

Turning Safety into Competence: Minimally Exploitable Robot Policies via Safety-Filtered Reinforcement Learning

Ruihan Wu^{1,*}, Rui Yang^{1,*}, Donggeon David Oh², Duy P. Nguyen^{2,†}, and Haimin Hu^{1,†}



Fig. 1: Our proposed **Safety to Competence (S2C)** enforces robot safety without sacrificing competitive performance by training multi-agent reinforcement learning (RL) policies with a pretrained robust safety filter embedded in the environment. Here, a quadrupedal robot (orange trajectory) plays a touchdown game against a human opponent (grey trajectory), where the attacker must reach the green touchdown region before timeout; the robot loses immediately if it falls, crosses the red field boundary, or initiates a collision. **Top:** S2C learns effective attacking and defending tactics. As attacker, it feints past the human to score a touchdown; as defender, it tracks rapid human direction changes and holds the line. **Bottom:** safe RL baselines fail frequently under competitive opponent strategies: the ET attacker exits the field while turning, the CPO attacker collides with the human, and the ET defender falls while pursuing the human. Project webpage: <https://alliance-ai.cs.jhu.edu/s2c/>.

Abstract—Robots deployed for competitive tasks must outmaneuver their opponents without sacrificing safety. Existing approaches, including safe reinforcement learning (RL), train a single policy to achieve task success and avoid failures simultaneously. This coupling can complicate training and leave the learned policy exploitable by deliberate attacks. We propose **Safety to Competence (S2C)**, a two-stage RL framework that separates safety synthesis from competitive task learning. We formulate competitive interactions as safety-critical Markov games and prove that perfect filtering preserves policy non-exploitability when all players commit to safe maneuvers. S2C learns a robust safety filter via adversarial RL, embeds it in the environment during task policy training, and retains the same filter at deployment. In simulated touchdown games, S2C outperforms eight safe RL baselines, achieving the highest win rate and Elo rating, and the lowest exploitability. Hardware stress tests against a human opponent confirm S2C’s competence.

¹R. Wu, R. Yang, and H. Hu are with the Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218, USA. {rwu63,yrui5}@jh.edu, haimin@cs.jhu.edu

²D. D. Oh and D. P. Nguyen are with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544, USA. {do9948,duyn}@princeton.edu

*Equal contribution. †Equal advising.

I. INTRODUCTION

A wide range of robotic applications involves competitive interactions with other agents in close proximity, ranging from drone racing [1] and motorsports [2, 3] to robot soccer [4] and pursuit–evasion games [5]. These tasks require strategic play under strict safety constraints, since a safety violation can cause catastrophic hardware damage or bodily harm and, in these games, result in an immediate loss. Existing approaches for safety-critical, competitive robot motion planning, including a broad class of safe reinforcement learning (RL) algorithms [6], focus on synthesizing a single policy that achieves safety and task competence simultaneously. Common techniques include penalizing safety violations in the reward [7], constraining the expected cumulative safety cost [8, 9], or terminating the episode upon a violation [10]. Coupling the two objectives in this way complicates training and leaves a trade-off between safety and competence that is difficult to tune [7]. Training therefore tends to converge to an overly conservative or overly aggressive strategy. Both are vulnerable to deliberate attack by an exploitative

opponent (see, for example, Figs. 1 and 2).

Safety filters [11] offer a principled approach to separate safety from the task policy. A safety filter supervises the task policy and modifies its proposed action to prevent an upcoming failure. Recent work demonstrates that (adversarial) RL can be used to synthesize such filters for robots that interact with antagonistic opponents and operate in high-dimensional observation and action spaces [12–15]. Restricting the task policy’s actions with a filter is often seen as sacrificing the robot’s overall performance. Recent work [16] addresses this concern by proving that for a single agent, perfect (i.e., least restrictive) filtering preserves the asymptotic optimal return in safety-constrained RL. However, this guarantee does not carry over directly to competitive multi-agent settings, where each player’s task performance and safety depend on the opponent’s strategy.

Contributions. This paper establishes a game-theoretic foundation for safety-critical, competitive robot interaction and develops a practical algorithmic framework for minimizing policy exploitability without sacrificing safety. *Our key insight is that separating safety from task objectives in RL can substantially reduce policy exploitability*, provided the same safety filter is applied during both training and deployment. Our key contributions are:

- We formulate competitive robot interactions as safety-critical Markov games and show for the first time that perfect safety filters yield non-exploitable Nash equilibria within the class of safe policies.
- We propose a two-stage S2C framework that first learns a robust safety filter and then embeds it in the RL environment to learn safe and competitive robot policies.
- We extensively evaluate S2C in simulated quadruped touchdown games. It achieves the highest win rates and, in the symmetric case, the highest Elo rating and lowest empirical exploitability compared to eight safe RL baselines. Hardware tests further demonstrate S2C’s robustness to human strategies unseen during training.

II. RELATED WORK

Safety Filters. A safety filter supervises a task controller and modifies its proposed action only when necessary to avoid catastrophic failures. Existing filter designs include, for example, model predictive safety filters (MPSFs) [17], control barrier functions (CBFs) [18, 19], and Hamilton–Jacobi (HJ) reachability [20]. Applying these methods to high-dimensional systems with black-box dynamics presents modeling and computational challenges. MPSFs require a predictive dynamics model and online trajectory optimization, which can be computationally expensive. Conventional model-based CBF filters require known dynamics information; constructing such a certificate is challenging for nonlinear dynamics and calls for a case-by-case analysis. Classical grid-based reachability methods scale poorly with the state dimension. Neural approximations have recently enabled filter synthesis for higher-dimensional robotic systems [21, 22]. To account for disturbances that deliberately seek to cause

failures, adversarial safety RL methods train a disturbance policy against the controller, jointly learning a safety critic and a fallback policy [12, 23]. Oh et al. [15] use this approach to synthesize a robust Q-control barrier function (Q-CBF), enabling model-free, smooth safety filtering. We adopt this technique in our S2C framework to approximate perfect filters and achieve task-efficient safety overrides.

