects' physical attributes, robotic constraints, and unpredictabilities in controller behaviors.

During our experiments, we discerned that the high aleatoric uncertainty present in our dataset could potentially hinder the training process. To counter this, we utilized a specific loss function: $\mathcal{L}_{Y_{max}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \forall y_i \in Y_{max}$. Within this function, Y_{max} denotes a subset of samples. Specifically, it contains the n highest-ranked grasp points on a given object.

6.7 Evaluating Simulation Effectiveness via Grasping

Our study delves into the realm of robotic suction grasping within the context of industrial warehouse shelving [54]. As illustrated in Fig.6.1, the setup showcases a robot adjacent to an industrial shelf filled with various objects. Notably, the entrance to these shelves is on the side, posing greater challenges for suction grasping. This sideway configuration complicates the grasping process, as robotic motions might inadvertently displace items or cause them to fall. This setting, therefore, offers a rigorous testing ground for our investigations.

Detailing our system configuration: We utilized a Universal Robots UR16e robot for our

	Dyn(Full)	Dyn w/ MSR	Dyn w/o PEN	Dex	Cen
Total Success Rate	88.05%	86.75%	82.93%	81.12%	78.78%
Success Std	0.30	0.32	0.36	0.36	0.40

Table 6.1: The first row of the table displays the grasping success rate for each method, calculated from all 1300 picks. The second row provides the standard deviation of the success rate for each method across various scenarios. The first three columns of the table present an ablation comparison for our DYNAMO-GRASP (*DYN*) method, while *Dex* and *Cen* represent the DexNet and Centroid methods, respectively.

trials, outfitted with a Robotiq EPick suction gripper. Additionally, an Intel Realsense L515 camera is affixed to the robot's wrist for real-time visual feedback. Our tests incorporated a diverse set of objects, varying in shape, size, and material properties.

Our analysis compares the performance of three distinct techniques: 1. our method DYNAMO-GRASP (Dyn), 2. DexNet3.0 (Dex), and 3. the Centroid method (Cen). DexNet3.0 stands as a state-of-the-art suction grasping strategy and serves as our primary benchmark. In contrast, the Centroid method, which simply targets an object's central point for suction, has previously demonstrated its efficacy in events such as the Amazon Robotics Challenge [62, 194].

6.7.1 Large-scale, Diverse Scenario Assessment, and Ablation Test

To provide an exhaustive evaluation of the efficacy and resilience of different suction grasping techniques, we devised 260 distinct picking scenarios. These were created within the same simulation framework that was utilized for curating our training dataset. For each scenario, every method underwent five simulated suction grasps, culminating in 1,300 simulation-based evaluations for each technique. Notably, the generated scenarios encompassed a heightened degree of variability in object orientation compared to the training dataset. As a result, these scenarios often presented multifaceted object movements during the picking process.

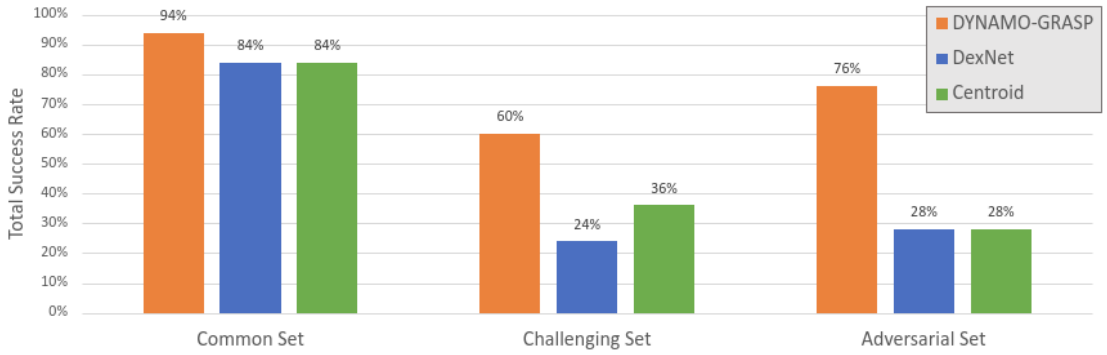

Figure 6.4: Comparison of the total success rates of different methods underscores their real-world performance on the three evaluation sets described Sec.6.7.2. The total success rate is computed by dividing the number of successful grasps by the total number of attempts within an evaluation set.

An analysis of the first, fourth, and fifth columns of Table.6.1 underscores that our proposed technique significantly outperforms both DexNet and the Centroid method in terms of success rate and stability across a myriad of scenarios. **Remarkably, our method registered an unrivaled success rate of 88.05% and demonstrated the most stable performance across diverse scenarios.** A closer look at the first three columns of Table.6.1 offers insights into an ablation study. This study elucidates the impact of individual elements of our learning approach on the effective training of our model. While *Dyn(Full)* signifies our final model iteration, *Dyn w/ MSR* denotes a model variant trained using the standard MSR loss rather than the $\mathcal{L}_{Y_{max}}$ elaborated in Sec.6.6. Lastly, *Dyn w/o PEN* eliminates the penalization term p_{move} during the labeling phase.

6.7.2 Real-world Evaluation

To evaluate performance in practical scenarios, we conducted 375 real-world suction grasps to assess the different techniques. We designed three unique scenario sets for this experiment: the *Common set*, *Challenging set*, and *Adversarial set*. Each set represents a unique challenge spectrum for suction grasping. Detailed statistics from these trials, compared with simulated

ones, are provided in Table.6.2, 6.3, and 6.4.

The Common Set: From the 260 scenarios previously mentioned in Sec.6.7.1, we selected ten and replicated them in a real-world setting using objects of analogous dimensions to the simulated ones. Each technique was tested with five grasps for each scenario. This set mirrors standard picking challenges in the given warehouse setting. As illustrated in Fig.6.4, **our model showcases the superior performance, boasting a success rate of 94%, averaging 4.7 successful grasps from five attempts with a standard deviation of 0.67**. In comparison, both DexNet and the Centroid method averaged 4.2 successful grasps, but their higher standard deviations (0.92 and 1.03 respectively) suggest more variable results.

The Challenging Set and Adversarial Set: Our focus intensifies on the tougher cases. Hence, we crafted two real-world scenario sets to rigorously test the three techniques. The *Challenging Set* includes five scenarios from the 260 detailed in Sec.6.7.1. These were chosen based on their low combined success rate in simulation, representing the toughest cases our simulation offered without human influence. The *Adversarial Set*, on the other hand, consists of five scenarios curated by a human expert, aimed at particularly challenging the grippers with everyday items not encountered during training. As seen in Fig.6.4 and Table.6.3, 6.4, **DYNAMO-GRASP significantly bests the other two techniques in both overall success rate and stable performance under tough conditions**. In the challenging set, our approach hit a success rate of 60%, while it achieved 76% in the adversarial set. Contrarily, DexNet and the Centroid method recorded success rates of 24% and 36% respectively for the challenging set, and both reached 28% in the adversarial set. Moreover, DYNAMO-GRASP consistently delivered over four successful grasps from five tries in more than half the scenarios across both sets. The competitors, on the other hand, struggled, seldom achieving even three successful grasps in the tests within these evaluation sets.

