Stackelberg MADDPG: Learning Emergent Behaviors via Information Asymmetry in Competitive Games

Boling Yang,[1 *] **Liyuan Zheng,** [1 *] **Lillian J. Ratliff,** [1] **Byron Boots,** [1] **Joshua R. Smith,** [1]

[1] University of Washington
[*] These Authors Contributed Equally
{bolingy, liyuanz8, ratliffl, bboots, jrs}@uw.edu

Abstract

Using competitive multi-agent reinforcement learning (MARL) methods to solve physically grounded problems, such as robust control and interactive manipulation tasks, has become more popular in the robotics community. 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 information advantage: the leader in ST-MADDPG updates using its total policy gradient, meaning it differentiates through the local best response of the follower. In a simple competitive robotics environment, we show that the leader learns a better policy by exploiting this information advantage and is able to either dominate the game or alleviate the native disadvantage from the game environment. In two practical robotic problems, ST-MADDPG allows the leader to learn more sophisticated and complex strategies that work well even against an unseen strong opponent.

1 Introduction

Multi-agent Reinforcement Learning (MARL) addresses the sequential decision-making problem of multiple autonomous agents that interact with each other in a common environment, each of which aims to optimize its own long-term return (Zhang, Yang, and Başar 2019). Purely competitive settings form an important class of sub-problems in MARL, and are typically formulated as a zero-sum two-player game using the framework of competitive Markov decision processes (Filar and Vrieze 2012). There has been much success in using competitive MARL methods to solve such problems, especially for symmetric games including extensive form games on finite action spaces such as chess and video games (Silver et al. 2017; Berner et al. 2019; Vinyals et al. 2019). These algorithms typically use a co-evolution training scheme in which the competing agents continually create new tasks for each other and incrementally improve their own policies by solving these new tasks. However, once one or more evolved agents fails to sufficiently challenge their opponent, subsequent training is unlikely to result in further progress due to a lack of pressure for adaptation. This cessation of the co-evolution process indicates that the agents have reached an equilibrium.

Recently, competitive MARL methods have gained attention from the robotics community and have been used to solve physically grounded problems, such as adversarial learning for robust control, autonomous task generation, and complex robot behavior learning (Dennis et al. 2020; Baker et al. 2019; Yang et al. 2021b,a). However, these problems are typically asymmetric in practice. Unlike a symmetric game where all agents have the same knowledge and the same ability to act, an asymmetric game requires the agents to solve their own task while coupled in an imbalanced competitive environment. One agent could gain advantage from having an easier initial task, and learn to exploit the advantage to quickly dominate the game (Pinto et al. 2017). This will prematurely terminate the co-evolving process and all agents are trapped in a low quality equilibrium. For example, in a simulated boxing game, if a player is able to punch significantly harder than the other, it can easily execute a knockout. Such a player could learn to knock out the opponent at the very beginning of a match, leaving no chance for the opponent to explore for better counter strategies such as strategic footwork to avoid the knockout blow.

To overcome this challenge, one common approach is to generate a large amount of diversified samples using population-based methods and distributed sampling (Bansal et al. 2017; Baker et al. 2019). Yet, this approach is resource consuming. For complex problems, such as controlling agents that have bodies with a high degree of freedom, this approach usually requires sampling data on multiple high performance computers for days. With a sufficient amount of engineering effort, some policy initialization methods such as reward shaping and imitation learning could be used to initialize the system to a desired state (Won, Gopinath, and Hodgins 2021; Yang et al. 2021a). The minimax regret strategy is a risk-neutral decision-maker that has been demonstrated to prevent the stronger player from dominating the game in adversarial learning (Dennis et al. 2020). Yet, the proposed best response update suffers from instabil-

ity. More importantly, by simply treating two players equally with a symmetric information structure and simultaneous learning dynamics, these methods fail to capture the inherent imbalanced underlying structure of the environment. In addition, as presented in (Foerster et al. 2018; Prajapat et al. 2020; Zheng et al. 2022), MARL methods that use simultaneous gradient descent ascent updates could result in poor convergence properties in practice.

In this paper, we aim to solve this problem by directly modifying the game dynamics to aid in re-balancing the bias in asymmetric environments. We leverage the Stackelberg game structure (Von Stackelberg 2010) to introduce a hierarchical order of play, and therefore an asymmetric interaction structure, into competitive games.

In a two-player Stackelberg game, the leader knows that the follower will react to its announced strategy. As a result of this structure, the leader optimizes its objective accounting for the anticipated response of the follower, while the follower selects a myopic best response to the leader's action to optimize its own objective. As a result, the leader stands to benefit from the Stackelberg game structure be achieving a better equilibrium payoff compared to that in a normal competitive game (Başar and Olsder 1998). This is a desirable property when one agent is the primary agent in the task (e.g., robust control with adversaries) or when one agent has initial or inherent disadvantage due to the asymmetric game environment and a re-balance of power is sought, as we will demonstrate in our experiments. The main **contributions** of this paper are listed below:

A Novel MARL Algorithm: ST-MADDPG. We formulate the competitive MARL problem as a Stackelberg game. By adopting the total derivative Stackelberg learning update rule, we extend the current state-of-the-art MARL algorithm MADDPG (Lowe et al. 2017) to a novel Stackelberg version, termed Stackelberg MADDPG (ST-MADDPG).