Safe Reinforcement Learning. Existing safe RL methods predominantly incorporate safety into policy learning through reward penalties or cost constraints [6]. A widely adopted safe RL formulation is the constrained Markov decision process (CMDP), which maximizes the expected task payoff subject to a budget on the expected cumulative safety cost. Constrained Policy Optimization (CPO) [9] approximately enforces this budget during policy updates, while Lagrangian methods [7] convert the constraint into an adaptive penalty. Early termination instead ends the episode at a failure event, but does not actively prevent the policy from reaching that condition. These approaches do not generally guarantee safety in competitive interactions. A positive expected-cost budget can still permit rare, yet catastrophic violations, and reward penalties may favor higher task return at the expense of safety on some trajectories. Moreover, the trajectory distribution depends on the opponent’s strategy; a policy that satisfies its cost budget against training opponents may exceed it against an unseen opponent at deployment. In this paper, we advocate for a safety–task separation principle that first trains a robust safety filter and then embeds it in task-policy training. We prove that, under perfect filtering, Nash equilibria of the filtered game induce non-exploitable policies. Empirically, we demonstrate that this approach achieves substantially stronger performance and lower exploitability than the above safe RL baselines.

Game-Theoretic Multi-Robot Interaction. Dynamic games model robot interactions where each agent’s outcome depends on both its own policy and others’ responses [24]. A popular model-based game solver, the iterative linear-quadratic game [25], computes local feedback Nash equilibria by linearizing the dynamics and quadratizing the players’ objectives. Lidard et al. [26] extend this approach to incorporate a data-driven reference policy. These approaches generally require known dynamics and differentiable objectives, and their runtime computational cost can limit their applicability to high-dimensional robotic systems. For complex or unknown dynamics, multi-agent reinforcement learning (MARL) instead learns strategic policies through simulated interactions. Centralized-training methods use joint information during training to mitigate the nonstationarity caused by simultaneously learning agents [27], while self-play and population-based methods train against an evolving set of opponents [28]. Independent proximal policy optimization (IPPO) [29] provides a decentralized approach by optimizing policies for each agent using its own observations without requiring a centralized critic or an explicit opponent model. We use IPPO as the task-policy training method in the filtered environment.

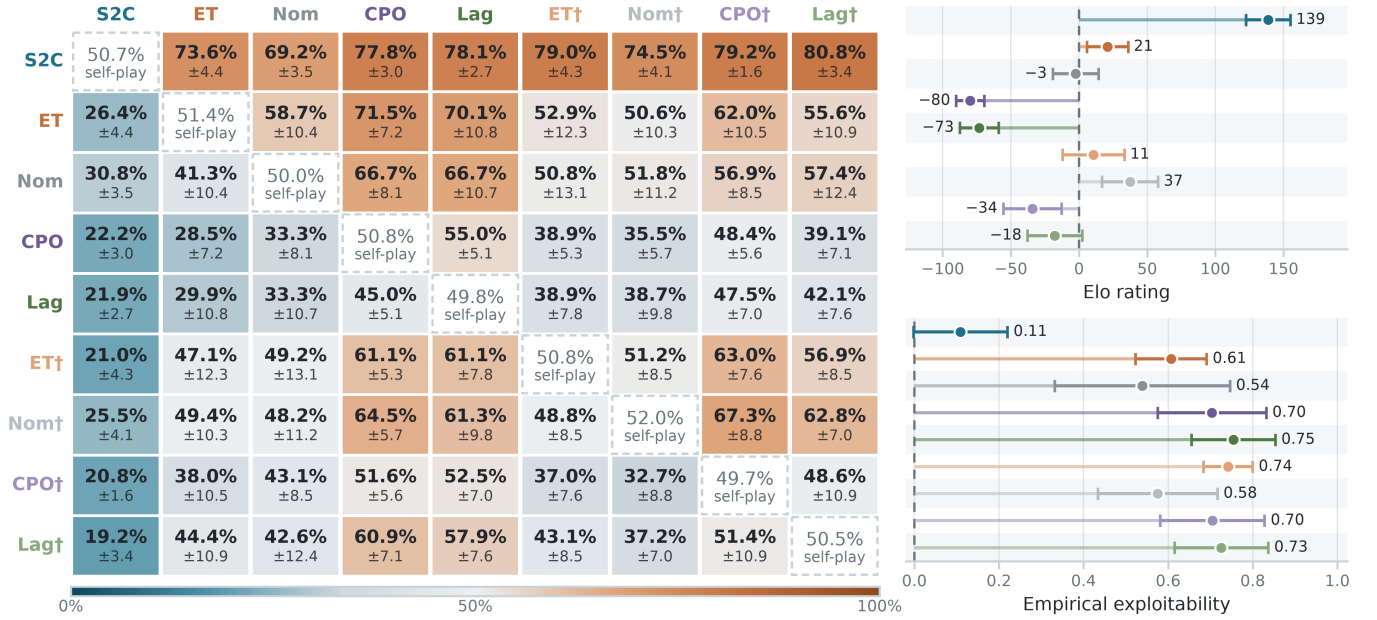


Fig. 2: Symmetric touchdown game results. Each cell corresponds to 3900 trials. S2C outperforms all baselines, achieving the highest Elo rating and lowest empirical exploitability. **Left:** round-robin win-rate matrix, where each cell reports the row method’s win rate against the column method, averaged over all pairs of training seeds. **Top right:** Elo ratings computed jointly based on all matches (higher is better). **Bottom right:** empirical exploitability, computed for each seed as the net win rate achieved by its strongest evaluated opponent and then averaged across seeds (lower is better).

III. PROBLEM FORMULATION

We consider competitive, safety-critical interactions between two robots (or two groups of robots), in which each player seeks to maximize its task performance over time against its opponent; a player *loses immediately* if it violates a safety constraint. Formally, we model such interactions with the following Markov game.

Definition 1 (Safety-Critical Markov Game). A Safety-Critical Markov Game (SC-MG) is defined as the tuple

$$\mathcal{G}_{\text{SC}} := (\mathcal{S}, \mathcal{A}^1, \mathcal{A}^2, P, \gamma; r, \rho; g^1, g^2; g^{1,c}, g^{2,c}; \alpha), \quad (1)$$

where $s = (s^1, s^2) \in \mathcal{S}$ is the physical joint state, and $\mathcal{A}^i$ is the action space for player $i \in \{1, 2\}$, $P(\cdot | s, a^1, a^2)$ is the transition kernel, $\gamma \in (0, 1)$ is a discount factor, task reward $r : \mathcal{S} \rightarrow \mathbb{R}$ is bounded, parameter $\rho > 0$ specifies the absorbing win-loss reward, and α is the *adjudication map*, which we explain next.