Qualitative Analysis. Figure 6.5 depicts the grasp points chosen by various methods and indicates the success of each attempt during the adversarial evaluation. The figure offers

Figure 6.5: Real-world adversarial evaluation with five grasp points for each configuration: DYNAMO GRASP (our method), DexNet, and Centroid. The color-coded points represent the suggested grasp points success and failure from various algorithms. The successfully identified grasp points are marked by the color along the label "success" and "failure".

insights into the areas chosen by each method for grasping and sheds light on which areas are more likely to lead to successful grasps. For example, in the first scenario, a tall bottle is partially propped up by a box in the back. The test checks the grasp method's awareness of potential object toppling. DYNAMO-GRASP chose the bottle's lower part, ensuring the box supported the pick. Some grasp points chosen by the other two methods were higher up on the bottle leading to toppling movements. Similarly, in scenarios two, four, and five, **DYNAMO-GRASP tends to select grasp points from regions that are overlooked by the other methods, resulting in more successful grasps in these scenarios.**

6.8 Future Work

Our work on DYNAMO-GRASP developed a simulation characterized by a reasonably realistic physics model. This software infrastructure established a robust foundation for the gamification of the warehouse manipulation problem. For future work, we propose a training strategy that utilizes this simulation for initial pretraining, followed by the use of real-world interaction data to fine-tune both task-generation and task-solving policies.

Task-solving Policy Training: In training the task-solving policy, we diverge from typical RL methods that optimize a single reward function and instead consider a range of reward functions r^g, parameterized by a task $g \in \mathcal{G}$. Each task g aligns with a set of states S^g, indicating successful suction picking of the target object by the robot. The policy's reward, $R^g = \sum_{t=0}^{T} r_t^g$, employs the function $r_t^g(\mathbf{s_t}, a_t, \mathbf{s_{t+1}}) = \mathbb{1}\{\mathbf{s_{t+1}} \in S^g\}$ to assess whether the robot has picked up the target object. Assuming a true task distribution $p_g(g)$ we aim to excel in, the optimal task-solving policy maximizes the average success probability across all tasks sampled from $p_g(g)$. It is formally represented as $\pi^*(a_t|\mathbf{s_t}, g) = \arg\max_\pi \mathbb{E}_{g\sim p_g(.)} R^g(\pi)$.

Task-generation Policy Training: To establish an autocurriculum where the task generator adaptively creates suitably challenging tasks based on the performance of the task-solving policy, we will estimate a label $y_g \in \{0, 1\}$ for all tasks g utilized in the previous training iteration. This label indicates whether $g \in GOID_i$, where $GOID_i := \{g : R_{min} \leq R^g(\pi_i) \leq R_{max} \subseteq \mathcal{G}\}$ represents the set of intermediate difficulty tasks for π_i at the i^{th} training iteration. R_{min} and R_{max} are hyperparameters that set a performance range within which we aim to concentrate on new tasks. The task-generation policy $G(z, \mathbf{s}) \rightarrow \mathbf{a_{task}}$ uses a noise vector z and the current state $\mathbf{s}$ of the objects in the container to generate a sequence of actions that transitions the container to a desired task $g \in GOID_i$. Different from the parameterized robot pushing action a, $\mathbf{a_{task}}$ is in an even simpler, predefined movements set such as left or right swipes, or triggering a device that scrambles the container. A discriminator model $D(G(z, \mathbf{s}), \mathbf{s})$ is trained to discriminate between $\mathbf{a_{task}}$ from $p_{data}(g)$ with a label $y_g = 1$ and the generated $G(z, \mathbf{s})$ conditioned on $\mathbf{s}$.

6.9 Discussion

This study addresses the intricate issue of object movement during suction grasping, a problem not yet fully solved by existing state-of-the-art techniques. We present DYNAMO-GRASP, an innovative grasp point detection method that incorporates the anticipated movement of objects to optimize the success of suction grasping. In both simulation and practical environments, DYNAMO-GRASP has shown to elevate grasping efficiency and consistency. Remarkably, in real-world trials featuring complex scenarios, our approach has demonstrated a success rate that surpasses competing methods by as much as 48%.

One limitation of our approach is the reliance on a simulated dataset that predominantly features objects with simple, regular geometries. This constraint may affect DYNAMO-GRASP's effectiveness on objects with more complex or irregular shapes. Our empirical evaluations also mainly involved objects with straightforward geometries like cubes and cylinders. Future investigations could explore developing sophisticated heuristics that amalgamate insights from DYNAMO-GRASP with those from DexNet. While DYNAMO-GRASP concentrates on the dynamics of object movement, DexNet focuses on predicting the quality of suction contact based on the geometry of the object's surface. Fusing the advantages of both could potentially yield superior results in niche-grasping scenarios.

Common Set						
	Real world experiments			sim experiments		
	DYN	Dex	Cen	DYN	Dex	Cen
Scenario 1	3	2	3	5	5	5
Scenario 2	5	4	5	5	5	5
Scenario 3	5	4	5	5	5	5
Scenario 4	5	5	5	5	5	5
Scenario 5	5	5	5	5	5	5
Scenario 6	5	5	5	5	5	5
Scenario 7	4	4	4	5	5	5
Scenario 8	5	4	4	0	5	5
Scenario 9	5	4	2	5	0	5
Scenario 10	5	5	4	5	5	5
Avg. Success Grasps	4.7	4.2	4.2	4.5	4.5	5
Std. Dev.	0.675	0.919	1.033	1.581	1.581	0
Total Success Rate	94%	84%	84%	90%	90%	100%

Table 6.2: Comparative evaluation of grasp success rates in common scenarios for three methodologies: DYNAMO-GRASP (DYN), DexNet (Dex), and Centroid (Cen). The table enumerates the average success rates, standard deviations, and total success rates for each method.

Challenging Set						
	Real world experiments			sim experiments		
	DYN	Dex	Cen	DYN	Dex	Cen
Scenario 1	4	1	2	5	0	0
Scenario 2	2	2	3	0	3	3
Scenario 3	4	1	0	5	0	0
Scenario 4	0	1	3	0	0	2
Scenario 5	5	1	1	5	0	0
Avg. Success Grasps	3	1.2	1.8	3	0.6	1
Std. Dev.	2	0.447	1.304	2.739	1.342	1.414
Total Success Rate	60%	24%	36%	60%	12%	20%

Table 6.3: Comparative evaluation of grasp success rates in challenging scenarios for three methodologies: DYNAMO-GRASP (DYN), DexNet (Dex), and Centroid (Cen). The table enumerates the average success rates, standard deviations, and total success rates for each method.