Multi-agent Co-evolution in Asymmetric Environment. From a novel perspective, we study how information and force exertion asymmetries affect the agents' performance and behaviors during the multi-agent co-evolution process. We first design a simple competitive RL benchmark with continuous control space: *competitive-cartpoles* (Figure 1.1). In this environment, we demonstrate that the use of the Stackelberg gradient updates provide an information advantage to the leader agent that compensates for the agent's initial or inherent disadvantage and leads to better performance.

Application to Practical Robotics Tasks. The practical effectiveness of the proposed algorithm is demonstrated in two tasks. In a robust control problem (Figure 1.2), having an information advantage during adversarial training allows the resulting robot to better survive adversarial and intense random disturbances. In a multi-agent competitive fencing game (Figure 1.3), ST-MADDPG allows the robot to learn complex strategies for better performance. Notably, one of the agents learned a strategy that is similar to the best performing human strategy recorded in a previous user study.

2 Preliminaries

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

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 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 $r : \mathcal{S} \times \mathcal{A}^1 \times \mathcal{A}^2 \to [0, 1]$ 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 $\pi : s_0 \sim \rho, a^1_t \sim \pi^1(\cdot|s_t), a^2_t \sim \pi^2(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a^1_t, a^2_t)$.

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.

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 a 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:

$$\min_{\theta_1\in\Theta_1} \{J_1(\theta_1, \theta_2) \,\big|\, \theta_2 \in \arg\min_{\phi\in\Theta_2} J_2(\theta_1, \phi)\}, \tag{L}$$

$$\min_{\theta_2\in\Theta_2} J_2(\theta_1, \theta_2). \tag{F}$$

Since the leader assumes the follower chooses a best response $\theta^*_2(\theta_1) = \arg\min_\phi J_2(\theta_1, \phi)$, the follower's decision variables are implicitly a function of the leader's. In

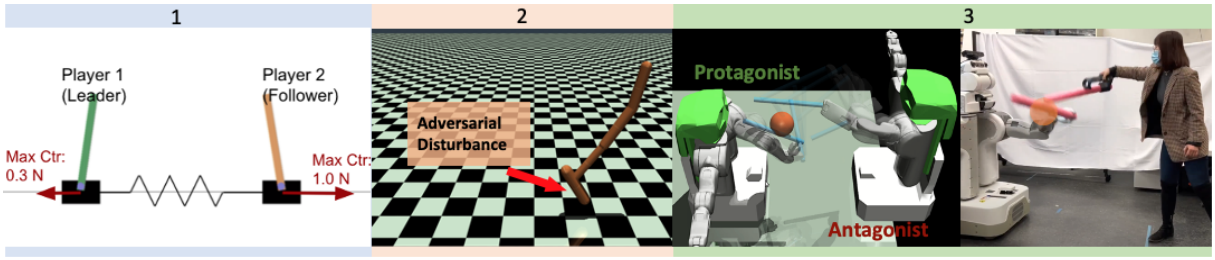


Figure 1: This work focuses on three competitive robotics tasks with physical interaction. 1. **Competitive-cartpoles** is a simple one-dimensional continuous control environment. 2. **Hopper with adversarial disturbances** is a classic robust control problem. 3. **The fencing game** is a competitive-HRI benchmark.

deriving sufficient conditions for the optimization problem in (L), the leader utilizes this information in computing the total derivative of its cost:

$$\nabla J_1(\theta_1, \theta_2^*(\theta_1)) = \nabla_1 J_1(\theta) + (\nabla \theta_2^*(\theta_1))^\top \nabla_2 J_1(\theta),$$

where $\nabla \theta_2^*(\theta_1) = -(\nabla_2^2 J_2(\theta))^{-1} \nabla_{21} J_2(\theta)$[1] by the implicit function theorem (Krantz and Parks 2002).

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 $\nabla_2 J_2(\theta_1, \theta_2) = 0$ and $\nabla_2^2 J_2(\theta_1, \theta_2) > 0$. This gives rise to the following equilibrium concept which characterizes sufficient conditions for a local Stackelberg equilibrium.

Definition 1 (Differential Stackelberg Equilibrium, Fiez, Chasnov, and Ratliff 2020)**.** *The joint strategy profile* $\theta^* = (\theta_1^*, \theta_2^*) \in \Theta_1 \times \Theta_2$ *is a differential Stackelberg equilibrium if* $\nabla J_1(\theta^*) = 0$, $\nabla_2 J_2(\theta^*) = 0$, $\nabla^2 J_1(\theta^*) > 0$, *and* $\nabla_2^2 J_2(\theta^*) > 0$.

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 $\theta_{2,k+1} = \theta_{2,k} - \alpha_2 \nabla_2 J_2(\theta_{1,k}, \theta_{2,k})$, where α_i, $i = 1, 2$ are the leader and follower learning rates.

MADDPG. (Lowe et al. 2017) showed that naïve policy gradient methods perform poorly in simple multi-agent continuous control tasks and proposed more advanced MARL algorithm termed 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_θ^i of each agent using the global critic. Consider the deterministic policy setting, each player has policy μ_θ^i, with parameter θ^i. 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 $\nabla_{\theta^2} J(\theta^1, \theta^2) = \mathbb{E}_{\xi \sim \mathcal{D}}\big[\nabla_{\theta^2} \mu_\theta^2(s) \nabla_{a^2} Q_w(s, a^1, a^2)|_{a^2 = \mu_\theta^2(s)}\big]$.