Safety Decomposition. For each player i , safety is specified by a *solo* margin function $g^i(s^i)$ and a *coupled* margin $g^{i,c}(s^1, s^2)$, where negative values indicate violations. The total safety margin is $g_{\text{tot}}^i(s) := \min\{g^i(s^i), g^{i,c}(s^1, s^2)\}$, and the corresponding *failure set* is $\mathcal{F}^i := \{s \in \mathcal{S} \mid g_{\text{tot}}^i(s) < 0\}$. If s enters $\mathcal{F}^1 \cup \mathcal{F}^2$, an *adjudicated outcome* is triggered and the game is terminated.

Fault Attribution. In an SC-MG, the winning condition depends on fault attribution. Conceptually, a player i wins if the opposing party violates safety. However, it is possible that both players’ safety specifications are violated simultaneously, i.e., $s \in \mathcal{F}^1 \cap \mathcal{F}^2$. Below, we specify how such states are adjudicated.

Assumption 1 (Unique Fault Attribution). The adjudication map $\alpha : \mathcal{S} \rightarrow \{1, 2\}$ identifies the *unique* player held at fault at each state $s \in \mathcal{F}^1 \cap \mathcal{F}^2$, and $\alpha(s) = i$ whenever $s \in \mathcal{F}^i \setminus \mathcal{F}^{-i}$.

Running Example: Throughout the paper, we use a symmetric two-player quadruped touchdown game as a running example. The game is played on a 5.2×3.0 m field, with two Unitree Go2 quadrupedal robots initialized at opposite ends and competing to reach the opponent’s touchdown line first. A player receives a (sparse) reward $r = +1$ when it first reaches its specified touchdown line and -1 when the opponent does. The solo safety margin g^i is defined as the minimum distance to the ground and the field boundary, and the coupled margin $g^{i,c}$ measures the relative distance to the opponent. The adjudication map α is defined such that, in the event of an inter-robot collision, the robot with the higher velocity projected toward the opponent (i.e., one that closes faster on its opponent) is attributed fault.

In order to formally show that SC-MG (1) is zero-sum, we consider an equivalent state space $\tilde{\mathcal{S}} := (\mathcal{S} \setminus (\mathcal{F}^1 \cup \mathcal{F}^2)) \cup \{\top, \perp\}$, where $\top$ and $\perp$ are *absorbing states* denoting an immediate win and loss for Player 1. Subsequently, we consider the following augmented reward:

$$\bar{r}(s) := \begin{cases} +\rho, & s = \top, \\ -\rho, & s = \perp, \\ r(s), & s \in \mathcal{S} \setminus (\mathcal{F}^1 \cup \mathcal{F}^2). \end{cases} \quad (2)$$

At each time, player i receives stage reward $\sigma^i \bar{r}$, where $\sigma^1 := +1$ and $\sigma^2 := -1$. Consequently, along every realized trajectory, $\sum_{t=0}^{\infty} \gamma^t (\sigma^1 \bar{r}(s_t) + \sigma^2 \bar{r}(s_t)) = 0$.

Proposition 1. *Under Assumption 1, SC-MG (1) is zero-sum.*

Exploitability and Game Value. To quantify a policy’s *robust competitiveness* in this zero-sum game, we next establish the game’s *value* and introduce *exploitability*, which measures how vulnerable a robot is to a best-responding, exploitative opponent that deliberately targets its weaknesses.

Recall that the SC-MG (1) is a discounted zero-sum stochastic game with bounded stage rewards. Under the standard minimax assumptions [30], the discounted stochastic game guarantees that its max–min and min–max payoffs coincide. We therefore obtain the following proposition.

Proposition 2 (Existence of the Game Value). *The SC-MG (1) admits a value at every $s \in \bar{\mathcal{S}}$:*

$$\begin{aligned} V^*(s) &= \sup_{\pi^1 \in \Pi^1} \inf_{\pi^2 \in \Pi^2} J(\pi^1, \pi^2)(s) \\ &= \inf_{\pi^2 \in \Pi^2} \sup_{\pi^1 \in \Pi^1} J(\pi^1, \pi^2)(s), \end{aligned} \quad (3)$$

where $J(\pi^1, \pi^2)(s) := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \bar{r}(s_t) \mid s_0 = s]$.

Proposition 2 implies that there exists a stationary *Nash equilibrium* (NE) policy pair $(\pi^{1,*}, \pi^{2,*})$ that guarantees the value V^* against every opponent policy. In this case, an NE policy is *non-exploitable*: neither player can improve its expected payoff by changing to another policy [24, Ch. 2].

Definition 2 (Exploitability). For a policy π^i , its *exploitability* is defined as the gap between the game value and the worst-case payoff:

$$\text{expl}(\pi^i)(s) := \sigma^i V^*(s) - \inf_{\pi^{-i} \in \Pi^{-i}} \sigma^i J(\pi^1, \pi^2)(s), \quad (4)$$

where $\sigma^1 = +1$ and $\sigma^2 = -1$.

At an NE $(\pi^{1,*}, \pi^{2,*})$ of SC-MG, neither player’s strategy can be exploited, i.e., $\text{expl}(\pi^{1,*})(s) = \text{expl}(\pi^{2,*})(s) = 0$.

IV. ANALYSIS: NON-EXPLOITABILITY UNDER PERFECT SAFETY FILTERING

In this paper, our goal is to improve robot competitiveness with safe yet minimally exploitable policies. However, direct equilibrium learning in an SC-MG is challenging since it couples task optimization with safety reasoning; each player needs to preempt safety risks induced by close-call interactions with the opponent while searching for winning strategies. To this end, we propose a two-stage policy synthesis framework that *separates safety enforcement from competitive task objectives* (Fig. 3). First, we synthesize a robust safety filter [11] for each player using HJ-based adversarial RL. Second, we train task policies in an RL environment that incorporates the filters into its transition dynamics, and deploy each policy with the same filter. Because the same filters are used during both training and deployment, the task policy adapts to the filtered game dynamics without requiring an explicit filter model or additional reward terms.

In this section, we establish the theoretical foundation for safety–task separation in zero-sum, safety-critical games. Our main result shows that, under perfect filtering, an NE

of the filtered game induces an NE of the original SC-MG in the space of safe policies. Consequently, equilibrium policies executed under the same filters remain safe and non-exploitable against safety-aware opponents. We then introduce a practical instantiation of this framework in Sec. V.

