2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Stackelberg Games for Learning Emergent Behaviors During Competitive Autocurricula

Boling Yang[1], Liyuan Zheng[2], Lillian J. Ratliff[2], Byron Boots[1], Joshua R. Smith[1,2]

***Abstract*— Autocurricular training is an important sub-area of multi-agent reinforcement learning (MARL) that allows multiple agents to learn emergent skills in an unsupervised co-evolving scheme. The robotics community has experimented autocurricular training 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 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. We first demonstrate that the leader's advantage from ST-MADDPG can be used to alleviate the inherent asymmetry in the environment. By exploiting the leader's advantage, ST-MADDPG improves the quality of a co-evolution process and results in more sophisticated and complex strategies that work well even against an unseen strong opponent.**

I. Introduction

Multi-agent Reinforcement Learning (MARL) addresses the sequential decision-making problem of multiple autonomous agents that interact in a common environment, each of which aims to optimize its own long-term return [1]. 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 [2]. There has been much success in using competitive autocurricular methods to solve such problems, especially for symmetric games including extensive form games on finite action spaces such as chess and video games [3], [4], [5]. These methods 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 fail 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 autocurricular 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 [6], [7], [8], [9]. 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 tasks while coupled in an imbalanced competitive environment. One agent could gain an advantage from having an easier initial task, and learn to exploit the advantage to quickly dominate the game [10]. This will prematurely terminate the co-evolving process and all agents will be 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 better counter strategies such as strategic footwork to avoid the knockout blow.

One common approach to overcome this challenge is to generate a massive amount of diversified samples using population-based methods and distributed sampling [7], [11]. But these methods are extremely computationally intensive and only suitable for running in a large computer cluster.

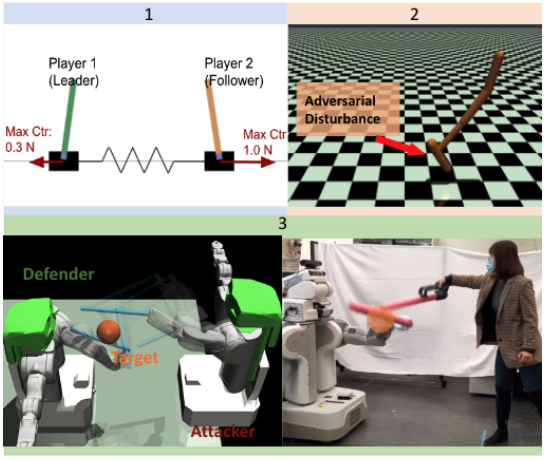


Fig. 1. This work focuses on three competitive robotics tasks with physical interaction. 1. **Competitive-cartpoles** 2. **Hopper with adversarial disturbances** and 3. **The fencing game**. Although our experiment focuses on collecting a large number of gameplay samples from simulation to evaluate the 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.

This work was supported in part by NSF award EFMA-1832795 and the UW + Amazon Science Hub

[1] Paul G. Allen School of Computer Science & Engineering, University of Washington, bolingy@cs.washington.edu

[2] Electrical and Computer Engineering Department, University of Washington

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 [8], [12]. Other training strategies such as minimax regret, goal-conditioned policy, and intrinsic motivation mitigate the stronger player from dominating the game in some adversarial learning settings [6], [13], [14]. However, 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, MARL methods that use simultaneous gradient descent ascent updates could result in poor convergence properties in practice [15], [16], [17]. Asymmetric gradient-based algorithms in multi-agent learning have been actively studied in recent years in normal-form game settings [18], [19]. Yet, this theoretical research has not been discussed from an application-oriented perspective.

We aim to solve this problem by directly modifying the game dynamics to aid in re-balancing the bias in asymmetric environments. In this paper, we leverage the Stackelberg game structure [20] to introduce a hierarchical order of play, and therefore an asymmetric interaction structure, into competitive Markov games [21]. 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 by achieving a better equilibrium payoff compared to that in a normal competitive game [22]. 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 an 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.

