Defending Against Unknown Corrupted Agents: Reinforcement Learning of Adversarially Robust Nash Equilibria

Andi Nika\textsuperscript{1} 
andinika@mpi-sws.org

Jonathan Nöther\textsuperscript{1,2} 
s8jonoet@stud.uni-saarland.de

Adish Singla\textsuperscript{1} 
adishs@mpi-sws.org

Goran Radanović\textsuperscript{1} 
gradanovic@mpi-sws.org

\textsuperscript{1} Max Planck Institute for Software Systems (MPI-SWS), Saarbrücken, Germany
\textsuperscript{2} University of Saarland, Saarbrücken, Germany

Abstract

We consider a Multi-agent Reinforcement Learning (MARL) setting, in which an attacker can arbitrarily corrupt any subset of up to \( k \) out of \( n \) agents at deployment. Our goal is to design agents that are robust against such an attack, by accounting for the presence of corrupted agents at test time. To that end, we introduce a novel solution concept, the Adversarially Robust Nash Equilibrium (ARNEQ), and provide theoretical proof of its existence in general-sum Markov games. Furthermore, we introduce a proof-of-concept model-based approach to computing it and theoretically prove its convergence under standard assumptions. We also present a practical approach called Adversarially Robust Training (ART), an independent learning algorithm based on stochastic gradient descent ascent. Our experiments in both cooperative and mixed cooperative-competitive environments demonstrate ART’s effectiveness and practical value in enhancing MARL resilience against adversarial behavior.

1 Introduction

The growing prevalence of automated systems has facilitated a fertile ground for implementing the celebrated multi-agent reinforcement learning (MARL) framework to solve a wide range of important problems. Prominent examples include finance [Shavandi and Khedmati, 2022, Lee et al., 2007], sensor networks [Cortes et al., 2004, Choi et al., 2009], autonomous vehicles [Shalev-Shwartz et al., 2016, Zhou et al., 2020, Palanisamy, 2020] and gaming [Vinyals et al., 2019, Perolat et al., 2022].

One of the main underlying assumptions of this practical framework has been the inherently rational and selfish nature of the learning agents, according to which their sole objective is to optimize their utilities, an assumption stemming from game theory [Osborne and Rubinstein, 1994] and economics since the time of Smith [1776]. Indeed, the overwhelming majority of MARL successes have been made by algorithms that are designed to optimize their objectives.

However, this assumption has been challenged by different lines of research, only sporadically in the past, but increasingly more in recent years. For example, Chen et al. [2023] study the problem of robustness of distributed RL systems against Byzantine attacks, where a fraction of agents can report fake data to the center. Another line of research, which has gained a lot of popularity recently, studies the problem of adversarial policies in MARL [Guo et al., 2021], where an agent faces an
Attacker selects up to \( k \) policies

Policies are deployed

Robust training

Attacker modifies selected policies

Figure 1: Attack-defense interaction is depicted. First, the agents are trained without any adversarial threat. Then, the attacker observes all the trained policies of the agents and arbitrarily picks any up to \( k \) of them. Once the chosen policies are arbitrarily perturbed, they are deployed. Our goal is to train policies that remain robust under attack.

adversarial opponent trained in a zero-sum Markov game regime. Eliaz [2002] considers the problem of implementing games with potentially faulty agents, that is, agents who do not necessarily aim at optimizing their utilities but might behave erratically due to some system malfunction. Furthermore, Babaioff et al. [2007] consider congestion games with malicious players and study the game-theoretic properties of such games.

The shared underlying insight among these various lines of work is that in reality, certain agents could exhibit unpredictable behavior, stemming from system malfunctions or unanticipated adversarial objectives. This insight is further evidenced in various safety-critical applications of MARL, where the presence of such agents might seriously deteriorate their performance. For example, in traffic management [Kuyer et al., 2008] and autonomous driving systems, some agents, be it autonomous or human, might cause traffic jams or fatal accidents due to their erratic behavior, while in distributed energy systems [Roesch et al., 2020] they might compromise the stability and reliability of the power grid.