[1]The partial derivative of $J(\theta_1, \theta_2)$ with respect to the θ_i is denoted by $\nabla_i J(\theta_1, \theta_2)$ and the total derivative of $J(\theta_1, h(\theta_1))$ for some function h, is denoted ∇J where $\nabla J(\theta_1, h(\theta_1)) = \nabla_1 J(\theta_1, h(\theta_1)) + (\nabla h(\theta_1))^\top \nabla_2 J(\theta_1, h(\theta_1))$.

The critic objective is defined as the mean square Bellman error $L(w) = \mathbb{E}_{\xi \sim \mathcal{D}}[(Q_w(s, a^1, a^2) - (r + \gamma Q_{w'}(s', \mu_{\theta'}^1(s'), \mu_{\theta'}^1(s'))))^2]$, 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.

In MADDPG with competitive setting, the centralized critic is updated by gradient descent and the two agent's policy are update by simultaneous gradient descent and ascent $\theta^1 \leftarrow \theta^1 + \alpha^1 \nabla_{\theta^1} J(\theta^1, \theta^2)$, $\theta^2 \leftarrow \theta^2 - \alpha^2 \nabla_{\theta^2} J(\theta^1, \theta^2)$.

3 Stackelberg MADDPG Algorithm

In this section, we introduce our novel ST-MADDPG algorithm. A central feature of ST-MADDPG is that the leader agent exploits the knowledge that the follower will respond to its action in deriving its gradient based update. Namely, the total derivative learning update gives the information advantage to the leader by anticipating the follower's update during learning and leads to Stackelberg equilibrium convergence in a wide range of applications such as generative adversarial networks and actor-critic networks (Fiez, Chasnov, and Ratliff 2020; Zheng et al. 2022). According to Başar and Olsder (1998, Chapter 4), in the two-player game with unique follower best responses, the payoff of the leader in Stackelberg equilibrium is better than Nash equilibrium, which is desired in many applications. The full ST-MADDPG algorithm is shown in Algorithm 1 in Appendix A.1.

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

$$\theta^1 \leftarrow \theta^1 + \alpha^1 \nabla J(\theta^1, \theta^2),$$
$$\theta^2 \leftarrow \theta^2 - \alpha^2 \nabla_{\theta^2} J(\theta^1, \theta^2),$$

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

$$\nabla J(\theta^1, \theta^2) = \nabla_{\theta^1} J(\theta^1, \theta^2) - \nabla_{\theta^1 \theta^2} J(\theta^1, \theta^2)(\nabla_{\theta^2}^2 J(\theta^1, \theta^2))^{-1} \nabla_{\theta^2} J(\theta^1, \theta^2). \quad (1)$$

The second order terms of the total derivative in (1) can be computed by applying chain rule:

$$\nabla_{\theta^1 \theta^2} J(\theta^1, \theta^2) = \mathbb{E}_{\xi \sim \mathcal{D}}[\nabla_{\theta^1} \mu_\theta^1(s) \nabla_{a^1 a^2} Q_w(s, a^1, a^2) (\nabla_{\theta^2} \mu_\theta^2(s))^T|_{a^1 = \mu_\theta^1(s), a^2 = \mu_\theta^2(s)}],$$
$$\nabla_{\theta^2}^2 J(\theta^1, \theta^2) = \mathbb{E}_{\xi \sim \mathcal{D}}\left[\nabla_{\theta^2}^2 \mu_\theta^2(s) \nabla_{a^2} Q_w(s, a^1, a^2)|_{a^2 = \mu_\theta^2(s)}\right].$$

To obtain an estimator of the total derivative $\nabla J(\theta^1, \theta^2)$, each part of (1) is computed by sampling from a replay buffer. The inverse-Hessian-vector product can be efficiently computed by conjugate gradient (Zheng et al. 2022).

Implicit Map Regularization. The total derivative in the Stackelberg gradient dynamics requires computing the inverse of follower Hessian $\nabla^2_{\theta^2} J(\theta^1, \theta^2)$. Since policy networks in practical reinforcement learning problems may be highly non-convex, $(\nabla^2_{\theta^2} J(\theta^1, \theta^2))^{-1}$ can be ill-conditioned. Thus, instead of computing this term directly, in practice we compute a regularized variant of the form $(\nabla^2_{\theta^2} J(\theta^1, \theta^2) + \lambda I)^{-1}$. This regularization method can be interpreted as the leader viewing the follower as optimizing a regularized cost $J(\theta^1, \theta^2) + \frac{\lambda}{2}\|\theta^2\|^2$, while the follower actually optimizes $J(\theta^1, \theta^2)$. The regularization λ interpolates between the Stackelberg and individual gradient updates for the leader.

Proposition 1. *Consider a Stackelberg game where the leader updates using the regularized total gradient* $\nabla^\lambda J_1(\theta) = \nabla_1 J_1(\theta) - \nabla^\top_{21} J_2(\theta)(\nabla^2_2 J_2(\theta) + \lambda I)^{-1} \nabla_2 J_1(\theta)$. *The following limiting conditions hold: 1)* $\nabla^\lambda J_1(\theta) \to \nabla J_1(\theta)$ *as* $\lambda \to 0$*; 2)* $\nabla^\lambda J_1(\theta) \to \nabla_1 J_1(\theta)$ *as* $\lambda \to \infty$.

4 Experiments

In this section, we report on three experiment environments that provide insight into the following three main questions: **(Q1)**: How do agents with a continuous action spaces behave under capability and information asymmetries?; **(Q2)** Can a weaker agent's inherent disadvantage be compensated by the information advantage from ST-MADDPG? **(Q3)**: Does the proposed algorithm create better autonomous agents that solve real-world robotics problems?