This paper has the following main contributions: **1. A Novel Autocurricular Algorithm:** We formulate the two-player 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 [23] to a novel Stackelberg version, termed Stackelberg MADDPG (ST-MADDPG). **2. Re-balancing Asymmetry with ST-MADDPG:** To demonstrate how asymmetries affect an autocurricular process, we explicitly study how force exertion asymmetry affects the agents' performance in the *competitive-cartpoles* environment (Fig.1.1) and show ST-MADDPG mitigates the environment asymmetry. **3. Improved Performance and Efficiency:** In a robust control problem (Fig 1.2), the leader advantage during adversarial training makes the resulting robot **3.19**$\times$ and **2.22**$\times$ more robust against adversarial and intense random disturbances, respectively, compared to standard MARL setting. **4. Learning Complex Emergent Behaviors:** In a competitive fencing game (Fig 1.3), the leader's advantage in ST-MADDPG motivates some attacking agents to behave more aggressively and learn complex strategies for superior performance. Notably, two of the best-performing attackers learned to trick the opponent to move to less manipulable joint configurations which temporarily limit the opponent's maneuverability.

II. Preliminaries

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

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 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

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

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 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:

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

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

Since the leader assumes the follower chooses a best response $\theta_2^*(\theta_1) = \arg\max_\phi J_2(\theta_1, \theta_2)$, 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 [24].

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 [19])**.** *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.

MADDPG. Lowe, et al.[23] showed that naïve policy gradient methods perform poorly in simple multi-agent continuous control tasks and proposed a 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^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:

$$L(w) = \mathbb{E}_{\xi \sim \mathcal{D}}[(Q_w(s, a^1, a^2) - (r + \gamma Q_{w'}(s', \mu_{\theta_1'}(s'), \mu_{\theta_2'}(s')))^2]. \quad (1)$$

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.

[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 $\nabla J(\theta_1, h(\theta_1)) = \nabla_{\theta_1} J(\theta_1, h(\theta_1)) + (\nabla h(\theta_1))^\top \nabla_{\theta_2} J(\theta_1, h(\theta_1))$.

[2]Following the setting and notation in origin DDPG algorithm [23], we use μ to represent deterministic policy to differentiate it from stochastic ones.

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 $\theta_1 \leftarrow \theta_1 + \alpha^1 \nabla_{\theta_1} J(\theta_1, \theta_2)$, $\theta_2 \leftarrow \theta_2 - \alpha^2 \nabla_{\theta_2} J(\theta_1, \theta_2)$.

III. 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 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 [19], [17]. According to [22, 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.

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

$$\theta_1 \leftarrow \theta_1 + \alpha^1 \nabla J(\theta_1, \theta_2), \quad (2)$$

$$\theta_2 \leftarrow \theta_2 - \alpha^2 \nabla_{\theta_2} J(\theta_1, \theta_2), \quad (3)$$

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

$$\nabla J(\theta_1, \theta_2) = \nabla_{\theta_1} J(\theta_1, \theta_2) - \nabla_{\theta_1 \theta_2} J(\theta_1, \theta_2)(\nabla^2_{\theta_2} J(\theta_1, \theta_2))^{-1} \nabla_{\theta_2} J(\theta_1, \theta_2). \quad (4)$$

The second order terms of the total derivative in (4) can be computed by applying the chain rule:

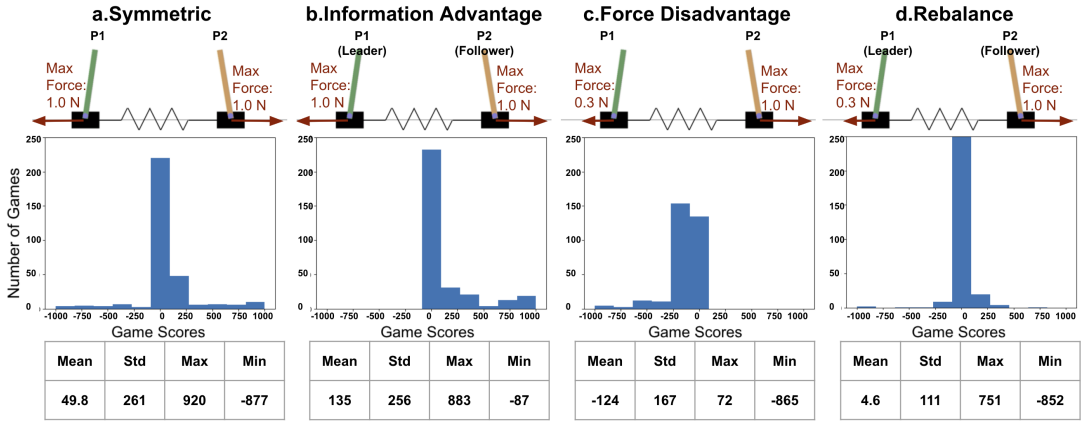
$$\nabla_{\theta_1 \theta_2} J(\theta_1, \theta_2) = \mathbb{E}_{\xi \sim \mathcal{D}}[\nabla_{\theta_1} \mu_{\theta_1}(s) \nabla_{a^1 a^2} Q_w(s, a^1, a^2) (\nabla_{\theta_2} \mu_{\theta_2}(s))^T|_{a^1 = \mu_{\theta_1}(s), a^2 = \mu_{\theta_2}(s)}],$$

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

To obtain an estimator of the total derivative $\nabla J(\theta_1, \theta_2)$, each part of (4) is computed by sampling from a replay buffer. The inverse-Hessian-vector product can be efficiently computed by conjugate gradient [17].

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 $\nabla^\lambda J_1(\theta) = \nabla_{\theta_1} J_1(\theta) - \nabla^\top_{\theta_2 \theta_1} J_2(\theta)(\nabla^2_{\theta_2} J_2(\theta) +$*



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

Fig. 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).

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 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 (1);
    update the leader policy using the total gradient computed by (2) and (4);
    update the follower policy using the policy gradient by (3);
    update the target networks;
  end
end

$\lambda I)^{-1}\nabla_{\theta_2} J_1(\theta)$. *The following limiting conditions hold: 1)* $\nabla^\lambda J_1(\theta) \to \nabla J_1(\theta)$ *as* $\lambda \to 0$*; 2)* $\nabla^\lambda J_1(\theta) \to \nabla_{\theta_1} J_1(\theta)$ *as* $\lambda \to \infty$.

IV. Experiments

In this section, we report on three experiment environments that provide insight into the following questions: **(Q1)**: How do different asymmetries in the training environment affect the performance and behavior of the agents? **(Q2)**: How to solve real-world robotics problems by exploiting the Stackelberg information structure? **(Q3)**: Is the total derivative update method used by ST-MADDPG better than an alternative approximation method? Note that the trend of the cumulative reward of learning does not increase monotonically in most of the competitive MARL environments as in well-trained single-agent or cooperative MARL environments. Hence, the main goal of this work is to allow agents to learn skills purely in simulation via MARL without needing real-world data and a carefully designed curriculum. A successful training process creates agents that are robust against unseen challenges (e.g., unseen strong opponents or an unseen form of disturbance). Therefore, for all experimental environments, an agent's performance is evaluated by one or more unseen opponents.

Competitive-Cartpoles. To answer **Q1**, we proposed a two-player zero-sum competitive game, 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 zero rewards 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 **Q2, 3**, we first focus on creating a robust control policy for the classic hopper environment using adversarial training [25], [10]. 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 **Q2**, we consider a zero-sum competitive game proposed by [8]. This game represents more practical robotics challenges such as complex robot kinematics, high uncertainty transition dynamics, and highly asymmetric game mechanism. This game is a two-player attack and defense game where the attacker aims to

maximize its game score by attacking a predefined target area with a sword, without making contact with the protector's sword. The protector aims to minimize the attacker's score by defending the target area. The detailed game rules can be found in the project website or [8].