Be that as it may, the aforementioned previous work lacks a simultaneously practical and conceptually general treatment of the problem of malicious agents in multi-agent systems. While Eliaz [2002] proposes an interesting solution concept, that of a faulty-tolerant Nash equilibrium, their focus is on the implementation problem, not a learning one, so the question of efficiently computing such an equilibrium is left open. The line of work initiated by Gleave et al. [2020] on adversarial policies focuses on two-player games, and their considerations are mostly from the attack’s perspective. Chen et al. [2023] consider distributed RL systems, where agents share the same goal, while Babaioff et al. [2007] study the properties of a specific class of games with malicious players, with fixed identities.

Motivated by the above shortcomings, we are interested in providing a formulation of the problem that captures a general attack model. In particular, we are interested in a model that: (i) involves \( n \)-player Markov games; (ii) the attacker can manipulate any up to \( k \) out of \( n \) trained policies; (iii) the attacker can arbitrarily modify the controlled policies. Our aim is to find an efficient defense that takes into account such an attack model and serves as a robust training procedure for the learning agents.

To that end, we formulate a general-sum Markov game with \( n \) players, whereby an attacker is given access to all agents’ policies at test-time, previously trained on a clean environment. The attacker then arbitrarily picks and perturbs any up to \( k < n \) of them, at which point all policies are deployed. Ultimately, we are interested in devising a robust training procedure that accounts for the potential attack at test time and can serve as a defense against it. The attack-defense model is described in Figure 1. For a detailed discussion on other related work, see Appendix A. Below we describe our main contributions.

• First, in the same spirit as [Eliaz, 2002, Zhang et al., 2020], we introduce a new notion of equilibrium, the \( k \)-Adversarially Robust Nash Equilibrium (ARNEQ), designed to be robust against the worst-case arbitrary perturbations of any \( k \) out of \( n \) policies, and show its existence in general-sum Markov games.

• Second, we propose a model-based procedure to compute such a strategy and show that it converges to an ARNEQ under standard assumptions on the game structure.

• Third, motivated by gradient-based decentralized methods in Markov games, we propose Adversarially Robust Training (ART) for MARL, an efficient gradient-based learning algorithm, that can be run independently among all players at training time.

• Finally, we provide extensive experiments on both cooperative and mixed environments that showcase the efficiency of ART as a robust defense. Our results show that ART converges to a stable joint strategy and is able to substantially improve performance, which can be clearly observed in our qualitative results.
2 Problem Formulation

In this section, we first introduce the relevant notation, then give the necessary background of the problem, after which we formally introduce our problem.

Notation. As usual $\langle \cdot, \cdot \rangle$ will denote the inner product between two vectors from the same vector space, $[n]$ denotes the set of natural numbers up to and including $n$. We denote by $\| \cdot \|$ the Euclidean norm and by $\Delta(X)$ the set of probability simplices on the set $X$.

2.1 Preliminaries

Let $G = (\mathcal{N}, S, A_1 \times \ldots \times A_n, R_1, \ldots, R_n, \mathcal{P}, \gamma, \mu)$ be a Markov game. Here $\mathcal{N} = [n]$ represents the set of $n$ players, $S$ denotes the state space with cardinality $S$, $A := A_1 \times \ldots \times A_n$ is the joint action space with $A_i$ denoting the action space of player $i$ with cardinality $A_i$. The reward of player $i$ is a function of the state and joint action, i.e., $R_i : S \times A \rightarrow [0, 1]$. The transition kernel is given by $\mathcal{P} : S \times A \rightarrow \Delta(S)$, with $\mathcal{P}(s', s, a)$ denoting the probability that the game transitions into state $s'$ given that joint action $a$ is taken in state $s$. The discount factor is denoted by $\gamma \in [0, 1)$, and $\mu \in \Delta(S)$ denotes the initial state distribution. Finally, we will denote by $\mathcal{N}_i$ the set of subsets of cardinality no greater than $k$ of $X' \setminus \{i\}$, and we let $\Omega_i = \Delta(\mathcal{N}_i)$.

Stationary policies are mappings from states to distributions over actions. Formally, we let $\pi_i : S \rightarrow \Delta(A_i)$ be a stationary policy for player $i$, lying in the policy space $\Pi_i$. Moreover, we denote by $\Pi = \Pi_1 \times \ldots \times \Pi_n$ the joint policy space of all players, with elements denoted by $\pi$. In Sections 4 and 5, we will consider neural policy classes. Given action $a$, policy $\pi$ and a set $K \in \mathcal{N}$, we define $a_K = (a_j)_{j \in K}$ and $a_{-K} = (a_j)_{j \notin K}$ for action $a$ and $\pi_K = (\pi_j)_{j \in K}$ and $\pi_{-K} = (\pi_j)_{j \notin K}$ for policy $\pi$.