Adversarial Set			
	Real world experiments		
	DYN	Dex	Cen
Scenario 1	5	3	2
Scenario 2	3	0	0
Scenario 3	1	0	1
Scenario 4	5	3	1
Scenario 5	5	1	3
Avg. Success Grasps	3.8	1.4	1.4
Std. Dev.	1.789	1.517	1.14
Total Success Rate	76%	28%	28%

Table 6.4: Comparative evaluation of grasp success rates in adversarial scenarios for three methodologies: DYNAMO-GRASP (DYN), DexNet (Dex), and Centroid (Cen). The table enumerates the average success rates, standard deviations, and total success rates for each method.

Chapter 7

CONCLUSION

The culmination of this thesis provides a panoramic view of the progressive frontiers in robotic manipulation and Human-Robot Interaction (HRI), underscored by robust experimental research and practical implementation. The narrative that threads through the presented chapters is one of integration and forward momentum, pooling insights from benchmarks, sensor technology, deep learning, competitive HRI, multi-agent reinforcement learning, and innovative grasping techniques.

From the establishment of a versatile benchmark in robotic manipulation, designed to hone the precision of robots in sequential tasks, to the pioneering pre-touch sensor technology that elevates pose estimation, this research has fortified the foundation of intelligent robotic systems. The exploration and application of deep learning, especially in enhancing object pose estimation, paves the way for a future where robots possess not only the dexterity but also the perceptive faculties to perform complex tasks with minimal human intervention.

The venture into competitive HRI has illuminated the potential for robots to serve as compelling partners in exercise and competitive tasks, thereby enhancing human experience and engagement. The successful application of multi-agent reinforcement learning in these contexts illustrates a symbiotic evolution of the capabilities of both competing players. We have also observed humans learning to adapt and respond to robot strategies; however, the increasingly sophisticated policies of robots make this adaptation less conspicuous. This presents significant research potential in utilizing these characteristics to facilitate human learning in sports, physical therapy, and other interactive learning settings.

The introduction of the Stackelberg-MADDPG algorithm in asymmetric, competitive games via multi-agent reinforcement learning (MARL) marks a significant advance, balancing

the dynamics between agents to ensure equitable and effective training outcomes. This approach acknowledges the complexities of real-world interactions, where imbalances may occur, and seeks to offer a solution that could be extrapolated to broader applications.

Finally, the development of DYNAMO-GRASP addresses the nuanced challenge of object movement during suction grasping, demonstrating a marked improvement in the efficiency of robotic picking. By integrating dynamic predictions into the grasping process, this methodology showcases the substantial benefits of considering object movement in real-time applications, charting a course for future enhancements that may integrate geometric analysis for even more refined results.

In conclusion, this thesis encapsulates the vibrant and intricate tapestry of modern robotics, where interdisciplinary methodologies converge to advance the state of the art. The research outcomes not only highlight the successes but also chart the potential directions for future inquiries, indicating an ecosystem ripe for innovation. In essence, the advancements presented herein do not signify an end but rather a springboard into a realm where robotic intelligence, interaction, and manipulation are seamlessly interwoven into the fabric of technological progress and human experience.

BIBLIOGRAPHY

[1] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[2] 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. *arXiv preprint arXiv:1808.00177*, 2018.

[3] Emanuele Antonioni, Vincenzo Suriani, Francesco Riccio, and Daniele Nardi. Game strategies for physical robot soccer players: a survey. *IEEE Transactions on Games*, 13(4):342–357, 2021.

[4] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528*, 2019.

[5] Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. Emergent complexity via multi-agent competition. *arXiv preprint arXiv:1710.03748*, 2017.

[6] Tamer Başar and Geert Jan Olsder. *Dynamic noncooperative game theory*. SIAM, 1998.

[7] CPM Special Bearings. Abb robots katana fight, 2013. URL https://www.youtube.com/watch?v=cR-YlZ9NdIA.

[8] Sven Behnke. Robot competitions-ideal benchmarks for robotics research. In *Proc. of*

IROS-2006 Workshop on Benchmarks in Robotics Research. Institute of Electrical and Electronics Engineers (IEEE), 2006.

[9] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[10] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[11] Tapomayukh Bhattacharjee, Ethan K Gordon, Rosario Scalise, Maria E Cabrera, Anat Caspi, Maya Cakmak, and Siddhartha S Srinivasa. Is more autonomy always better? exploring preferences of users with mobility impairments in robot-assisted feeding. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pages 181–190, 2020.

[12] C Blanes, M Mellado, Coral Ortiz, and A Valera. Technologies for robot grippers in pick and place operations for fresh fruits and vegetables. *Spanish Journal of Agricultural Research*, 9(4):1130–1141, 2011.

[13] Andreea Bobu, Dexter RR Scobee, Jaime F Fisac, S Shankar Sastry, and Anca D Dragan. Less is more: Rethinking probabilistic models of human behavior. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pages 429–437, 2020.

[14] Andrea Bonarini, Serenella Besio, et al. *Robot Play for All: Developing Toys and Games for Disability*. Springer, 2022.

[15] Adam Bry and Nicholas Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *2011 IEEE international conference on robotics and automation*, pages 723–730. IEEE, 2011.

[16] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

[17] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set. *IEEE Robotics & Automation Magazine*, 22(3):36–52, 2015.

[18] Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57–83, 2002.

[19] Hanwen Cao, Hao-Shu Fang, Wenhai Liu, and Cewu Lu. Suctionnet-1billion: A large-scale benchmark for suction grasping. *IEEE Robotics and Automation Letters*, 6(4): 8718–8725, 2021.

[20] Ly-Yu Chang, Joshua R Smith, and Dieter Fox. Interactive singulation of objects from a pile. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3875–3882. IEEE, 2012.

[21] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

[22] Min Chen, Stefanos Nikolaidis, Harold Soh, David Hsu, and Siddhartha Srinivasa. Trust-aware decision making for human-robot collaboration: Model learning and planning. *ACM Transactions on Human-Robot Interaction (THRI)*, 9(2):1–23, 2020.

[23] Tiffany L Chen, Tapomayukh Bhattacharjee, Jenay M Beer, Lena H Ting, Madeleine E Hackney, Wendy A Rogers, and Charles C Kemp. Older adults' acceptance of a robot for partner dance-based exercise. *PloS one*, 12(10):e0182736, 2017.

[24] FB Christiansen and V Loeschcke. Evolution and competition. In *Population biology*, pages 367–394. Springer, 1990.

[25] Ping Yong Chua, T Ilschner, and Darwin G Caldwell. Robotic manipulation of food products–a review. *Industrial Robot: An International Journal*, 30(4):345–354, 2003.

[26] Alexandre Coninx, Paul Baxter, Elettra Oleari, Sara Bellini, Bert Bierman, O Henkemans, Lola Cañamero, Piero Cosi, Valentin Enescu, R Espinoza, et al. Towards long-term social child-robot interaction: using multi-activity switching to engage young users. *Journal of Human-Robot Interaction*, 2016.