Note that the trend of the cumulative reward of learning does not increase monotonically in competitive MARL environments as in well trained single-agent or cooperative MARL environments. Hence, to evaluate an agent's performance, we choose to collect gameplay data by having the trained agents play multiple games against its co-evolving partner from training or a hand-designed reference opponent. Further execution details are described in Section 4.1 and 4.2.

Competitive-Cartpoles. In order to answer **Q1** and **Q2**, we proposed a two-player zero-sum competitive game in which each agent solves a one-dimensional control task. As shown in Figure 2, this environment contains two regular cartpole agents. The dynamics of the two agents are coupled by a spring, where each end of the spring connects to one of the agent's bodies. Both agents will get a zero reward when they balance their own poles at the upright position simultaneously. If one of the agents loses its balance, this agent will receive a reward of -1 for every subsequent time step in the future until the game ends. The still balanced agent will get a reward of $+1$ for every time step until it also loses its balance and ends the game. As a result, the goal of each agent is to prevent its own pole from falling over, while seeking to break the balance of the opponent by introducing disturbing forces via the spring.

Hopper with Adversarial Disturbance. To investigate **Q3**, we will first focus on creating a robust control policy for the classic hopper environment using adversarial training (Duan et al. 2016; Pinto et al. 2017). Here, the first agent controls the classic hopper robot with four rigid links and three actuated joints. The second agent learns to introduce adversarial two-dimensional forces applied to the foot of the hopper.

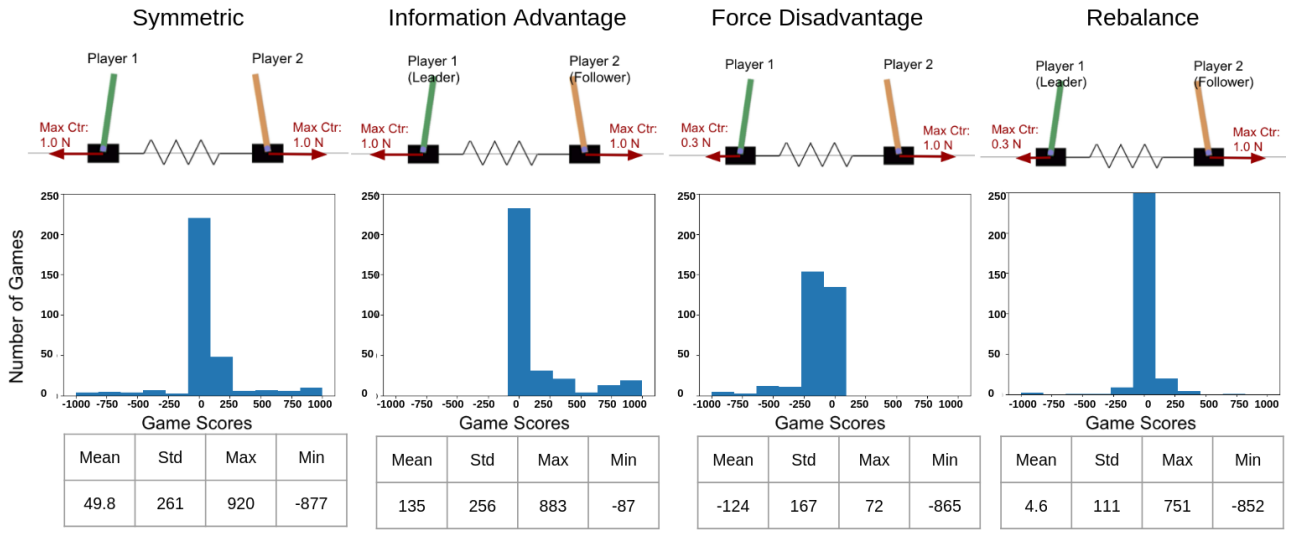
The Fencing Game. To further examine **Q3**, we consider an asymmetric zero-sum competitive game with complex environment dynamics proposed by Yang et al. (2021a). This game is a two player attack and defend game where one player is the antagonist who aims to maximize its game score by attacking a predefined target area with a sword, without making contact with the opponent's sword. The other agent is the protagonist who aims to minimize the antagonist's score by defending the target area. The game rules are detailed in Appendix A.4. This game is a challenging competitive MARL problem due to the fact that its highly asymmetric. In order to gain positive rewards, the antagonist has to reach out to the target area. However, being in the target area also correlates to a huge risk of being penalized by the protagonist. This gives the antagonist a harder task to solve compared to the protagonist.

4.1 Learning Under Asymmetric Advantage

This section explicitly studies the performance and behavior of the trained agents in the competitive-cartpoles environment under symmetric and asymmetric settings. We first demonstrate how ST-MADDPG can provide an information advantage to an agent and improve its performance. We then show that given an asymmetric environment where one agent has a force exertion advantage over the other, ST-MADDPG can be used to retain a balance in agents' performance.

Information Advantage. This experiment starts with a symmetric competitive-cartpoles environment, where both agents have the same ability to act. To understand how the information advantage inherent in the Stackelberg game structure affects the system's co-evolution process, we ran both MADDPG and ST-MADDPG methods on the competitive-cartpoles environment. MADDPG training represents a symmetric evolution environment, and ST-MADDPG training gives an information advantage to the leader (player 1).