The value function represents the expected discounted cumulative reward of a given player with respect to a given joint policy, starting from a given state. Formally, given policy $\pi \in \Pi$ and state $s \in S$, the value function with respect to player $i$ is given by $V^i(\pi) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t R_i(s_t, a_t) | s_0 = s, \pi, \mathcal{P} \right]$, where the sequence $s_0, a_0, s_1, a_1, \ldots$ denotes the traversed state-action tuples when the initial state is $s$, actions are taken using $\pi$ and the transitions follow $\mathcal{P}$. Furthermore, we let $V^\mu_i(\pi) = \mathbb{E}_{s,a} [V^i(\pi^\mu)]$ denote the value function with respect to the initial state distribution. The value function satisfies the Bellman equation $V^\mu_i(\pi) = \sum_{s \in \mathcal{S}} q_i(s) \sum_{a \in A} \prod_{j \in \mathcal{N}_i} \pi_j(a_j | s) (B V_i)(s, a)$, with $B$ representing the Bellman operator acting on $V_i$, defined as $(B V_i)(s, a) := R_i(s, a) + \gamma \mathcal{P}(s, a, V_i(\pi))$, where $V_i(\pi)$ denotes the $S$-dimensional vector with values $V^i_1(\pi)$ and $\mathcal{P}(s, a)$ denotes the $S$-dimensional vector with entries $\mathcal{P}(s', s, a)$, for all $s' \in S$. Next, we formally define the notion of a Nash equilibrium.

Definition 1 A joint policy $\pi^*$ is said to be a Nash equilibrium (NE) strategy if no player can be better off by deviating from it. Formally, we have $V^i_1(\pi^*) \geq V^i_1(\pi_i', \pi_{-i})$, for all $\pi_i' \in \Pi_i$ and $i \in \mathcal{N}$. Here $\pi_{-i}$ denotes the joint policy of the players other than $i$.

Further, a joint strategy $\pi^*$ is said to be an $\epsilon$-approximate NE strategy if $V^i_1(\pi^*) \geq V^i_1(\pi_i', \pi_{-i}) - \epsilon$, for all $\pi_i' \in \Pi_i$ and $i \in \mathcal{N}$. If an NE joint strategy is stochastic, it is said to be a mixed NE strategy.

Fink [1964] showed that a mixed NE strategy always exists in any given $n$-player Markov game.

2.2 Attack Model

As already described in Section 1, we consider test-time attacks, whereby an attacker is allowed access to all $n$ trained policies and can arbitrarily pick and manipulate any up to $k < n$ of them. It’s important to highlight that the attacker’s ability to select any subset of $k$ trained policies adds an additional layer of complexity to the defense problem. Any information about the attack is necessary to obtain an efficient defense against it. In our setting, the learning agents are unaware of who among them will be attacked at test time, and so, do not know a priori which agents to treat as adversarial.

2.3 Defense Objective

Our focus in this paper is on the defense front. In particular, we are interested in devising training procedures that take into account the potential future threat at deployment. At training time, there is
no attack present. The only information the agents have about the attack is the parameter $k$ and its realization at test time. Thus, our aim is to propose a training procedure for each individual agent, taking into account this parameter, and the fact that any other agent may behave adversarially towards it. In the next section, we will introduce a new solution concept that is designed to capture the nature of the attack and provide such a robust defense.

3 Theoretical Results

In this section, we first formulate our solution concept, namely, the adversarially robust Nash equilibrium. Then, we show its existence in general-sum Markov games. Finally, we provide a model-based approach to its computation and prove its convergence.

3.1 The Adversarially Robust Nash Equilibrium

In this section we introduce our solution concept which is tailored to our attack model and defense objective. Our goal here is to provide the agents with a defense strategy that they can use independently of the other agents. Similar to robust MARL with model uncertainty [Zhang et al., 2020], we consider robustness in MARL under attack uncertainty. The best that player $i$ can do with the given information is to solve the worst-case problem, i.e. to be robust against the worst-case attacker (in terms of the agents under attack and the objectives). To that end, we propose the Adversarially Robust Nash Equilibrium (ARNEQ), defined as follows.