A. Learning Under Asymmetric Advantage

This experiment explicitly studies the equilibrium of two symmetric and two asymmetric settings in the competitive-cartpoles environment. To evaluate the equilibrium of each setting, we created four pairs of agents with four different random seeds. We ran a tournament for each setting resulting in 320 game scores and trajectories. The tournament ensured each agent was evaluated by all unseen opponents from different random seeds. The project website contains extra tournament details.

Leader Advantage. This experiment starts with a symmetric competitive-cartpoles environment, where both agents have the same ability to act (i.e. Fig.2.a). To understand how the leader advantage inherent in the Stackelberg game structure affects the auto-curricular process, we ran both MADDPG and ST-MADDPG methods on the competitive-cartpoles environment. MADDPG training represents a symmetric environment, and ST-MADDPG training gives a leader advantage to player 1 (i.e. Fig.2.b).

Under the symmetric setting (i.e., MADDPG), the performance of player 1 and player 2 are similar. While the majority of the games in the tournament 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 performances in most cases, each of them can occasionally outperform the other by a lot. In contrast, when given a leader 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.

Re-balancing 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 an evaluation of their equilibrium with tournaments. As shown in Fig.2 c and d, under a substantial force disadvantage, player 1's performance was significantly worse than player 2 after the MADDPG training. However, 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 leader 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.

B. Hopper Against Adversarial and Random Disturbance

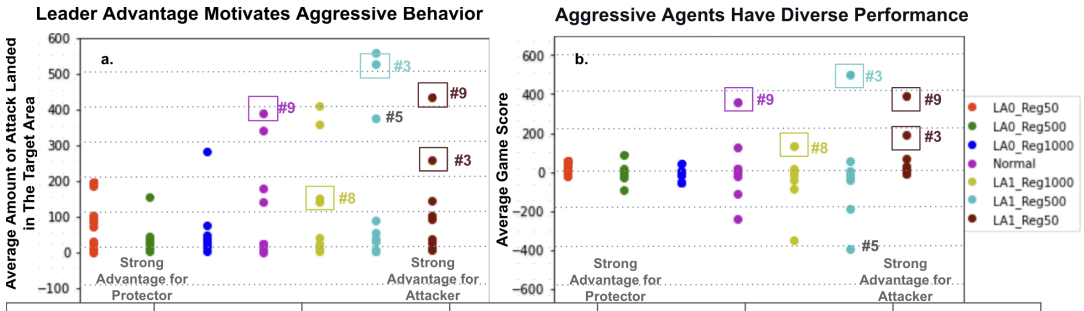
This work focuses on deriving and implementing a variation of MADDPG algorithm that uses the total derivative learning update to construct the Stackelberg information structure. While other individual learning algorithms are also popular in auto-curriculum works [6], [7], their decentralized value networks estimation via surrogate functions makes the direct total derivative computation infeasible. Alternatively, the Stackelberg information structure can be approximated by using different amounts of update steps for the leader and the follower in an individual learning setting [26]. The follower's best response update in a Stackelberg game is approximated by allowing the follower to make significantly more update steps than the leader for each batch of training data.

In this robust control problem, we first compare the ST-MADDPG with MADDPG. We then also evaluated the approximated Stackelberg update with two PPO-based training settings. The first PPO approach trains all agents with the original PPO in a decentralized manner, which is commonly used in auto-curriculum literature [6], [7], [8]. Afterward, we created a PPO variation with an approximated Stackelberg dynamic, ST-PPO, by having the follower (adversarial disturbance) take ten times more update steps than the leader (hopper).