For each of these two methods, we created four pairs of agents with four different random seeds. In order to compare the agents' performance between the two training methods, we ran a tournament and resulted 320 game scores and trajectories for each of the methods. The tournament details are discussed in Appendix A.2. The first two columns in Figure 2 summarize the statistics for the two tournaments. Note that the tournament game scores in this section refer to the scores of player 1. Therefore, a game will have a positive



Mean	Std	Max	Min
49.8	261	920	-877

Mean	Std	Max	Min
135	256	883	-87

Mean	Std	Max	Min
-124	167	72	-865

Mean	Std	Max	Min
4.6	111	751	-852

Figure 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.

score if player 1 wins, a negative score if player 2 wins, and zero if the two players are tied.

Under the symmetric setting (i.e., MADDPG), the performance of player 1 and player 2 are similar. The tournament has a mean score of 49.8. While the majority of the games were scored between -90 to 90, the rest of the games covered almost the entire score range from -877 to 920. This indicates that while the two players have similar performance in most cases, each of them can occasionally outperform the other by a lot. In contrast, when given an information advantage during training (i.e., ST-MADDPG), player 1 won more games with a larger mean score of 135. Player 2 only got -87 on its best win, meaning that the follower could never significantly outperform the leader. Therefore, the leader has better overall performance compared to the follower. When observing the agents' behaviors by replaying the collected trajectories, we found that the two players resulting from the symmetric environment usually compete intensively by pushing and pulling each other via the spring. While they are able to keep their own poles upright, they fail to break the balance of the other agent and win the game in most of the competitions. Meanwhile, for the agents from ST-MADDPG, the leader manages to learn a policy to pull the follower out of the frame to win the game. Video demonstration of the robots' behaviors can be found in this link.

Re-balancing Asymmetric Environment. Given that ST-MADDPG creates an information advantage that improves a specific agent's performance, we want to test if this information advantage can be used to compensate for a disadvantage that is assigned to an agent by the asymmetric environment. We created an asymmetric competitive-cartpoles environment by giving player 1 a force disadvantage, where player 1 has a decreased maximum control effort that is only 30% as much as player 2's maximum effort. Afterward, we once again trained agents using both MADDPG and ST-MADDPG (player 1 as the leader) with four random seeds and generated evaluation data with two tournaments.

As expected, under a substantial force disadvantage, player 1's performance was significantly worse than player 2 after the MADDPG training. However, as shown in Figure 2, when Stackelberg gradient updates are applied, the two players' performances are equivalent. With a mean score of 4.57, maximum score of 751, and a minimum score of -852, the information advantage is able to compensate for the force disadvantage for player 1 and generated a score distribution that is similar to the symmetric environment described above.

4.2 Application in Practical Robotics Problems

The previous section shows that ST-MADDPG can be used to change the learning dynamics in a multi-agent competitive game, which allows the system to converge to another equilibrium that is potentially more desirable to one or both agents in the system. In this section, we further explore ST-MADDPG in two practical robotics problems and evaluate whether it truly improves the robots' performance when competing against a strong and unseen opponent.

Hopper with Adversarial Disturbance In this robust control problem, we found that providing an information advantage to the robot in adversarial training can further improve the robustness of a robot control policy. The ST-MADDPG trained hopper agents outperformed the MADDPG trained hopper agents under both adversarial attacks and random disturbances with multiple intensity levels. The training details and experimental setups are further discussed in Appendix A.3.

Adversarial Attack. We used MADDPG and ST-MADDPG algorithms to create four pairs of hoppers and adversaries with four random seeds respectively. Each pair of agents were evaluated with 50 games, resulting in 200

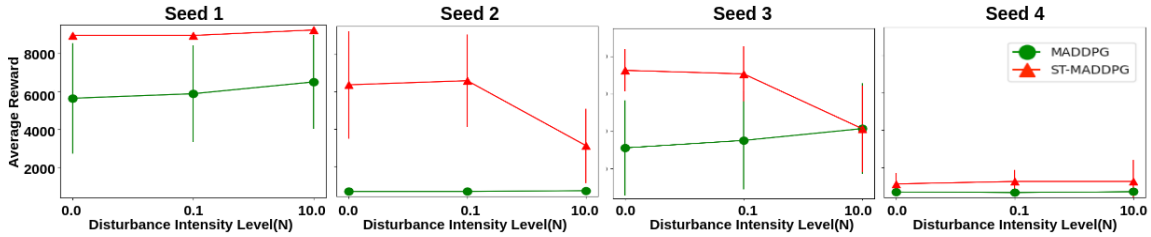


Figure 3: Performance comparison between four Hopper agents under three different levels of random disturbances. The error bars denote standard deviation.

game scores for each method. We found that the ST-MADDPG trained hoppers were able to survive significantly longer (**6002.5** avg. reward) than those from MADDPG training (**2877.3** avg. reward) under adversarial attacks.

High Intensity Random Disturbance. The generalizability of all hopper agents was tested under three environments with no disturbances and two different levels of strong random disturbances respectively (e.g., $0N$, $0.1N$, and $10.N$). Each agent ran 50 trials in each of the environments, and Figure 3 compares the agents' performance from the two training methods. Even though the maximum strength of the adversaries in training was bounded by $0.001N$, some of the agents (e.g., MADDPG:50%, ST-MADDPG:100%) still managed to receive more than 1000 average rewards under random disturbances with the maximum strength of $10N$. ST-MADDPG trained agents greatly outperformed MADDPG trained agents in all three levels of intensities. Therefore, ST-MADDPG policies are robust enough to maintain high performance under unseen scenarios.