**Definition 2** A joint policy $\pi^*$ is said to be an ARNEQ if there exist $S$-dimensional vectors $V_1^*, \ldots, V_n^*$ such that, for every player $i \in N$ and state $s \in S$, we have

$$
\pi^*_i(\cdot|s) \in \arg \max_{\pi_i \in \Pi_i} \min_{\omega_i \in \Omega_i} \mathbb{E}_{K \sim \omega_i} \left[ \sum_{a \in A} \pi_i(a_i|s) \prod_{j \in K} \tilde{\pi}_j^i(a_j|s) \prod_{l \notin K \cup \{i\}} \pi_l^*(a_l|s)(BV^*_i)(s, a) \right],
$$

where we denote by $\tilde{\pi}_j^i$ the joint adversarial policy $(\tilde{\pi}_j^i)_{j \neq i}$ with respect to player $i$, where $\tilde{\pi}_j^i \in \Pi_j$, for each $j \neq i$, and $\Omega_i = \Delta(N_i)$. 

Before we proceed any further, it is important to point out that this notion of equilibrium does not correspond to that in zero-sum Markov games, or even adversarial team Markov games. In fact, the game induced by the defensive behavior of the agents is not even a static game which all agents share with each other. To see this, consider the following example.

**Example 1** Let $n = 3$ and $k = 1$. When computing an ARNEQ, each agent has to consider its worst-case adversary. Suppose that the worst-case adversary of agent 1 may be agent 2, while the worst-case adversary of agent 2 may be agent 3. So agent 1 and agent 2 are not playing the same game, since the benign agents and worst-case adversarial agents for each of them are different.

We will first show that such an equilibrium always exists in general-sum Markov games. We defer the proof of the result to the Appendix (see the Supplementary Material).

**Theorem 1** Let $G$ be a finite general-sum Markov game with $n$ players, that is, suppose $\max\{n, S, A\} < \infty$, where $A = \max_{i \in [n]} |A_i|$, and let $k < n$. Assume that an arbitrary subset of at most $k$ players use arbitrarily modified utilities, and assume that the mixed strategies of all players lie in compact sets. Then, an ARNEQ, as in Definition 2, exists.

The next natural question is how to find such an equilibrium. In the next section, we will focus on devising a model-based centralized approach to computing an ARNEQ and prove its convergence.

3.2 Adversarially Robust Nash Q-Learning

Nash Q-Learning [Hu and Wellman, 2003] is a model-based approach to finding the Nash equilibrium of a Markov game. Essentially, each player maintains estimates of the Q-values of every player$^1$ and then computes the Nash equilibrium of the stage game. The Q-values are then updated based on the

$^1$We will use player and agent interchangeably.
outcome of applying the so-called Nash operator (analogous to the Bellman operator for Markov games) on the current Q-values. This method leverages a celebrated result from Filar and Vrieze [2012] which links the value functions that correspond to the Nash equilibrium of the entire Markov game with those corresponding to each individual stage game, which will be defined later.

First, we need to define an operator that is appropriate for our setting and satisfies the recursive property analogous to the Bellman operator. Based on Definition 2, we define the following update sequence, for any $i \in \mathcal{N}$, state $s \in S$ and $t \geq 0$:

$$\bar{V}_{i}^{t+1}(s) = \max_{\pi_i} \min_{\omega_i \in \Omega_i} \mathbb{E}_{K \sim \omega_i} \left[ \sum_{a \in A} \pi_i(a|s) \pi_{(K \cup \{i\}))(a -(K \cup \{i\})|s) \hat{\pi}_K^{i}(a_K|s)(B\bar{V}_{i}^{t})(s, a) \right],$$

where $\hat{\pi}_K^{i}$ denotes the joint benign policy computed at step $t$. Similarly, we define the Bellman backup for the Q-values with respect to an ARNEQ policy $\pi^*$, for any state-action tuple $(s, a)$, as follows, for given learning rates $\alpha_t \in [0, 1]$:

$$\bar{Q}_{i}^{t}(s, a) = \mathcal{R}_i(s, a) + \gamma \sum_{s' \in S} \mathcal{P}(s'|s, a) \mathbb{E}_{K \sim \omega_i} \left[ \sum_{a' \in A} \pi_i(a'|s') \pi_{(K \cup \{i\})(a' -(K \cup \{i\})|s') \hat{\pi}_K^{i}(a_K'|s') \bar{Q}_{i}^{t}(s', a') \right],$$

where $\omega_i^*$ and $\hat{\pi}_K^{i,*}$ are the adversarial counterparts of the ARNEQ policy of player $i$, and $\hat{\pi}_K^{i,*} = (\hat{\pi}_K^{i,*})_{i \in K}$. At this point, we need to introduce another relevant definition – an instantiation of the ARNEQ for a stage game defined in terms of the Q-values only for a given state. Our procedure will iteratively compute it for each encountered state.