Let $\Omega^{i*} \subseteq \mathcal{S} \setminus \mathcal{F}^i$ denote the *maximal robust controlled-invariant set* [31] for player i . Inside Ω^{i*} , player i is guaranteed a strategy to stay within Ω^{i*} regardless of the opponent’s actions. Next, we define the *safe action set* as

$$\mathcal{A}_{\text{safe}}^i(s) := \{a^i \mid P(\Omega^{i*} \mid s, a^i, a^{-i}) = 1, \forall a^{-i} \in \mathcal{A}^{-i}\}.$$

Definition 3 (Perfect Safety Filter [11]). A (robust) safety filter $\phi^i : \mathcal{S} \times \mathcal{A}^i \rightarrow \mathcal{A}^i$ is *perfect* if it satisfies $\phi^i(s, a) \in \mathcal{A}_{\text{safe}}^i(s)$, $\forall a \in \mathcal{A}^i$, $\forall s \in \Omega^i$, and leaves every safe action unchanged: $\phi^i(s, a) = a$, $\forall a \in \mathcal{A}_{\text{safe}}^i(s)$, $\forall s \in \Omega^i$.

Definition 4 (Filtered Game). Given perfect filters $\phi = (\phi^1, \phi^2)$, we define a *filtered game* as

$$\mathcal{G}_\phi := (\bar{\mathcal{S}}, \mathcal{A}^1, \mathcal{A}^2, P_\phi, \bar{r}, \gamma), \quad (5)$$

where $P_\phi(\cdot \mid s, a^1, a^2) := \bar{P}(\cdot \mid s, \phi^1(s, a^1), \phi^2(s, a^2))$.

Since perfect filters leave safe actions unchanged, any payoff improvement a player can achieve by deviating to another safe policy is also achievable in the filtered game. This property allows us to compare the equilibria of a filtered game $\mathcal{G}_\phi$ with those of $\mathcal{G}_{\text{SC}}^{\text{safe}}$, the original SC-MG with each player committed to a safe strategy.

Theorem 1 (Non-Exploitability under Perfect Filtering). *Let $\phi = (\phi^1, \phi^2)$ be perfect filters, and let $\mathcal{G}_{\text{SC}}^{\text{safe}}$ denote the original SC-MG with each player restricted to safe policy classes $\Pi_{\text{safe}}^i := \{\pi^i \in \Pi^i \mid \pi^i(\mathcal{A}_{\text{safe}}^i(s) \mid s) = 1, \forall s \in \Omega^{i*}\}$. For any initial state $s \in \Omega^*$, an NE $(\pi^{1,*}, \pi^{2,*})$ of $\mathcal{G}_\phi$ induces, through the same filters, an executed pair $(\pi_{\text{exec}}^1, \pi_{\text{exec}}^2)$ that is an NE of $\mathcal{G}_{\text{SC}}^{\text{safe}}$. Consequently, both players are non-exploitable in their respective games:*

$$\text{expl}_{\mathcal{G}_{\text{SC}}^{\text{safe}}}(\pi_{\text{exec}}^i)(s) = \text{expl}_{\mathcal{G}_\phi}(\pi^{i,*})(s) = 0. \quad (6)$$

Proof. Let $(\pi^{1,*}, \pi^{2,*})$ be an NE of $\mathcal{G}_\phi$. Suppose a player could improve its payoff by a safe unilateral deviation from the executed pair $(\pi_{\text{exec}}^1, \pi_{\text{exec}}^2)$ in $\mathcal{G}_{\text{SC}}^{\text{safe}}$. Since a perfect filter leaves safe actions unchanged, the deviating policy could be used as a nominal task policy in $\mathcal{G}_\phi$, which achieves the same improvement in $\mathcal{G}_\phi$, contradicting its NE property. Hence, the executed policy pair is an NE of $\mathcal{G}_{\text{SC}}^{\text{safe}}$. The equilibrium payoffs coincide, and Definition 2 ensures zero exploitability for both players in their respective games. $\square$

Theorem 1 shows that in competitive games, the *safety–task separation* does not necessarily sacrifice exploitability when both players are safety-aware. Based on this insight, Sec. V presents an RL framework for synthesizing competitive robot policies using learned robust safety filters.

V. METHOD: LEARNING MINIMALLY EXPLOITABLE ROBOT POLICIES WITH FILTERED RL

In this section, we present our main algorithmic contribution, **Safety to Competence (S2C)**, a two-stage RL

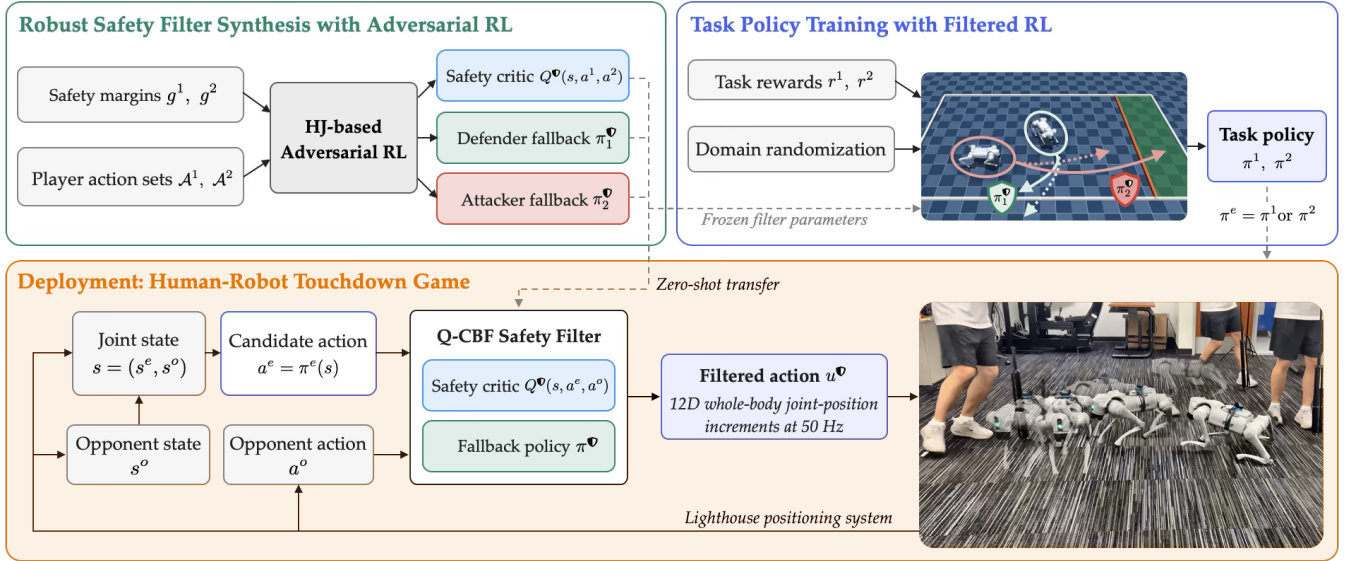


Fig. 3: Overview of the proposed S2C training and deployment pipeline, illustrated with the asymmetric touchdown game.