[27] Árpád Csathó and Béla Birkás. Early-life stressors, personality development, and fast life strategies: An evolutionary perspective on malevolent personality features. *Frontiers in psychology*, 9:305, 2018.

[28] Nikhil Chavan Dafle, Alberto Rodriguez, Robert Paolini, Bowei Tang, Siddhartha S Srinivasa, Michael Erdmann, Matthew T Mason, Ivan Lundberg, Harald Staab, and Thomas Fuhlbrigge. Extrinsic dexterity: In-hand manipulation with external forces. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1578–1585. IEEE, 2014.

[29] Eric Deng, Bilge Mutlu, Maja J Mataric, et al. Embodiment in socially interactive robots. *Foundations and Trends® in Robotics*, 7(4):251–356, 2019.

[30] Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. *arXiv preprint arXiv:2012.02096*, 2020.

[31] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International conference on machine learning*, pages 1329–1338. PMLR, 2016.

[32] Jeffrey H Dyer and Harbir Singh. The relational view: Cooperative strategy and sources of interorganizational competitive advantage. *Academy of management review*, 23(4):660–679, 1998.

[33] Aaron Edsinger and Charles C Kemp. Human-robot interaction for cooperative manipulation: Handing objects to one another. In *RO-MAN 2007-The 16th IEEE International Symposium on Robot and Human Interactive Communication*, pages 1167–1172. IEEE, 2007.

[34] Clemens Eppner, Sebastian Höfer, Rico Jonschkowski, Roberto Martín-Martín, Arne Sieverling, Vincent Wall, and Oliver Brock. Lessons from the amazon picking challenge: Four aspects of building robotic systems. In *Robotics: science and systems*, pages 4831–4835, 2016.

[35] Juan Fasola and Maja J Mataric. Robot exercise instructor: A socially assistive robot system to monitor and encourage physical exercise for the elderly. In *19th International Symposium in Robot and Human Interactive Communication*, pages 416–421. IEEE, 2010.

[36] Nima Fazeli, Miquel Oller, Jiajun Wu, Zheng Wu, Joshua B Tenenbaum, and Alberto Rodriguez. See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion. *Science Robotics*, 4(26):eaav3123, 2019.

[37] Deborah L Feltz, Samuel T Forlenza, Brian Winn, and Norbert L Kerr. Cyber buddy is better than no buddy: A test of the köhler motivation effect in exergames. *GAMES FOR HEALTH: Research, Development, and Clinical Applications*, 3(2):98–105, 2014.

[38] Tanner Fiez, Benjamin Chasnov, and Lillian J Ratliff. Implicit learning dynamics in stackelberg games: Equilibria characterization, convergence analysis, and empirical study. In *International Conference on Machine Learning*, 2020.

[39] Jerzy Filar and Koos Vrieze. *Competitive Markov decision processes*. Springer Science & Business Media, 2012.

[40] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion.

In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017.

[41] Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *Conference on robot learning*, pages 482–495. PMLR, 2017.

[42] Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *International conference on machine learning*, pages 1515–1528. PMLR, 2018.

[43] Jakob Foerster, Gregory Farquhar, Maruan Al-Shedivat, Tim Rocktäschel, Eric Xing, and Shimon Whiteson. DiCE: The infinitely differentiable Monte Carlo estimator. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1524–1533, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/foerster18a.html.

[44] Jakob N Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*, 2017.

[45] Masahiro Fujita and Koji Kageyama. An open architecture for robot entertainment. In *Proceedings of the First International Conference on Autonomous Agents*, AGENTS '97, page 435–442, New York, NY, USA, 1997. Association for Computing Machinery. ISBN 0897918770. doi: 10.1145/267658.267764. URL https://doi.org/10.1145/267658.267764.

[46] Masahiro Fujita, Hiroaki Kitano, and T Doi. Robot entertainment. *Robots for kids: Exploring new technologies for learning*, pages 37–72, 2000.

[47] Eric Ghysels, Pedro Santa-Clara, and Rossen Valkanov. There is a risk-return trade-off after all. *Journal of Financial Economics*, 76(3):509–548, 2005.

[48] Kieran Gilday, James Lilley, and Fumiya Iida. Suction cup based on particle jamming and its performance comparison in various fruit handling tasks. In *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 607–612. IEEE, 2020.

[49] Diane L Gill, Lavon Williams, Deborah A Dowd, Christina M Beaudoin, and Jeffrey J Martin. Competitive orientations and motives of adult sport and exercise participants. 1996.

[50] Ross Girshick. py-faster-rcnn. https://github.com/rbgirshick/py-faster-rcnn, 2015.

[51] Christopher M Glaze, Alexandre LS Filipowicz, Joseph W Kable, Vijay Balasubramanian, and Joshua I Gold. A bias–variance trade-off governs individual differences in on-line learning in an unpredictable environment. *Nature Human Behaviour*, 2(3): 213–224, 2018.

[52] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680, 2014.

[53] Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *international conference on machine learning*, pages 1311–1320. PMLR, 2017.

[54] Markus Grotz, Soofiyan Atar, Yi Li, Paolo Torrado, Boling Yang, Nick Walker, Michael Murray, Maya Cakmak, and Joshua R. Smith. Towards robustly picking unseen objects from densely packed shelves. In *RSS Workshop on Perception and Manipulation Challenges for Warehouse Automation*, 2023.

[55] Di Guo, Patrick Lancaster, Liang-Ting Jiang, Fuchun Sun, and Joshua R Smith. Transmissive optical pretouch sensing for robotic grasping. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 5891–5897. IEEE, 2015.

[56] Tuomas Haarnoja, Ben Moran, Guy Lever, Sandy H Huang, Dhruva Tirumala, Markus Wulfmeier, Jan Humplik, Saran Tunyasuvunakool, Noah Y Siegel, Roland Hafner, et al. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *arXiv preprint arXiv:2304.13653*, 2023.

[57] Sandra G Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pages 904–908. Sage publications Sage CA: Los Angeles, CA, 2006.

[58] Shun Hasegawa, Kentaro Wada, Kei Okada, and Masayuki Inaba. A three-fingered hand with a suction gripping system for warehouse automation. *Journal of Robotics and Mechatronics*, 31(2):289–304, 2019.

[59] He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. Opponent modeling in deep reinforcement learning. In *International conference on machine learning*, pages 1804–1813, 2016.

[60] IT Heazlewood, Joe Walsh, Mike Climstein, Stephen Burke, Jyrki Kettunen, KJ Adams, and Mark DeBeliso. Sport psychological constructs related to participation in the 2009 world masters games. *World Academy of Science, Engineering and Technology*, 7:2027–2032, 2011.

[61] Nicolas Heess, Dhruva TB, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, SM Eslami, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

[62] Carlos Hernandez, Mukunda Bharatheesha, Wilson Ko, Hans Gaiser, Jethro Tan, Kanter van Deurzen, Maarten de Vries, Bas Van Mil, Jeff van Egmond, Ruben Burger, et al.