The Fencing Game The antagonist in this game was assigned to solve a harder task compared to the protagonist. Therefore, this experiment focuses mainly on improving the performance of the antagonist. In the following subsections, we first demonstrate that the antagonist performs sub-optimally in a normal co-evolution process, and then show that the Stackelberg gradient updates improves the antagonist's performance when playing against its original opponent from training. As the leader in ST-MADDPG training, the antagonist was able to learn more sophisticated attacking strategies compared to the MADDPG trained antagonist. Afterward, we evaluated the antagonist agents' practical performance against a strong heuristic-based protagonist policy. We found that with the right amount of information advantage, the antagonist was able to win half of the games even when playing against a strong unseen opponent. Note that, the game scores discussed in this section refer to the protagonist's scores. A positive score indicates the victory of the protagonist and a negative score indicates the victory of the antagonist.

Improved Performance and Emergent Complexity. In this experiment, we trained the 1^{st} pair of agents under the original co-evolution environment of the game (i.e., MADDPG), and trained the 2^{nd}, 3^{rd} and 4^{th} pairs of agents by having the antagonist as the leader in the Stackelberg game setting (i.e., ST-MADDPG) with three different regularization values (i.e. λ) $5.5e5$, $1e6$, and $1.5e6$ respectively. We evaluated each pair of agents with 100 games, and the results are summarized in Appendix A.4 Table 1. The 1^{st} pair of agents have a mean score of 90.6, indicating the protagonist outperformed the antagonist. The 1^{st} antagonist was only able to learn a trivial attacking strategy that directly and repetitively reached out to the target area from one direction. Such an attacking strategy was overly greedy, making the antagonist fail to avoid most of the penalties. On the other hand, the antagonists' performance was significantly improved by ST-MADDPG training. The 2^{nd}, 3^{rd}, and 4^{th} antagonists outperformed their original opponents with average game scores of -31.6, -97.14, and -188.5 respectively. The improvement in the antagonists' performance was correlated to more complex behaviors and sophisticated strategies. The 2^{nd} and the 3^{rd} antagonists learned highly effective yet drastically different attacking strategies. Qualitative analysis of agents' behaviors are discussed in Appendix A.4.

Playing Against A Strong and Unseen Protagonist. In order to test if ST-MADDPG actually improves the equilibrium's quality, we further evaluated the four antagonists with a carefully designed heuristic-based protagonist policy. A higher quality equilibrium should result in more robust antagonist strategy with better performance when competing against an unseen opponent. By placing the protagonist's sword in between the target area and the point on the antagonist's sword that is closest to the target area, the heuristic-based policy exploits embedded knowledge of the game's rules in order to execute a strong defensive protagonist. The design of this heuristic-base policy is detailed in Appendix A.4. As shown in Table 2, with a mean score of 300.3 and a wining rate of 14%, the MADDPG trained (1^{st}) antagonist was dominated by the heuristic-based protagonist. In contrast, two of the ST-MADDPG trained antagonists with sophisticated behaviors were able to achieve significantly better results, where the 2^{nd} and 3^{rd} antagonist won 46% and 42% of the games, respectively. Although the 4^{th} antagonist has the best average reward out of the three ST-MADDPG trained agents in the previous experiment, it has the worst performance in this experiment. Therefore, to converge to a high quality equilibrium, it is crucial to carefully select a regularization value and maintain a good capability balance between the agents.

5 Conclusion

This work studies the application of MARL on asymmetric physically grounded competitive games. Due to the asymmetry in these environments, the co-evolution process of a multi-agent system could terminate prematurely and lead to a low-quality equilibrium. We proposed the Stackelberg-MADDPG algorithm, which formulates a two-player MARL problem as a Stackelberg game and provides an information advantage to one of the agents in the system. In a simple competitive physical game, we demonstrated that the agent's inherent advantage biases the training process, yet, the proposed algorithm can recreate a balance in the co-evolution environment. In robust control and interactive manipulation tasks, the proposed algorithm was able to create more robust and sophisticated policies compared to the state-of-the-art MARL algorithm MADDOG.

References

Baker, B.; Kanitscheider, I.; Markov, T.; Wu, Y.; Powell, G.; McGrew, B.; and Mordatch, I. 2019. Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528*.

Bansal, T.; Pachocki, J.; Sidor, S.; Sutskever, I.; and Mordatch, I. 2017. Emergent complexity via multi-agent competition. *arXiv preprint arXiv:1710.03748*.

Başar, T.; and Olsder, G. J. 1998. *Dynamic noncooperative game theory*. SIAM.

Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; Dębiak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; et al. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*.

Dennis, M.; Jaques, N.; Vinitsky, E.; Bayen, A.; Russell, S.; Critch, A.; and Levine, S. 2020. Emergent complexity and zero-shot transfer via unsupervised environment design. *arXiv preprint arXiv:2012.02096*.

Duan, Y.; Chen, X.; Houthooft, R.; Schulman, J.; and Abbeel, P. 2016. Benchmarking deep reinforcement learning for continuous control. In *International conference on machine learning*, 1329–1338. PMLR.

Fiez, T.; Chasnov, B.; and Ratliff, L. J. 2020. Implicit learning dynamics in stackelberg games: Equilibria characterization, convergence analysis, and empirical study. In *International Conference on Machine Learning*.