**Definition 3** Fix state $s \in S$. Given the Q-values $\bar{Q}_{i}^{t}(s, a)$, for all $i \in \mathcal{N}$ and $a \in A$, the joint tuple $(\pi_i, \hat{\pi}_K^i, \omega_i)_{i \in \mathcal{N}}$ is said to be a stage ARNEQ with respect to state $s$ if, for every $i \in \mathcal{N}$, we have

$$(\pi_i, \hat{\pi}_K^i, \omega_i) = \arg \max_{\pi_i, \hat{\pi}_K^i, \omega_i} \min_{\pi_i, \omega_i} \mathbb{E}_{K \sim \omega_i} \left[ \sum_{a \in A} \pi_i(a|s) \pi_{(K \cup \{i\})(a -(K \cup \{i\})|s) \hat{\pi}_K^i(a_K|s) \bar{Q}_{i}^{t}(s, a) \right].$$

Now we are ready to formulate the update rule for our procedure as follows:

$$\bar{Q}_{i}^{t+1}(s_t, a_t) = (1 - \alpha_t)\bar{Q}_{i}^{t}(s_t, a_t) + \alpha_t \left( \mathcal{R}_i(s_t, a_t) + \gamma \text{ARNEQ} \left( \bar{Q}_{i}^{t}(s_{t+1}) \right) \right), \quad (1)$$

where

$$\text{ARNEQ}(\bar{Q}_{i}^{t}(s_{t+1})) := \mathbb{E}_{K \sim \omega_i} \left[ \sum_{a \in A} \pi_{(K \cup \{i\})(a -(K \cup \{i\})|s_t) \hat{\pi}_K^i(a_K|s_t) \bar{Q}_{i}^{t}(s_{t+1}, a') \right],$$

with $\bar{Q}_{i}^{t}(s_{t+1}) = [\bar{Q}_{i}^{t}(s_{t+1}, a)]_{a \in A}$. Here, $a_t$ denotes the joint action taken at time step $t$ in state $s_t$ by all the agents, $\pi^t$ denotes the stage game ARNEQ joint policy computed with values $(\bar{Q}_{i}^{t})_{i \in \mathcal{N}}$.

Before stating the main result of this section, we first state the assumptions under which it holds. They are standard in similar model-based approaches to equilibrium computation Zhang et al. [2020], Hu and Wellman [2003], Yang et al. [2018].

**Assumption 1** Each state and action tuple has been visited infinitely often.

**Assumption 2** The rates $\alpha_t$ satisfy, for every $t \geq 0$:

- $0 \leq \alpha_t < 1$, $\sum_{t \geq 1} \alpha_t = \infty$ and $\sum_{t \geq 0} \alpha_t^2 < \infty$.
- $\alpha_t = 0$ for any $(s, a) \neq (s_t, a_t)$, that is, we only update the Q-values corresponding to traversed state-actions.

**Assumption 3** For each stage game, one of the following conditions holds:
• A stage ARNEQ is also a global optimum, that is, for any \( i \in N \) and \( \pi(\cdot|s) \), we have

\[
E_{K \sim \omega} \left[ \sum_{a \in A} \pi^t_K (a-K|s) \widehat{Q}_i^t(s, a) \right] \geq E_{K \sim \omega} \left[ \sum_{a \in A} \pi_K (a-K|s) Q_i^t(s, a) \right],
\]

for any \( \pi \in \Pi \), \( \pi_i \in \Pi_{-i} \) and \( \omega_i \in \Omega_i \).

• A given player’s payoff is increased if other benign players or its attacker deviate, that is, for any \( i \in N \) we have

\[
E_{K \sim \omega} \left[ \sum_{a \in A} \pi^t_K (a-K|s) \widehat{Q}_i^t(s, a) \right] \leq E_{K \sim \omega} \left[ \sum_{a \in A} \pi^t_i (a_i|s) \pi^t_{-(K \cup \{i\})} (a_{-(K \cup \{i\})}|s) Q_i^t(s, a) \right],
\]

for all \( \pi' \in \Pi \), \( \pi'_i \in \Pi_{-i} \) and \( \omega'_i \in \Omega_i \).