Team delft's robot winner of the amazon picking challenge 2016. In *RoboCup 2016: Robot World Cup XX 20*, pages 613–624. Springer, 2017.

[63] Suzanne Hidi and K Ann Renninger. The four-phase model of interest development. *Educational psychologist*, 41(2):111–127, 2006.

[64] Ryosuke Higo, Yuji Yamakawa, Taku Senoo, and Masatoshi Ishikawa. Rubik's cube handling using a high-speed multi-fingered hand and a high-speed vision system. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6609–6614. IEEE, 2018.

[65] W Daniel Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D: Nonlinear Phenomena*, 42(1-3):228–234, 1990.

[66] Dirk Holz, Alexandru E Ichim, Federico Tombari, Radu B Rusu, and Sven Behnke. Registration with the point cloud library: a modular framework for aligning in 3-d. *IEEE Robotics & Automation Magazine*, 22(4):110–124, 2015.

[67] Kaijen Hsiao, Paul Nangeroni, Manfred Huber, Ashutosh Saxena, and Andrew Y Ng. Reactive grasping using optical proximity sensors. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2098–2105. IEEE, 2009.

[68] Baichuan Huang, Shuai D. Han, Jingjin Yu, and Abdeslam Boularias. Visual foresight trees for object retrieval from clutter with nonprehensile rearrangement. *IEEE Robotics and Automation Letters*, 7(1):231–238, 2022. doi: 10.1109/LRA.2021.3123373.

[69] Tae Myung Huh, Kate Sanders, Michael Danielczuk, Monica Li, Yunliang Chen, Ken Goldberg, and Hannah S Stuart. A multi-chamber smart suction cup for adaptive gripping and haptic exploration. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1786–1793. IEEE, 2021.

[70] Iaroslav Ispolatov, Evgeniia Alekseeva, and Michael Doebeli. Competition-driven evolution of organismal complexity. *PLoS computational biology*, 15(10):e1007388, 2019.

[71] Seokhwan Jeong, Phillip Tran, and Jaydev P Desai. Integration of self-sealing suction cups on the flexotendon glove-ii robotic exoskeleton system. *IEEE Robotics and Automation Letters*, 5(2):867–874, 2020.

[72] Liang-Ting Jiang and Joshua R Smith. Seashell effect pretouch sensing for robotic grasping. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2851–2858. IEEE, 2012.

[73] Ping Jiang, Junji Oaki, Yoshiyuki Ishihara, Junichiro Ooga, Haifeng Han, Atsushi Sugahara, Seiji Tokura, Haruna Eto, Kazuma Komoda, and Akihito Ogawa. Learning suction graspability considering grasp quality and robot reachability for bin-picking. *Frontiers in Neurorobotics*, 16, 2022.

[74] Yun Jiang, Stephen Moseson, and Ashutosh Saxena. Efficient grasping from rgbd images: Learning using a new rectangle representation. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3304–3311. IEEE, 2011.

[75] Li-Hong Juang. Humanoid robots play chess using visual control. *Multimedia Tools and Applications*, pages 1–22, 2022.

[76] Gregory Kahn, Peter Sujan, Sachin Patil, Shaunak Bopardikar, Julian Ryde, Ken Goldberg, and Pieter Abbeel. Active exploration using trajectory optimization for robotic grasping in the presence of occlusions. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4783–4790. IEEE, 2015.

[77] Liyiming Ke, Ajinkya Kamat, Jingqiang Wang, Tapomayukh Bhattacharjee, Christoforos Mavrogiannis, and Siddhartha S Srinivasa. Telemanipulation with chopsticks: Analyzing human factors in user demonstrations. *arXiv preprint arXiv:2008.00101*, 2020.

[78] Dong-Ki Kim, Miao Liu, Shayegan Omidshafiei, Sebastian Lopez-Cot, Matthew Riemer, Golnaz Habibi, Gerald Tesauro, Sami Mourad, Murray Campbell, and

Jonathan P How. Learning hierarchical teaching policies for cooperative agents. *arXiv preprint arXiv:1903.03216*, 2019.

[79] Ramesh Kolluru, Kimon P Valavanis, and Timothy M Hebert. Modeling, analysis, and performance evaluation of a robotic gripper system for limp material handling. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28(3):480–486, 1998.

[80] Paweł Kołosowski, Adam Wolniakowski, and Kanstantsin Miatliuk. Collaborative robot system for playing chess. In *2020 International Conference Mechatronic Systems and Materials (MSM)*, pages 1–6. IEEE, 2020.

[81] Hatice Kose-Bagci, Ester Ferrari, Kerstin Dautenhahn, Dag Sverre Syrdal, and Chrystopher L Nehaniv. Effects of embodiment and gestures on social interaction in drumming games with a humanoid robot. *Advanced Robotics*, 23(14):1951–1996, 2009.

[82] Danica Kragic, Henrik I Christensen, et al. Survey on visual servoing for manipulation. *Computational Vision and Active Perception Laboratory, Fiskartorpsv*, 15, 2002.

[83] Steven George Krantz and Harold R Parks. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2002.

[84] Alap Kshirsagar, Bnaya Dreyfuss, Guy Ishai, Ori Heffetz, and Guy Hoffman. Monetary-incentive competition between humans and robots: experimental results. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 95–103. IEEE, 2019.

[85] Hikaru Kumamoto, Naoki Shirakura, Jun Takamatsu, and Tsukasa Ogasawara. Underwater suction gripper for object manipulation with an underwater robot. In *2021 IEEE International Conference on Mechatronics (ICM)*, pages 1–7. IEEE, 2021.

[86] 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.

[87] Patrick Lancaster, Boling Yang, and Joshua R Smith. Improved object pose estimation via deep pre-touch sensing. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2448–2455. IEEE, 2017.

[88] Barbara Landau, Linda B Smith, and Susan S Jones. The importance of shape in early lexical learning. *Cognitive development*, 3(3):299–321, 1988.

[89] Adam Leeper, Kaijen Hsiao, Eric Chu, and J Kenneth Salisbury. Using near-field stereo vision for robotic grasping in cluttered environments. In *Experimental Robotics*, pages 253–267. Springer, 2014.

[90] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

[91] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *arXiv preprint arXiv:1603.02199*, 2016.

[92] Rui Li, Robert Platt, Wenzhen Yuan, Andreas ten Pas, Nathan Roscup, Mandayam A Srinivasan, and Edward Adelson. Localization and manipulation of small parts using gelsight tactile sensing. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3988–3993. IEEE, 2014.

[93] Rui Li, Marc van Almkerk, Sanne van Waveren, Elizabeth Carter, and Iolanda Leite. Comparing human-robot proxemics between virtual reality and the real world. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 431–439. IEEE, 2019.

[94] 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.

[95] Tianyi Lin, Chi Jin, and Michael I Jordan. Near-optimal algorithms for minimax optimization. In *Conference on Learning Theory*, pages 2738–2779. PMLR, 2020.