Algorithm 1 S2C: Safety-Filtered Competitive RL

Input: margins $g^i, g^{i,c}$, task reward r , initial task policy π_{init}^i , robust filter synthesis algorithm FILTER_SYNTHESIS, multi-agent reinforcement learning algorithm MARL

Training Stage 1: Robust Filter Synthesis

- 1: $(Q^V, \pi_1^V, \pi_2^V) \leftarrow \text{FILTER_SYNTHESIS}(g^1, g^{1,c}, g^2, g^{2,c})$
- 2: $\text{ROBUST_FILTER} \leftarrow (Q^V, \pi_1^V, \pi_2^V)$

Training Stage 2: Filtered Multi-Agent RL

- 3: Initialize policy pool: $\mathcal{P} \leftarrow \emptyset$
- 4: Initialize task policies: $(\pi^1, \pi^2) \leftarrow (\pi_{\text{init}}^1, \pi_{\text{init}}^2)$
- 5: **for** $k = 1, 2, \dots$ **do**
- 6: Collect rollouts with ROBUST_FILTER; each player faces its opponent using π^{-i} or, with probability p , a policy in $\mathcal{P}$
- 7: Train π^1, π^2 with MARL
- 8: Periodically update $\mathcal{P} \leftarrow \mathcal{P} \cup \{\pi^1, \pi^2\}$

Deployment

- 9: $\pi^e \leftarrow \pi^1$ or π^2
- 10: $a_t^e \leftarrow \text{ROBUST_FILTER}(s_t, \pi^e)$

framework that trains safe and minimally exploitable robot policies. It sequentially trains a robust safety filter and a task policy with the filter embedded in the environment. Our overall system architecture and algorithmic approach are summarized in Fig. 3 and Algorithm 1. Our framework is modular and agnostic to the specific methods used for filter synthesis, task policy training, and filter deployment. In this paper, we instantiate filter synthesis, task-policy training, and filter deployment with MAGICs [12], IPPO [29], and Robust Q-CBF [3, 15], respectively.

A. Synthesizing Robust Safety Filters

For each player i , we first synthesize its robust safety filter independently of the task objective. We formulate safety as a zero-sum game between the robot and a worst-case opponent: the ego robot seeks to maintain its own safety, while the adversary actively drives it toward failure. This yields a learned *safety fallback policy* π_i^V trained to protect player i against adversarial opponent behavior.

To solve this safety game for general, high-dimensional interactive tasks (e.g., the touchdown game in the Running Example with 62D observation and 12D action spaces), we use MAGICs [12], a game-theoretic adversarial RL algorithm for neural safety synthesis. MAGICs jointly trains a safety fallback π_i^V , a worst-case adversarial policy, and a safety critic Q^V . It is shown to converge to a local minimax solution, providing a scalable approximation to HJ-based robust safety analysis that would yield a perfect safety filter [16]. The fallback π_i^V and critic Q^V form essential parts of the robust safety filter $\phi^i(s, \pi^i)$, which takes a state s and a task policy π^i as inputs, simulates the worst-case opponent action, and returns a filtered task action a^i . We explain the deployment of this robust filter in Sec. V-C.

Running Example: In the touchdown game, both the safety fallback and critic operate with a 62D observation comprising both ego and opponent’s positions of the body frame, body velocities and orientations, axial rotational rates, and angular velocity. The robot action $a^i \in \mathcal{A}^i = [-1, 1]^{12}$ is a normalized joint-position increment command for the 12 actuated leg joints, tracked by a joint-level PD controller. Both the safety fallback and the critic are three-layer MLPs, with hidden layer widths of 512 and 256 units, respectively, both trained with 2048 parallel environments in MuJoCo [32] on a workstation with an RTX 5080 GPU.

B. Learning Task Policies in the Filtered Environment

We train each agent’s task policy π^i in an RL environment that embeds its safety filter ϕ^i , as in the filtered game of Definition 4. To train both agents’ task policies simultaneously, our framework calls for a MARL algorithm, e.g., IPPO [29] or MAPPO [33]. At each training step, each player’s task policy π^i proposes a candidate action a^i , and the simulator executes the action modified by the safety filter. Crucially, the task policies are learned based on the agents’ interaction data with the environment, requiring neither explicit information

on the safety filter nor extra reward terms to account for safety overrides.

To stabilize training and prevent overfitting to a specific opponent strategy, we maintain a *policy pool* $\mathcal{P}$ of historical policy checkpoints [28, 34] and sample opponents from this pool in a fraction of parallel environments. We prioritize opponents with a win rate close to 50% to focus learning on balanced matches.

Running Example: *The task reward consists of a sparse ± 1 touchdown reward and a dense locomotion shaping term that is annealed after 1000 iterations. We train each player’s task policy π^i with IPPO [29]. The task policy uses the same observation space as the safety filter and outputs 12D joint-position targets, which are converted into joint-position increments prior to filtering. Each player’s policy is parameterized by an MLP with three hidden layers of 512, 256, and 128 neurons. Training is conducted across 8192 parallel environments with an opponent-pool sampling probability of $p = 0.35$, running for 5000 iterations.*

C. Deploying Task Policies with Robust Safety Filters

At deployment, each learned task policy π^i is paired with the same robust safety filter ϕ^i used during training. The filter is constructed from the safety critic Q^ϕ and fallback policy π^ϕ learned in Sec. V-A. At every control step, the task policy proposes a candidate action, which is passed through unchanged when it satisfies the learned safety constraint and otherwise corrected by the filter. Using the same filter during training and deployment is critical to our framework. During training, the task policy experiences the consequences of filter interventions directly through the filtered environment and learns to *cooperate with the filter* to minimize exploitability (Theorem 1) by applying task strategies that avoid unnecessary overrides.