The ST-MADDPG-trained hopper agents significantly outperformed the MADDPG-trained hopper agents under adversarial attacks (ST-MADDPG: 5113.6 avg. reward; MADDPG: 1601.6 avg. reward; avg. **319.3% improvement**) and random disturbances with multiple intensity levels (ST-MADDPG's average rewards were at least **2.22× better** than MADDPG). In contrast, the approximated Stackelberg information structure could only provide a mild (avg. **11%**) improvement to the leader during adversarial training. Similarly, the hopper agents from ST-PPO were slightly better (at most **1.14**×) than those from PPO in the random disturbance test. Therefore, **providing the leader advantage to the robot in adversarial training can further improve the robustness of a robot control policy.** With a similar computational complexity, **the total derivative update was more effective than the approximated Stackelberg update in terms of constructing the leader advantage.** More discussion on this experiment and the computational complexity of ST-MADDPG and ST-PPO can be found in our project website.

C. Co-evolution Under Complex Environment

The Fencing game is a challenging task for autocurricular training. The robot kinematics and the contact-rich nature of the game introduce high complexity and uncertainty



Top Performing Attackers	*1st Best:* LA1_Reg500 #3	*2nd Best:* LA1_Reg50 #9	*3rd Best:* Normal #9	*4th Best:* LA1_Reg50 #3	*5th Best:* LA1_Reg1000 #1	The Entire Population
Average Score	498.6	393.1	355.3	192.4	131.0	4.3
Winning Rate	100%	91%	90%	52%	76%	47%

Fig. 3. Evaluating autocurriculum trained attackers via an unseen strong heuristic-based protector. The name of each group provides information about the training setting, where the first part indicates the leader agent(i.e. LA0:Protector, LA1:Attacker), and the second part indicates the regularization value. There are ten agents in each setting, and the IDs of the top-performing attackers are printed on the right side of the corresponding points on both plots. **a.** Shows the average amount of attacks executed by an attacker in a single game (i.e. aggressiveness). As the training setting became more advantageous to the attackers, a subset of the attackers become more aggressive. **b.** Shows the average game score of 100 games between each attacker and the protector. The top five high-performing agents (highlighted with boxes) were some of the more aggressive agents in the population.

to the environment transition model. Moreover, its game mechanism is also asymmetric. Since the protector is only rewarded by making contact with the attacker within the target area, the attacker leads the MARL process by initiating the attacking actions. If the attacker learns to stop attacking to avoid penalties, the policy updates for both agents could be less effective in that they become stuck in exploring highly sub-optimal areas of the task space. Therefore, the attacker plays a more important role in the game and we will focus on evaluating the attacker's behavior and performance in this experiment.

This experiment studied how having different levels of advantage or disadvantage in the training process changes the attackers' behavior and performance. By assigning the protector as the leader in ST-MADDPG, we trained three groups of attackers under strong(λ=50), medium(λ=500), and weak(λ=1000) disadvantage. Similarly, we created three groups of attackers with three levels of leader advantage and one group from MADDPG. Ten pairs of agents are trained with ten different random seeds for each group. We then evaluated each of the 70 trained attackers with 100 games against a carefully designed heuristic-based protector policy. By placing the protector's sword in between the target area and the point on the attacker's sword closest to the target area, the heuristic-based policy exploits embedded knowledge of the game's rules to execute a strong defensive strategy. Having this heuristic baseline policy allows fair comparison between attackers. Autocurricula that converge to a higher quality equilibrium should result in a more robust attacker strategy with better performance when competing against this strong unseen protector. Additional details about the heuristic-based policy and this experiment are described in our project website.

Increase of Attackers' Aggressiveness. In all seven training settings, some attackers converged to conservative strategies that execute fewer attacking actions with relatively low positive(winning) or negative(losing) game scores. This result is not surprising given that initiating an attack action also correlates to a considerable risk of being penalized by the protector. However, as shown in Fig.3.a, when the attackers became more and more advantageous in training, a subset of the attackers became increasingly aggressive in terms of landing more attacks on the target area.