[96] Vincenzo Lippiello, Bruno Siciliano, and Luigi Villani. Position-based visual servoing in industrial multirobot cells using a hybrid camera configuration. *IEEE Transactions on Robotics*, 23(1):73–86, 2007.

[97] George Liset. Sensory motor learning: Developing a kinaesthetic sense in the throws. *New Studies in Athletics*, 21(1):51, 2006.

[98] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

[99] Siqi Liu, Guy Lever, Josh Merel, Saran Tunyasuvunakool, Nicolas Heess, and Thore Graepel. Emergent coordination through competition. *arXiv preprint arXiv:1902.07151*, 2019.

[100] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*, pages 6379–6390, 2017.

[101] Ruikun Luo, Rafi Hayne, and Dmitry Berenson. Unsupervised early prediction of human reaching for human–robot collaboration in shared workspaces. *Autonomous Robots*, 42(3):631–648, 2018.

[102] Kevin M Lynch and Matthew T Mason. Stable pushing: Mechanics, controllability, and planning. *The international journal of robotics research*, 15(6):533–556, 1996.

[103] Raymond R Ma and Aaron M Dollar. On dexterity and dexterous manipulation. In *2011 15th International Conference on Advanced Robotics (ICAR)*, pages 1–7. IEEE, 2011.

[104] Patrick MacAlpine, Daniel Urieli, Samuel Barrett, Shivaram Kalyanakrishnan, Francisco Barrera, Adrian Lopez-Mobilia, Nicolae Stiurca, Victor Vu, and Peter Stone. Ut austin villa 2011: a champion agent in the robocup 3d soccer simulation competition. In *AAMAS*, pages 129–136, 2012.

[105] Jeffrey Mahler, Matthew Matl, Xinyu Liu, Albert Li, David Gealy, and Ken Goldberg. Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning. In *2018 IEEE International Conference on robotics and automation (ICRA)*, pages 5620–5627. IEEE, 2018.

[106] Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, and Ken Goldberg. Learning ambidextrous robot grasping policies. *Science Robotics*, 4(26):eaau4984, 2019.

[107] Jeremy Maitin-Shepard, Marco Cusumano-Towner, Jinna Lei, and Pieter Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2308–2315. IEEE, 2010.

[108] Alexis Maldonado, Humberto Alvarez, and Michael Beetz. Improving robot manipulation through fingertip perception. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2947–2954. IEEE, 2012.

[109] Ezio Malis and Selim Benhimane. A unified approach to visual tracking and servoing. *Robotics and Autonomous Systems*, 52(1):39–52, 2005.

[110] James Martens et al. Deep learning via hessian-free optimization. In *ICML*, volume 27, pages 735–742, 2010.

[111] Francisco Martinez-Gil, Miguel Lozano, and Fernando Fernandez. Emergent behaviors and scalability for multi-agent reinforcement learning-based pedestrian models. *Simulation Modelling Practice and Theory*, 74:117–133, 2017.

[112] Matthew T Mason. Mechanics and planning of manipulator pushing operations. *The International Journal of Robotics Research*, 5(3):53–71, 1986.

[113] Maja J Matarić, Jon Eriksson, David J Feil-Seifer, and Carolee J Winstein. Socially assistive robotics for post-stroke rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 4(1):1–9, 2007.

[114] Elias Matsas and George-Christopher Vosniakos. Design of a virtual reality training system for human–robot collaboration in manufacturing tasks. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 11(2):139–153, 2017.

[115] Brian Mayton, Louis LeGrand, and Joshua R Smith. An electric field pretouch system for grasping and co-manipulation. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 831–838. IEEE, 2010.

[116] Eric V Mazumdar, Michael I Jordan, and S Shankar Sastry. On finding local nash equilibria (and only local nash equilibria) in zero-sum games. *arXiv preprint arXiv:1901.00838*, 2019.

[117] Barbara Mazzolai, Alessio Mondini, Francesca Tramacere, Gianluca Riccomi, Ali Sadeghi, Goffredo Giordano, Emanuela Del Dottore, Massimiliano Scaccia, Massimo Zampato, and Stefano Carminati. Octopus-inspired soft arm with suction cups for enhanced grasping tasks in confined environments. *Advanced Intelligent Systems*, 1(6): 1900041, 2019.

[118] Panayotis Mertikopoulos, Christos Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2703–2717. SIAM, 2018.

[119] R Morales, FJ Badesa, N Garcia-Aracil, JM Sabater, and L Zollo. Soft robotic manipulation of onions and artichokes in the food industry. *Advances in Mechanical Engineering*, 6:345291, 2014.

[120] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2901–2910, 2019.

[121] Stephan Muhlbacher-Karrer, Andre Gaschler, and Hubert Zangl. Responsive fingers—capacitive sensing during object manipulation. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 4394–4401. IEEE, 2015.

[122] John Edison Munoz and Kerstin Dautenhahn. Robo ludens: A game design taxonomy for multiplayer games using socially interactive robots. *ACM Transactions on Human-Robot Interaction (THRI)*, 10(4):1–28, 2021.

[123] Peter Muris, Harald Merckelbach, Henry Otgaar, and Ewout Meijer. The malevolent side of human nature: A meta-analysis and critical review of the literature on the dark triad (narcissism, machiavellianism, and psychopathy). *Perspectives on Psychological Science*, 12(2):183–204, 2017.

[124] Alan Murta. A general polygon clipping library. *Advanced Interfaces Group, Department of Computer Science, University of Manchester, Manchester, UK*, 2000.

[125] Bilge Mutlu, Steven Osman, Jodi Forlizzi, Jessica Hodgins, and Sara Kiesler. Perceptions of asimo: an exploration on co-operation and competition with humans and humanoid robots. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 351–352, 2006.

[126] Jared Nakahara, Boling Yang, and Joshua R Smith. Contact-less manipulation of millimeter-scale objects via ultrasonic levitation. In *2020 8th IEEE RAS/EMBS In-*

ternational Conference for Biomedical Robotics and Biomechatronics (BioRob), pages 264–271. IEEE, 2020.

[127] Stefanos Nikolaidis, Swaprava Nath, Ariel D Procaccia, and Siddhartha Srinivasa. Game-theoretic modeling of human adaptation in human-robot collaboration. In *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction*, pages 323–331, 2017.

[128] Allison M Okamura, Niels Smaby, and Mark R Cutkosky. An overview of dexterous manipulation. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 1, pages 255–262. IEEE, 2000.

[129] Albert S Olesen, Benedek B Gergaly, Emil A Ryberg, Mads R Thomsen, and Dimitrios Chrysostomou. A collaborative robot cell for random bin-picking based on deep learning policies and a multi-gripper switching strategy. *Procedia Manufacturing*, 51: 3–10, 2020.