Running Example: *For the touchdown game, we approximate the perfect filter in Definition 3 using a Q-CBF, which has been shown to robustly maintain safety while preserving task performance in quadrupedal locomotion [15], with the learned critic Q^ϕ and fallback π^ϕ . In particular, the filter performs a fixed number of gradient-descent steps to search for an action closest to the proposed task action and certified safe by Q^ϕ . If no such action is found, the filter uses the learned fallback policy π_i^ϕ .*

VI. EXPERIMENTS

We use our proposed **S2C** framework to synthesize the safety filter and train touchdown game policies in both symmetric and asymmetric scenarios. We evaluate their safety and competence at scale in simulation, and deploy the robot in both roles (attacker and defender) against a human opponent in a real-world experiment.

Hypotheses. Our experiments test two hypotheses that reflect the key advantages of **S2C**:

- **H1 (Safe competence).** **S2C** yields more competitive policies without sacrificing safety.
- **H2 (Robustness and generalization).** **S2C** generalizes better to out-of-distribution opponents.

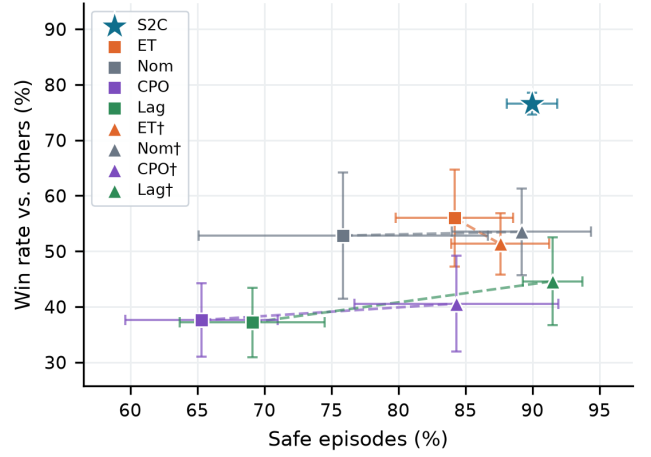


Fig. 4: Safety versus win rate in symmetric touchdown games.

Baselines. We compare **S2C** to four baseline safe RL methods that jointly optimize safety and task objectives.

- **Early termination (ET)** ends the current training episode on a safety violation.
- **Reward penalty (Nom)** adds a penalty to the reward for each safety violation.
- **Constrained policy optimization (CPO)** enforces a violation budget b^i during a trust-region update.
- **PPO-Lagrangian (Lag)** enforces a violation budget through an adaptive multiplier.
- **Filtered baselines (†)** equips each baseline above with the same filter used by **S2C** at deployment.

All baselines are trained with the same IPPO algorithm, network architecture, and training schedule as **S2C**. We also swept additional parameters for the **CPO** and **Lag** baselines.

Metrics. We select checkpoints that achieve the best self-play payoff for all nine methods and use them to form the evaluation pool $\mathcal{B}$. We evaluate robot performance with the following metrics:

- **Safe rate (SR).** The fraction of episodes free of a safety violation indicated by $g_{\text{tot}}^i(s) < 0$.
- **Win rate (WR).** The averaged share of won episodes against opponent methods.
- **Empirical exploitability.** We measure empirical exploitability for the symmetric game ($V^* = 0$). We then evaluate (4) on the empirical games over $\mathcal{B}$ [28] as $\max_{\pi' \in \mathcal{B}} \hat{J}(\pi', \pi)$, where $\hat{J}(\pi, \pi') = (n_{\text{win}}(\pi) - n_{\text{win}}(\pi'))/n$ is the net win rate over n episodes.
- **Elo.** We calculate Elo ratings for symmetric games based on all pool matches using the standard scale factor of 400, with bootstrap confidence intervals.

Simulation values are mean $\pm$ one standard deviation across the five random training seeds of each method.

A. Simulated Touchdown Games

Results. We report the results of the simulated symmetric game (Running Example) in this section. Following the setup in Sections V-A and V-B, we evaluate all nine methods in

TABLE I: Round-robin results in simulated touchdown games.

Method	Asymmetric game		Symmetric game	
	WR (%) $\uparrow$	SR (%) $\uparrow$	WR (%) $\uparrow$	SR (%) $\uparrow$
S2C	77.5 $\pm$ 2.8	87.9 $\pm$ 2.4	76.5 $\pm$ 2.0	90.0 $\pm$ 1.9
ET	68.1 $\pm$ 12.5	69.7 $\pm$ 18.5	56.0 $\pm$ 8.7	84.2 $\pm$ 4.4
Nom	51.4 $\pm$ 6.3	68.9 $\pm$ 8.6	52.8 $\pm$ 11.4	75.9 $\pm$ 10.8
CPO	22.3 $\pm$ 2.3	46.9 $\pm$ 5.2	37.6 $\pm$ 6.6	65.3 $\pm$ 5.7
Lag	63.0 $\pm$ 12.9	67.6 $\pm$ 18.2	37.2 $\pm$ 6.3	69.1 $\pm$ 5.4
ET $\dagger$	46.0 $\pm$ 3.4	69.1 $\pm$ 7.6	51.3 $\pm$ 5.5	87.6 $\pm$ 3.6
Nom $\dagger$	41.6 $\pm$ 2.4	76.6 $\pm$ 3.0	53.5 $\pm$ 7.8	89.2 $\pm$ 5.2
CPO $\dagger$	30.6 $\pm$ 2.3	71.4 $\pm$ 4.4	40.5 $\pm$ 8.6	84.3 $\pm$ 7.6
Lag $\dagger$	49.5 $\pm$ 5.0	75.7 $\pm$ 5.4	44.6 $\pm$ 7.9	91.5 $\pm$ 2.2

Each row corresponds to 31200 games. WR: win rate. SR: safe rate.

a round-robin tournament. As shown in Fig. 2 and Table I, **S2C** achieves the highest win rate across all pairwise matches with an average of 76.5%. **S2C** also achieves the highest Elo rating (139), significantly exceeding the second-place competitor, **Nom** $\dagger$ (37) by more than 100 points. In terms of empirical exploitability, the strongest pool opponent achieves a net win rate of 0.11 against **S2C**, compared to 0.54–0.75 against the baselines. Because the evaluation pool includes unfiltered baselines, this indicates that **S2C** remains hard to exploit even by reckless opponents not committed to using safe policies. Moreover, **S2C** achieves a high safe rate of 90.0%, close to that of **Lag** $\dagger$. These results validate **H1**.