Filar, J.; and Vrieze, K. 2012. *Competitive Markov decision processes*. Springer Science & Business Media.

Foerster, J.; Chen, R. Y.; Al-Shedivat, M.; Whiteson, S.; Abbeel, P.; and Mordatch, I. 2018. Learning with Opponent-Learning Awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, 122–130.

Krantz, S. G.; and Parks, H. R. 2002. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media.

Lowe, R.; WU, Y.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; and Mordatch, I. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *Advances in Neural Information Processing Systems*, 30: 6379–6390.

Pinto, L.; Davidson, J.; Sukthankar, R.; and Gupta, A. 2017. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, 2817–2826. PMLR.

Prajapat, M.; Azizzadenesheli, K.; Liniger, A.; Yue, Y.; and Anandkumar, A. 2020. Competitive Policy Optimization. *arXiv preprint arXiv:2006.10611*.

Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; et al. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354.

Von Stackelberg, H. 2010. *Market structure and equilibrium*. Springer Science & Business Media.

Won, J.; Gopinath, D.; and Hodgins, J. 2021. Control strategies for physically simulated characters performing two-player competitive sports. *ACM Transactions on Graphics (TOG)*, 40(4): 1–11.

Yang, B.; Habibi, G.; Lancaster, P.; Boots, B.; and Smith, J. 2021a. Motivating Physical Activity via Competitive Human-Robot Interaction. In *5th Annual Conference on Robot Learning*.

Yang, B.; Lancaster, P.; and Smith, J. R. 2017. Pre-touch sensing for sequential manipulation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 5088–5095. IEEE.

Yang, B.; Lancaster, P. E.; Srinivasa, S. S.; and Smith, J. R. 2020. Benchmarking Robot Manipulation With the Rubik's Cube. *IEEE Robotics and Automation Letters*, 5(2): 2094–2099.

Yang, B.; Xie, X.; Habibi, G.; and Smith, J. R. 2021b. Competitive Physical Human-Robot Game Play. In *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, 242–246.

Zhang, K.; Yang, Z.; and Başar, T. 2019. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635*.

Zheng, L.; Fiez, T.; Alumbaugh, Z.; Chasnov, B.; and Ratliff, L. J. 2022. Stackelberg Actor-Critic: Game-Theoretic Reinforcement Learning Algorithms. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

A Algorithm and Experiment Details

A.1 ST-MADDPG

Algorithm 1 details the ST-MADDPG algorithm proposed in this paper. In this work we select the implicit map regularization hyperparameter λ for each environment via a grid search. In general, the neural network may be highly non-convex and the hessian inverse can be ill-conditioned. A larger regularization prevents the gradient from exploding and yields smoother learning dynamics, as observed in our experiments as well as in other Stackelberg learning applications (Fiez, Chasnov, and Ratliff 2020; Zheng et al. 2022). How to trade-off between Stackelberg and normal gradient learning by picking the regularization optimally or even adaptively is a future direction.

Algorithm 1: 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 corrent 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 $\mathcal{D}$;
    set $y_i = r_i + Q_{w'}(s_i, \mu^1_{\theta'}(s_i), \mu^1_{\theta'}(s_i))$;
    update the critic by minimizing the loss:

$$L(w) = \frac{1}{N}\sum_{i=1}^{N}[\big(Q_w(s_i, a^1_i, a^2_i) - y_i\big)^2]$$

    update the leader policy using the total gradient computed by (1):

$$\theta^1 \leftarrow \theta^1 + \alpha^1 \nabla^\lambda J(\theta^1, \theta^2)$$

    update the follower policy using the policy gradient:

$$\theta^2 \leftarrow \theta^2 - \alpha^2 \nabla_{\theta^2} J(\theta^1, \theta^2)$$

    update the target networks:

$$w' \leftarrow \tau w + (1-\tau) w'$$

$$\theta^{i\prime} \leftarrow \tau \theta^i + (1-\tau)\theta^{i\prime}$$

  end
end

A.2 Competitive-Cartpoles

The maximum length of the competitive-cartpoles game is 1000 time-steps in both training and experiments. We ran two tournaments to sample evaluation data, one for the agents resulting from MADDPG and the other for those from ST-MADDPG. In a tournament, each of the four

player 1 agents (resulted from four random seeds) played 20 games against each of the four player 2 agents, resulting in 320 game scores and trajectories. This evaluation process allows an agent to play games with not only its original co-evolving opponent but also the opponents that are trained in different random seeds, providing a more comprehensive summary for each training setting. In this environment, the regularization values λ in all ST-MADDPG training were set to one.

A.3 Hopper

During training, the maximum length of the games was bounded by 1000 time-steps for a shorter training time. However, in the evaluation experiments, all trials have a maximum length of 3000 time-steps to better distinguish agents' performance. In this environment, the regularization values λ in all ST-MADDPG training were set to 5000.

A.4 The Fencing Game

The fencing game is a competitive benchmark for human-robot interaction using a PR2 robot. This robot is comparable to a human in terms of body size and arm flexibility (Yang et al. 2020; Yang, Lancaster, and Smith 2017).