[130] Sylvie CW Ong, Shao Wei Png, David Hsu, and Wee Sun Lee. Planning under uncertainty for robotic tasks with mixed observability. *The International Journal of Robotics Research*, 29(8):1053–1068, 2010.

[131] OpenAI OpenAI, Matthias Plappert, Raul Sampedro, Tao Xu, Ilge Akkaya, Vineet Kosaraju, Peter Welinder, Ruben D'Sa, Arthur Petron, Henrique P d O Pinto, et al. Asymmetric self-play for automatic goal discovery in robotic manipulation. *arXiv preprint arXiv:2101.04882*, 2021.

[132] Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous agents and multi-agent systems*, 11(3):387–434, 2005.

[133] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven

exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[134] Barak A Pearlmutter. Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160, 1994.

[135] Anna Petrovskaya and Oussama Khatib. Global localization of objects via touch. *Robotics, IEEE Transactions on*, 27(3):569–585, 2011.

[136] Jeff M Phillips, Ran Liu, and Carlo Tomasi. Outlier robust icp for minimizing fractional rmsd. In *3-D Digital Imaging and Modeling, 2007. 3DIM'07. Sixth International Conference on*, pages 427–434. IEEE, 2007.

[137] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 3406–3413. IEEE, 2016.

[138] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. *arXiv preprint arXiv:1703.02702*, 2017.

[139] Jan L Plass, Paul A O'Keefe, Bruce D Homer, Jennifer Case, Elizabeth O Hayward, Murphy Stein, and Ken Perlin. The impact of individual, competitive, and collaborative mathematics game play on learning, performance, and motivation. *Journal of educational psychology*, 105(4):1050, 2013.

[140] Iago Portela-Pino, Antonio López-Castedo, María José Martínez-Patiño, Teresa Valverde-Esteve, and José Domínguez-Alonso. Gender differences in motivation and barriers for the practice of physical exercise in adolescence. *International journal of environmental research and public health*, 17(1):168, 2020.

[141] Vijay Pradeep, Kurt Konolige, and Eric Berger. Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach. In *Experimental Robotics: The 12th International Symposium on Experimental Robotics*, pages 211–225. Springer, 2014.

[142] Manish Prajapat, Kamyar Azizzadenesheli, Alexander Liniger, Yisong Yue, and Anima Anandkumar. Competitive policy optimization. *arXiv preprint arXiv:2006.10611*, 2020.

[143] Xavier Provot et al. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface*, pages 147–147. Canadian Information Processing Society, 1995.

[144] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32, 2019.

[145] Aravind Rajeswaran, Igor Mordatch, and Vikash Kumar. A game theoretic framework for model based reinforcement learning. In *International conference on machine learning*, pages 7953–7963. PMLR, 2020.

[146] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1316–1322. IEEE, 2015.

[147] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[148] Lorenzo Riano. pr2 rubiks solver, 2011. URL https://github.com/uu-isrc-robotics/pr2_rubiks_solver.

[149] Tomas Rokicki, Herbert Kociemba, Morley Davidson, and John Dethridge. The diameter of the rubik's cube group is twenty. *SIAM Review*, 56(4):645–670, 2014.

[150] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE, 2011.

[151] Katie Salen, Katie Salen Tekinbaş, and Eric Zimmerman. *Rules of play: Game design fundamentals*. MIT press, 2004.

[152] Kosuke Sato, Keita Watanabe, Shuichi Mizuno, Masayoshi Manabe, Hiroaki Yano, and Hiroo Iwata. Development of a block machine for volleyball attack training. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1036–1041. IEEE, 2017.

[153] Jonathan Schaeffer, Neil Burch, Yngvi Björnsson, Akihiro Kishimoto, Martin Müller, Robert Lake, Paul Lu, and Steve Sutphen. Checkers is solved. *science*, 317(5844): 1518–1522, 2007.

[154] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[155] Max Schwarz, Anton Milan, Christian Lenz, Aura Munoz, Arul Selvam Periyasamy, Michael Schreiber, Sebastian Schüller, and Sven Behnke. Nimbro picking: Versatile part handling for warehouse automation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3032–3039. IEEE, 2017.

[156] Quanquan Shao, Jie Hu, Weiming Wang, Yi Fang, Wenhai Liu, Jin Qi, and Jin Ma. Suction grasp region prediction using self-supervised learning for object picking in dense clutter. In *2019 IEEE 5th International Conference on Mechatronics System and Robots (ICMSR)*, pages 7–12. IEEE, 2019.

[157] Haobin Shi, Gang Sun, Yuanpeng Wang, and Kao-Shing Hwang. Adaptive image-based visual servoing with temporary loss of the visual signal. *IEEE Transactions on Industrial Informatics*, 15(4):1956–1965, 2018.

[158] Elaine Short, Justin Hart, Michelle Vu, and Brian Scassellati. No fair!! an interaction with a cheating robot. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 219–226. IEEE, 2010.

[159] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[160] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[161] Kenneth O Stanley and Risto Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of artificial intelligence research*, 21:63–100, 2004.

[162] Bastian Steder, Radu Bogdan Rusu, Kurt Konolige, and Wolfram Burgard. Point feature extraction on 3d range scans taking into account object boundaries. In *Robotics and automation (icra), 2011 ieee international conference on*, pages 2601–2608. IEEE, 2011.

[163] Hannah S Stuart, Matteo Bagheri, Shiquan Wang, Heather Barnard, Audrey L Sheng, Merritt Jenkins, and Mark R Cutkosky. Suction helps in a pinch: Improving underwater manipulation with gentle suction flow. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 2279–2284. IEEE, 2015.

[164] Sainbayar Sukhbaatar, Zeming Lin, Ilya Kostrikov, Gabriel Synnaeve, Arthur Szlam, and Rob Fergus. Intrinsic motivation and automatic curricula via asymmetric self-play. *arXiv preprint arXiv:1703.05407*, 2017.

[165] Luise Süssenbach, Nina Riether, Sebastian Schneider, Ingmar Berger, Franz Kummert, Ingo Lütkebohle, and Karola Pitsch. A robot as fitness companion: towards an interactive action-based motivation model. In *The 23rd IEEE international symposium on robot and human interactive communication*, pages 286–293. IEEE, 2014.

[166] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. Multiagent cooperation and competition with deep reinforcement learning. *PloS one*, 12(4):e0172395, 2017.

[167] Jeffrey Trawick-Smith. The physical play and motor development of young children: A review of literature and implications for practice. *Center for Early Childhood Education, Eastern Connecticut State University*, 2014.

[168] Karl Tuyls, Julien Perolat, Marc Lanctot, Joel Z Leibo, and Thore Graepel. A generalised method for empirical game theoretic analysis. *arXiv preprint arXiv:1803.06376*, 2018.

[169] Stefan Ulbrich, Daniel Kappler, Tamim Asfour, Nikolaus Vahrenkamp, Alexander Bierbaum, Markus Przybylski, and Rüdiger Dillmann. The opengrasp benchmarking suite: An environment for the comparative analysis of grasping and dexterous manipulation. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1761–1767. IEEE, 2011.