Now we are ready to state the result. Its proof is deferred to the Appendix (see the Supplementary Material).

**Theorem 2** Under Assumptions 1, 2 and 3 (formally stated in the Appendix), the Adversarially Robust Nash Q-Learning procedure as given in Equation equation 1 converges to \( \widehat{Q}_i \), for every player \( i \in N \).

**Proof sketch.** The main ingredient of the proof is utilizing a previous result from [Hu and Wellman, 2003] which states that, if Assumptions 1 and 2 hold, and a given operator on the Q-functions is a contraction, then the procedure described above converges to an equilibrium. So the only thing to prove is that the ARNEQ operator is a contraction. We use Assumption 3 to that end, by separately considering both cases of the assumption. With this, all the conditions of the utilized result are satisfied, and thus we conclude convergence to an ARNEQ. □

This result completes the theoretical characterization of our problem. Although the proposed procedure is simple and intuitive, with the crucial benefit of satisfying strong theoretical guarantees, there are also several unfortunate drawbacks associated with it. First, note that the update rule given in Equation 1 requires knowledge of the equilibrium policies of the benign agents in every iteration, which in turn requires knowledge of the Q-values of all agents, from every agent’s point of view. This is a downside that all centralized, value-based, algorithms in MARL, such as Nash Q-Learning, share. Second, even if knowledge of the Q-values of all agents can be guaranteed, the problem of computing a Nash equilibrium from given utilities in a general-sum Markov game is known to be computationally hard [Daskalakis et al., 2009]. Finally, the theoretical guarantees of the proposed method heavily rely on the stated assumptions. Such assumptions may not always be satisfied in practice, where the irregularities in the individual utilities do not need to satisfy saddle point or global optima conditions. Motivated by the above, our next goal is thus to find a more practical and efficient approach to finding ARNEQ policies. In the next section, we introduce a model-free gradient-based algorithm that is able to empirically provide an efficient defense in various MARL environments.

### 4 Adversarially Robust Training for MARL

In this section, we propose an independent learning algorithm, Adversarially Robust Training (ART) for MARL, a gradient-based method that uses a two-timescale update rule, designed to provide a practically efficient defense against unknown adversarial policies.

Our robust notion of equilibrium imposes a specific training strategy. Note that the adversarial policies that complement an ARNEQ strategy profile are not necessarily identical across different players. Thus, each player, apart from computing its policy, also needs to compute its own set of adversarial agents and their associated adversarial policies. Thus, three different components need to be learned independently. Since each individual problem is a \( \max \min \) problem, it is natural to consider the Gradient Descent Ascent (GDA) learning paradigm [Lin et al., 2020]. This method has
been widely used in the context of Markov games and has been shown to converge to equilibria in more structured regimes [Daskalakis et al., 2020].

Recall from the previous section that $\omega_i, \pi_i$, and $\tilde{\pi}_i$ denote the adversarial subset selection policy, the benign policy, and adversarial joint policy with respect to agent $i$, respectively. We parametrize them as $\theta_i, \tilde{\theta}_i$ and $\tilde{\theta}_i$, respectively. Based on these definitions, we define the loss with respect to a given adversarial subset selection policy $\omega_i$, dependent on joint policy $\pi$ and adversarial policy $\tilde{\pi}_i$, as

$$L_i(\theta_i, \tilde{\theta}_i) = \mathbb{E}_{K \sim \omega_i(\theta_i)} \left( V^i_i \left( \pi_K(\tilde{\theta}_K), \tilde{\pi}_K(\tilde{\theta}_K) \right) \right).$$

Based on this definition, and the fact that we can write the above loss as an inner product, we have $\nabla_{\theta_i} L_i(\theta_i, \tilde{\theta}_i) = (\nabla_{\theta_i} \omega_i(\theta_i), V^i_i(\pi(\tilde{\theta}), \tilde{\pi}(\tilde{\theta})))$, where $V^i_i(\pi(\tilde{\theta}), \tilde{\pi}(\tilde{\theta}))$ denotes the $K \in \mathbb{N}$.