Game Rules. Algorithm 2 summarizes the scoring mechanism of the fencing game. The antagonist will get one point by placing its sword within the orange spherical(target) area located between the two agents. But the antagonist will receive a negative ten points of score penalty if its sword is placed within the target area and makes contact with the protagonist's sword simultaneously. Meanwhile, the goal for the protagonist agent on the left is to minimize the antagonist's score by giving him score penalties. Additionally, the antagonist will get 10 points of reward if the protagonist's sword is placed within the target area, passively waiting for the antagonist to attack for more than 2 seconds. Each agent has a seven dimension continuous control space. Each game will last for 1000 time-steps (i.e. 10 seconds)

Algorithm 2: The Fencing Game Scoring Mechanism

```
Initialize: Game score s = 0; Timestep = 0.01 Sec;
  Game horizon = 20 Sec
bat_a → Antagonist's bat
bat_p → Protagonist's bat
target → 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
```

Heuristic-based Protagonist Policy. We aimed to design a strong baseline heuristic policy to create an intense human robot gameplay experience. Given an observation of the world, the robot orients its bat perpendicular to the human's bat with random angular offsets drawn uniformly from -25 to 25 degrees on the x, y, and z axes. In order to ensure that the robot is always executing a competitive defense, the policy commands the robot to position the center of its bat in between the target area and the point on the human's bat that is closest to the target area:

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

Where $\bar{b_p}$, $\bar{tar}$, $\bar{h_{up}}$ and $\bar{h_{low}}$ represent the position of the robot's bat frame, the center of the target area, the upper end of human's bat, and the lower end of human's bat respectively. $\bar{h_{close}}$ indicates the point on the human's bat that is closest to the center of the target area, and L_{sword} indicates the length of a bat. The function $uniform(0.5, 1)$ randomly determines how far apart the robot's bat should be from the human's bat. In addition, there is a 50% chance for the robot to execute the desired bat position calculated from the last time step instead of the latest desired pose. The added uncertainties introduce randomness to the robot's behavior. This heuristic allows the robot to dominate the fencing game when it can move faster or as fast as the antagonist. In this experiment, the physical capability of the antagonist and protagonist agents are identical.

Qualitative Result. Table 1 and Table 2 summarizes the antagonists' performance when playing against their original opponents from training and the heuristic-based protagonist respectively. A positive game score indicates that the protagonist wins the game, and a negative score indicates the antagonist wins.

Quantitative Result – Emergent Behaviors. The improvement in the antagonists' performance was correlated to more complex behaviors and sophisticated strategies. Figure 4 visualizes the state visitation frequency of the two players. For example, the 2^{nd} antagonist learned to patiently prepare the attacks further away from the target area, and initiate the attacks when the protagonist's arm gets to a relatively less manipulable state. On the other hand, the 3^{rd} antagonist agent was able to aggressively position its sword closely to the target during the whole game. By carefully adjusting its sword's position and orientation with respect to the protagonist's end-effector pose, the antagonist always maintains a small amount of distance from the protagonist's sword without being penalized. This highly efficient maneuver is similar to the best performing human strategy we observed in our previous user study (Yang et al. 2021a). Video demonstration of the robots' and human player's behaviors described in this section can be found in this link.

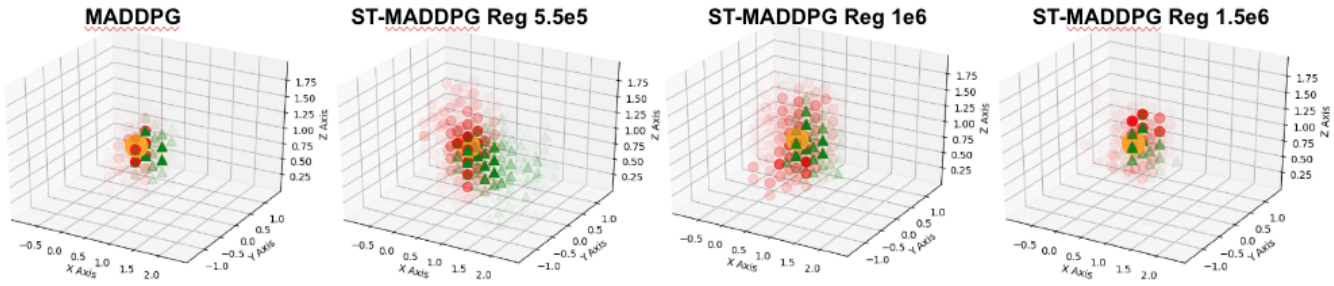


Figure 4: Each plot demonstrates the state visitation frequency of the protagonist's (i.e. red) and antagonist's (i.e. green) sword frame in task space. The orange sphere represents the target area, and a darker marker corresponds to a more frequently visited area.

Games Against Original Protagonist							
	Mean	Std	Max	Min	Tie	Prtg' Win	Antg' Win
MADDPG	90.6	246	1283	-177	2%	58%	40%
ST-MADDPG 5.5e5	-31.6	91.5	419	-192	6%	22%	72%
ST-MADDPG 1e6	-97.14	95.5	273	-300	0%	10%	90%
ST-MADDPG 1.5e6	-188.5	294	1276	-811	0%	8%	92%

Table 1: Statistical analysis for games between MARL trained antagonists and protagonists.

Games Against The Heuristic-based Protagonist							
	Mean	Std	Max	Min	Tie	Prtg' Win	Antg' Win
MADDPG	300.3	365.4	2042	-210	0%	86%	14%
ST-MADDPG 5.5e5	84.3	177.4	713	-94	2%	52%	46%
ST-MADDPG 1e6	120.9	189.6	606	-95	0%	58%	42%
ST-MADDPG 1.5e6	287.4	487	2479	-374	0%	74%	26%

Table 2: Statistical analysis for games between MARL trained antagonists and a heuristic-based protagonist.