Emergent Complexity. Increased engagement in the competition could help develop better attack strategies. Indeed, Fig.3.b shows that the top five best-performing attackers are all located on the right half of the plot, where their co-evolving opponents are not the leader of a Stackelberg game. The top two attackers come from two ST-MADDPG settings where the attackers were the leader with a regularization value of 500 and 50, respectively. They both learned to trick the protector into moving to a less manipulable joint configuration. These attackers then score with relatively low risk while the protector is partially trapped and busy moving out of that configuration. Yet, some other aggressive policies were overly engaged and resulted in poor performance (e.g. agent#5 from LA1_Reg1000 and LA1_Reg500 in Fig.3).

V. Conclusion

We proposed the ST-MADDPG algorithm, which formulates a two-player MARL problem as a Stackelberg game and provides an advantage to one of the agents in the system. We demonstrated that an agent's inherent advantage over the other could bias the training process towards unfavorable equilibria. Our proposed algorithm re-balances such environments to improve the quality of resulted agents. Furthermore, the total derivative update used in ST-MADDPG was more effective than an alternative approximated Stackelberg update in constructing the leader's advantage.

REFERENCES

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

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

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

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

[5] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

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

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

[8] B. Yang, G. Habibi, P. Lancaster, B. Boots, and J. Smith, "Motivating physical activity via competitive human-robot interaction," in *5th Annual Conference on Robot Learning*, 2021.

[9] B. Yang, X. Xie, G. Habibi, and J. R. Smith, "Competitive physical human-robot game play," in *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, 2021, pp. 242–246.

[10] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2817–2826.

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

[12] J. Won, D. Gopinath, and J. Hodgins, "Control strategies for physically simulated characters performing two-player competitive sports," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–11, 2021.

[13] O. OpenAI, M. Plappert, R. Sampedro, T. Xu, I. Akkaya, V. Kosaraju, P. Welinder, R. D'Sa, A. Petron, H. P. d. O. Pinto *et al.*, "Asymmetric self-play for automatic goal discovery in robotic manipulation," *arXiv preprint arXiv:2101.04882*, 2021.

[14] S. Sukhbaatar, Z. Lin, I. Kostrikov, G. Synnaeve, A. Szlam, and R. Fergus, "Intrinsic motivation and automatic curricula via asymmetric self-play," *arXiv preprint arXiv:1703.05407*, 2017.

[15] J. Foerster, R. Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, and I. Mordatch, "Learning with opponent-learning awareness," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '18, 2018, p. 122–130.

[16] M. Prajapat, K. Azizzadenesheli, A. Liniger, Y. Yue, and A. Anandkumar, "Competitive policy optimization," *arXiv preprint arXiv:2006.10611*, 2020.

[17] L. Zheng, T. Fiez, Z. Alumbaugh, B. Chasnov, and L. J. Ratliff, "Stackelberg actor-critic: Game-theoretic reinforcement learning algorithms," *arXiv preprint arXiv:2109.12286*, 2021.

[18] T. Lin, C. Jin, and M. I. Jordan, "Near-optimal algorithms for minimax optimization," in *Conference on Learning Theory*. PMLR, 2020, pp. 2738–2779.

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

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

[21] Y. Yang and J. Wang, "An overview of multi-agent reinforcement learning from game theoretical perspective," *arXiv preprint arXiv:2011.00583*, 2020.

[22] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, 1998.

[23] R. Lowe, Y. WU, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in Neural Information Processing Systems*, vol. 30, pp. 6379–6390, 2017.

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

[25] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International conference on machine learning*. PMLR, 2016, pp. 1329–1338.

[26] A. Rajeswaran, I. Mordatch, and V. Kumar, "A game theoretic framework for model based reinforcement learning," in *International conference on machine learning*. PMLR, 2020, pp. 7953–7963.