Filtered ablation. We conduct an ablation study by deploying the learned safety filter in **S2C** for the baselines at runtime (see Figs. 2 and 4). The filter enhances safety as expected: the safe rates of filtered baselines improve across the board by up to 22.4 percentage points (pp), bringing both **Lag** $\dagger$ and **Nom** $\dagger$ to safety levels comparable to **S2C**. However, this added safety does not transfer to a notable gain in the robot’s competitive competence: baseline win rates change by at most 7.4pp, remaining substantially below **S2C**, and their exploitability stays high. This is because the baseline policies were trained in unfiltered environments, and therefore did not learn to cooperate with the safety filter. In some states, they repeatedly propose unsafe actions, so the filter intervenes frequently and thereby degrades the task performance. This ablation shows that a safety filter alone does not grant strategic competence; rather, competence must be co-learned within the filtered environment as in **S2C**.

Qualitative observations. During training, we observe that competitive tactics such as feints and abrupt stop-and-turn maneuvers (see Fig. 1) emerge noticeably earlier in **S2C** than in the baseline methods. We attribute this to the safety filter intervening to brake or stabilize the robot whenever it approaches danger. It spares the policy from having to discover constraint-satisfying strategies through trial-and-error failures, allowing it to focus its exploratory effort more directly on task objectives. In contrast, because the baselines must map safety boundaries through catastrophic failures, individual runs often collapse into overly conservative play or stereotyped gaits that specific opponents can easily exploit during evaluation.

TABLE II: Hardware trials against a human opponent.

Robot attacking				
Method	TDR (%) $\uparrow$	SR (%) $\uparrow$	HPS (m/s)	HPA (m/s ²)
S2C	70.0	80.0	2.14 $\pm$ 0.36	5.83 $\pm$ 1.30
ET	50.0	60.0	1.86 $\pm$ 0.26	3.17 $\pm$ 1.03
Nom	30.0	30.0	1.46 $\pm$ 0.19	2.35 $\pm$ 1.23
CPO	0.0	0.0	1.30 $\pm$ 0.38	2.37 $\pm$ 0.73
Lag	20.0	20.0	0.74 $\pm$ 0.43	1.04 $\pm$ 0.58
Robot defending				
Method	TOR (%) $\uparrow$	SR (%) $\uparrow$	HPS (m/s)	HPA (m/s ²)
S2C	70.0	90.0	2.40 $\pm$ 0.38	4.31 $\pm$ 1.14
ET	60.0	80.0	2.07 $\pm$ 0.16	4.34 $\pm$ 1.21
Nom	40.0	50.0	1.22 $\pm$ 0.15	2.46 $\pm$ 0.79
CPO	0.0	100.0	1.30 $\pm$ 0.22	2.15 $\pm$ 0.70
Lag	10.0	60.0	1.10 $\pm$ 0.21	1.85 $\pm$ 0.66

Each row corresponds to ten trials. TDR/TOR: touchdown/timeout rate. SR: safe rate. HPS: human’s peak speed. HPA: human’s peak acceleration.

B. Hardware Stress Tests Against Human Opponents

For the real-world evaluation, we consider an asymmetric attack–defense game where the attacker needs to reach a target line within a time limit to win. This asymmetry induces more intense interactions, such as frequent close contact between the two players. The asymmetric game is trained and evaluated in simulation under the same protocol as the symmetric one, where **S2C** attains both the highest win rate and the highest safe rate (Table I). We further conduct this game in hardware experiments and evaluate the policies against a human opponent.

Hardware System Setup. We use an HTC Vive Lighthouse system to localize both the Unitree Go2 quadrupedal robot and the human. This system consists of two base stations placed at the diagonal corners of the field and two SteamVR trackers mounted on the robot and its human opponent. Policy inference and the safety filter run on the Jetson Nano at 50 Hz. All five methods are trained under the same domain-randomization strategy. We evaluate **S2C** and all four baselines for ten rounds in each attacker/defender role.

Results. Table II summarizes task completion, safe rates, and opponent kinematics across the hardware trials. We record the following numbers as measures of human competence:

- **Human peak speed (HPS).** The human opponent’s peak speed in m/s.
- **Human peak acceleration (HPA).** The human opponent’s peak acceleration in m/s².

Across both roles, **S2C** achieves the highest task completion (70%) while maintaining high safe rates (80% as attacker and 90% as defender). In comparison, **ET** and **Nom** compromise safety under competitive pressure, with safe rates dropping to 30%–60% when attacking. Meanwhile, **CPO** fails to reconcile the two objectives: it causes a safety violation on every attacking trial (zero touchdowns), and achieves zero defensive violations only by passively traversing the field, unable to contest the human attacker in close proximity. From the HPS and HPA values, **S2C** has to respond to the most demanding human movements with the highest mean

peak speed in both roles, the highest attacking acceleration (5.83 m/s^2), and defending acceleration comparable to that of **ET**. Those human strategies were not encountered during training. These trials provide evidence that **S2C** is more robust against out-of-distribution opponents and support **H2**.

VII. CONCLUSION

Limitations. While our analysis assumes perfect safety filters, deriving formal guarantees for ϵ -Nash equilibria under approximation errors stemming from imperfect filters and task policies remains an open research question. We also see an exciting opportunity to integrate S2C with latent-space safety filters with visual feedback [35] for competitive robot interactions in unstructured, open-world environments.

Summary. In this paper, we introduced S2C, a two-stage framework that decouples safety from competitive task-policy learning in multi-robot interactions. We proved that under perfect safety filtering, equilibrium policies of the filtered game remain non-exploitable within the safe policy class of the original safety-critical Markov game. We validated S2C on a quadruped touchdown game in both simulation and hardware tests. The results demonstrate that S2C achieves the highest task performance and the lowest exploitability across the evaluation pool while maintaining high safe rates. Our work establishes a principled framework for learning safe and minimally exploitable robot policies for competitive interactions.