[170] Nikolaus Vahrenkamp, Steven Wieland, Pedram Azad, David Gonzalez, Tamim Asfour, and Riidiger Dillmann. Visual servoing for humanoid grasping and manipulation tasks. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 406–412. IEEE, 2008.

[171] Jur Van Den Berg, Sachin Patil, and Ron Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *The International Journal of Robotics Research*, 31(11):1263–1278, 2012.

[172] Johanna H Van der Lee, Robert C Wagenaar, Gustaaf J Lankhorst, Tanneke W Vogelaar, Walter L Devillé, and Lex M Bouter. Forced use of the upper extremity in chronic stroke patients: results from a single-blind randomized clinical trial. *Stroke*, 30(11):2369–2375, 1999.

[173] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[174] M Viru, Anthony Hackney, K Karelson, T Janson, M Kuus, and A Viru. Competition effects on physiological responses to exercise: performance, cardiorespiratory and hormonal factors. *Acta Physiologica Hungarica*, 97(1):22–30, 2010.

[175] Heinrich Von Stackelberg. *Market structure and equilibrium*. Springer Science & Business Media, 2010.

[176] Li Wang, Yushu Liu, Hongbin Deng, and Yuanqing Xu. Obstacle-avoidance path planning for soccer robots using particle swarm optimization. In *2006 IEEE International Conference on Robotics and Biomimetics*, pages 1233–1238. IEEE, 2006.

[177] Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O Stanley. Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions. *arXiv preprint arXiv:1901.01753*, 2019.

[178] Rui Wang, Yinglong Miao, and Kostas E Bekris. Efficient and high-quality prehensile rearrangement in cluttered and confined spaces. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1968–1975. IEEE, 2022.

[179] Thomas Wisspeintner, Tijn Van Der Zant, Luca Iocchi, and Stefan Schiffer. Robocup@ home: Scientific competition and benchmarking for domestic service robots. *Interaction Studies*, 10(3):392–426, 2009.

[180] Ryan Wistort and Joshua R Smith. Electric field servoing for robotic manipulation. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 494–499. IEEE, 2008.

[181] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[182] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Control strategies for physically simulated characters performing two-player competitive sports. *ACM Transactions on Graphics (TOG)*, 40(4):1–11, 2021.

[183] Hongtao Wu, Jikai Ye, Xin Meng, Chris Paxton, and Gregory S Chirikjian. Transporters with visual foresight for solving unseen rearrangement tasks. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10756–10763. IEEE, 2022.

[184] Boling Yang, Patrick Lancaster, and Joshua R Smith. Pre-touch sensing for sequential manipulation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5088–5095. IEEE, 2017.

[185] Boling Yang, Patrick E Lancaster, Siddhartha S Srinivasa, and Joshua R Smith. Benchmarking robot manipulation with the rubik's cube. *IEEE Robotics and Automation Letters*, 5(2):2094–2099, 2020.

[186] Boling Yang, Golnaz Habibi, Patrick Lancaster, Byron Boots, and Joshua Smith. Motivating physical activity via competitive human-robot interaction. In *5th Annual Conference on Robot Learning*, 2021.

[187] Boling Yang, Xiangyu Xie, Golnaz Habibi, and Joshua R Smith. Competitive physical human-robot game play. In *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, pages 242–246, 2021.

[188] Boling Yang, Golnaz Habibi, Patrick Lancaster, Byron Boots, and Joshua Smith. Motivating physical activity via competitive human-robot interaction. In *Conference on Robot Learning*, pages 839–849. PMLR, 2022.

[189] Boling Yang, Soofiyan Layakalli Atar, Markus Grotz, Byron Boots, and Joshua Smith. Dynamo-grasp: Dynamics-aware optimization for grasp point detection in suction grippers. In *7th Annual Conference on Robot Learning*, 2023.

[190] Boling Yang, Liyuan Zheng, Lillian J Ratliff, Byron Boots, and Joshua R Smith. Stackelberg games for learning emergent behaviors during competitive autocurricula. *arXiv preprint arXiv:2305.03735*, 2023.

[191] Manman Yang, Leijian Yu, Cuebong Wong, Carmelo Mineo, Erfu Yang, Iain Bomphray, and Ruoyu Huang. A cooperative mobile robot and manipulator system (comrms) for transport and lay-up of fibre plies in modern composite material manufacture. *The International Journal of Advanced Manufacturing Technology*, pages 1–17, 2021.

[192] Yaodong Yang and Jun Wang. An overview of multi-agent reinforcement learning from game theoretical perspective. *arXiv preprint arXiv:2011.00583*, 2020.

[193] Yasuyoshi Yokokohji, Yukihiro Iida, and Tsuneo Yoshikawa. 'toy problem'as the benchmark test for teleoperation systems. *Advanced Robotics*, 17(3):253–273, 2003.

[194] Kuan-Ting Yu, Nima Fazeli, Nikhil Chavan-Dafle, Orion Taylor, Elliott Donlon, Guillermo Diaz Lankenau, and Alberto Rodriguez. A summary of team mit's approach to the amazon picking challenge 2015. *arXiv preprint arXiv:1604.03639*, 2016.

[195] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

[196] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, et al. Robotic pick-and-

place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. *The International Journal of Robotics Research*, 41(7):690–705, 2022.

[197] Nan Zeng, Mohammad Ayyub, Haichun Sun, Xu Wen, Ping Xiang, Zan Gao, et al. Effects of physical activity on motor skills and cognitive development in early childhood: a systematic review. *BioMed research international*, 2017, 2017.

[198] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635*, 2019.

[199] Tongjia Zhang, Chengrui Zhang, and Tianliang Hu. A robotic grasp detection method based on auto-annotated dataset in disordered manufacturing scenarios. *Robotics and Computer-Integrated Manufacturing*, 76:102329, 2022.

[200] Liyuan Zheng, Tanner Fiez, Zane Alumbaugh, Benjamin Chasnov, and Lillian J Ratliff. Stackelberg actor-critic: Game-theoretic reinforcement learning algorithms. *arXiv preprint arXiv:2109.12286*, 2021.

[201] Cezary Zielinski, Wojciech Szynkiewicz, Tomasz Winiarski, and Maciej Staniak. Rubik's cube puzzle as a benchmark for service robots. In *Proceedings of the 12th IEEE International Conference on Methods and Models in Automation and Robotics, MMAR*, pages 579–84, 2006.

[202] Cezary Zieliski, Tomasz Winiarski, Wojciech Szynkiewicz, Maciej Staniak, Witold Czajewski, and Tomasz Kornuta. Mrroc++ based controller of a dual arm robot system manipulating a rubiks cube. *Citeseer, Tech. Rep.*, 2007.

[203] e . Yaskawa bushido project / industrial robot vs sword master, 2015. URL https://www.youtube.com/watch?v=O3XyDLbaUmU.