REFERENCES

- [1] E. Kaufmann, L. Bauersfeld, A. Loquercio, et al. “Champion-level drone racing using deep reinforcement learning”. *Nature* 620.7976 (2023), pp. 982–987.
- [2] P. R. Wurman, S. Barrett, K. Kawamoto, et al. “Outracing champion Gran Turismo drivers with deep reinforcement learning”. *Nature* 602.7896 (2022), pp. 223–228.
- [3] D. D. Oh, J. Lidard, H. Hu, et al. “Safety with agency: Human-centered safety filter with application to AI-assisted motorsports”. *Robotics: Science and Systems*. 2025.
- [4] T. Haarnoja, B. Moran, G. Lever, et al. “Learning agile soccer skills for a bipedal robot with deep reinforcement learning”. *Science Robotics* 9.89 (2024), eadi8022.
- [5] A. Bajcsy, A. Loquercio, A. Kumar, and J. Malik. “Learning vision-based pursuit-evasion robot policies”. *IEEE International Conference on Robotics and Automation*. 2024, pp. 9197–9204.
- [6] J. García and F. Fernández. “A comprehensive survey on safe reinforcement learning”. *Journal of Machine Learning Research* 16.42 (2015), pp. 1437–1480.
- [7] C. Tessler, D. J. Mankowitz, and S. Mannor. “Reward constrained policy optimization”. *International Conference on Learning Representations*. 2019.
- [8] E. Altman. “Constrained Markov decision processes”. Tech. rep. RR-2574. INRIA, 1995.
- [9] J. Achiam, D. Held, A. Tamar, and P. Abbeel. “Constrained policy optimization”. *International Conference on Machine Learning. Proceedings of Machine Learning Research*. 2017, pp. 22–31.
- [10] H. Sun, Z. Xu, M. Fang, et al. “Safe exploration by solving early terminated MDP”. 2021. arXiv: 2107.04200.
- [11] K.-C. Hsu, H. Hu, and J. F. Fisac. “The safety filter: A unified view of safety-critical control in autonomous systems”. *Annual Review of Control, Robotics, and Autonomous Systems* 7.1 (2024).
- [12] J. Wang, H. Hu, D. P. Nguyen, and J. F. Fisac. “MAGICS: Adversarial RL with minimax actors guided by implicit critic Stackelberg for convergent neural synthesis of robot safety”. *Algorithmic Foundations of Robotics*. 2024.
- [13] D. P. Nguyen, K.-C. Hsu, W. Yu, et al. “Gameplay filters: Robust zero-shot safety through adversarial imagination”. *Conference on Robot Learning*. PMLR. 2025, pp. 387–407.
- [14] J. Seo, K. Nakamura, and A. Bajcsy. “Uncertainty-aware latent safety filters for avoiding out-of-distribution failures”. *Conference on Robot Learning*. 2025, pp. 4442–4472.
- [15] D. D. Oh, D. P. Nguyen, H. Hu, and J. F. Fisac. “Synthesis and deployment of maximal robust control barrier functions through adversarial reinforcement learning”. *IEEE Conference on Decision and Control*. 2026.
- [16] D. D. Oh, D. P. Nguyen, H. Hu, and J. F. Fisac. “Provably optimal reinforcement learning under safety filtering”. *Conference of the International Association for Safe and Ethical Artificial Intelligence*. 2026.
- [17] K. P. Wabersich and M. N. Zeilinger. “Predictive control barrier functions: Enhanced safety mechanisms for learning-based control”. *IEEE Transactions on Automatic Control* 68.5 (2022), pp. 2638–2651.
- [18] A. D. Ames, S. Coogan, M. Egerstedt, et al. “Control barrier functions: Theory and applications”. *European Control Conference*. 2019, pp. 3420–3431.
- [19] A. Robey, H. Hu, L. Lindemann, et al. “Learning Control Barrier Functions from Expert Demonstrations”. *IEEE Conference on Decision and Control*. 2020, pp. 3717–3724.
- [20] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin. “Hamilton–Jacobi reachability: A brief overview and recent advances”. *IEEE Conference on Decision and Control*. 2017, pp. 2242–2253.
- [21] J. F. Fisac, N. F. Lugovoy, V. Rubies-Royo, et al. “Bridging Hamilton–Jacobi Safety Analysis and Reinforcement Learning”. *IEEE International Conference on Robotics and Automation*. 2019, pp. 8550–8556.
- [22] S. Bansal and C. J. Tomlin. “DeepReach: A deep learning approach to high-dimensional reachability”. *IEEE International Conference on Robotics and Automation*. 2021, pp. 1817–1824.
- [23] K.-C. Hsu, D. P. Nguyen, and J. F. Fisac. “ISAACS: Iterative soft adversarial actor-critic for safety”. *Learning for Dynamics and Control Conference*. 2023.
- [24] T. Başar and G. J. Olsder. “Dynamic noncooperative game theory”. Philadelphia: SIAM, 1998.
- [25] D. Fridovich-Keil, E. Ratner, L. Peters, et al. “Efficient Iterative Linear-Quadratic Approximations for Nonlinear Multi-Player General-Sum Differential Games”. *IEEE International Conference on Robotics and Automation*. 2020, pp. 1475–1481.
- [26] J. Lidard, H. Hu, A. Hancock, et al. “Blending data-driven priors in dynamic games”. *Robotics: Science and Systems* (2024).
- [27] R. Lowe, Y. Wu, A. Tamar, et al. “Multi-agent actor-critic for mixed cooperative-competitive environments”. *Advances in Neural Information Processing Systems* 30 (2017).
- [28] M. Lanctot, V. F. Zambaldi, A. Gruslys, et al. “A unified game-theoretic approach to multiagent reinforcement learning”. *Advances in Neural Information Processing Systems* 30 (2017).
- [29] C. Schroeder de Witt, T. Gupta, D. Makovychuk, et al. “Is independent learning all you need in the StarCraft multi-agent challenge?” 2020. arXiv: 2011.09533.
- [30] L. S. Shapley. “Stochastic games”. *Proceedings of the National Academy of Sciences* 39.10 (1953), pp. 1095–1100.
- [31] F. Blanchini. “Set invariance in control”. *Automatica* 35.11 (1999), pp. 1747–1767.
- [32] E. Todorov, T. Erez, and Y. Tassa. “MuJoCo: A physics engine for model-based control”. *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, pp. 5026–5033.
- [33] C. Yu, A. Velu, E. Vinitzky, et al. “The surprising effectiveness of PPO in cooperative multi-agent games”. *Advances in Neural Information Processing Systems* 35 (2022), pp. 24611–24624.
- [34] J. Heinrich, M. Lanctot, and D. Silver. “Fictitious self-play in extensive-form games”. *International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. 2015, pp. 805–813.
- [35] K. Nakamura, L. Peters, and A. Bajcsy. “Generalizing safety beyond collision-avoidance via latent-space reachability analysis”. *Robotics: Science and Systems*. 2025.