The loss, with respect to player $i$, of the adversarial policy $\tilde{\pi}_i$, depending on $\omega_i$ and $\pi$ is similarly defined. However, the gradient here is with respect to $\tilde{\theta}_i$. Note that, for each component $j$ of the adversarial policy $\tilde{\pi}_i$ the gradient $\nabla_{\tilde{\theta}_i} L_i(\theta_i, \tilde{\theta}_i)$ is proportional to $\sum_{K:K \in \mathbb{N}, j \in K} V^i_i(\pi_K(\tilde{\theta}_K), \tilde{\pi}_K(\tilde{\theta}_K))$, since for those subsets $K$ that do not contain $j$ as an adversarial component, the gradient becomes 0.

Our learning protocol proceeds as follows. In every round $t \geq 1$, player $i \in \mathbb{N}$ maintains three gradient updates, one for the adversarial subset parameters $\theta_i^t$, one for the adversarial policy parameters $\tilde{\theta}_i^t$ and one for its own benign policy parameters $\tilde{\theta}_i^t$. Furthermore, we set the same learning rate $\eta_A$ for both $\omega_i^t$ and $\tilde{\pi}_i^t$, since they both comprise the adversarial component of the problem.

Algorithm 1 ART for MARL (for player $i$)

**Input**: Number of potential adversaries $k$; accuracy parameter $\epsilon > 0$; learning rates $\eta_A$ and $\eta_B$; number of episodes $T$. **Initialize**: $\theta_i^0 = 0, \tilde{\theta}_i^0 = 0, \tilde{\theta}_j^0 = 0$, for all $i \in \mathbb{N}, j \neq i$.

1: for $t = 1, 2, \ldots, T$ do:
2: \hspace{1em} Sample adversarial subsets from $\omega_i^{t-1}$.
3: \hspace{1em} $\theta_i^t \leftarrow \theta_i^{t-1} - \eta_A \nabla_{\theta_i} L_i \left( \theta_i^{t-1}, \tilde{\theta}_i^{t-1}, \tilde{\theta}_j^{t-1} \right)$
4: \hspace{1em} $\tilde{\theta}_i^t \leftarrow \tilde{\theta}_i^{t-1} + \eta_B \nabla_{\tilde{\theta}_i} L_i \left( \theta_i^t, \tilde{\theta}_i^{t-1}, \tilde{\theta}_j^{t-1} \right)$
5: for $j \in \mathbb{N}$ do:
6: \hspace{1em} $\tilde{\theta}_j^t \leftarrow \tilde{\theta}_j^{t-1} - \eta_A \nabla_{\tilde{\theta}_j} L_i \left( \theta_i^t, \tilde{\theta}_i^{t-1}, \tilde{\theta}_j^{t-1} \right)$
7: \hspace{2em} end for
8: end for

At the beginning of the round, player $i$ collects samples of subsets from the adversarial subset selection policy of the previous round $\omega_i^{t-1}$. In order to perform the gradient updates of round $t$, player $i$ needs to collect several roll-outs. We denote by $\nabla L_i(\theta_i, \tilde{\theta}_i)$ the gradient estimate used for the updates based on the collected roll-outs. The pseudocode of the described method is given in Algorithm 1.

5 Experimental Results

We evaluate policies trained using ART in two cooperative environments and one environment where the agents do not have aligned utilities. More specifically, we consider the Spread multi-agent-particle environment [Lowe et al., 2017], a cooperative multi-agent extension of the MuJoCo Ant environment [Todorov et al., 2012], and the cooperative Pursuit environment Gupta et al. [2017]. For detailed descriptions and parameters of the environments, see Appendix D.

5.1 Baselines

We compare the effectiveness of ART with Naive Training, where no adversarial agents are present during the training phase. This provides a baseline for evaluating the resilience of robustly trained
where the subset of agents controlled by the adversary is fixed during the training process. We aim to only consider $k$ agents who have a common reward function. The table reports the total reward of these agents. Note that, for Ant, we report the worst-case total reward, where the minimum is taken over all possible adversaries to minimize the total reward of one of the benign agents, considered to be a victim.

For Independent Spread, we report the worst-case total reward, where the minimum is taken over all possible victims and their adversaries. Since Ant and Pursuit are cooperative and symmetric environments, we consider an attacker that controls a specific subset of agents with cardinality $k$, while the remaining agents are victims who have a common reward function. The table reports the total reward of these agents. Note that, for Ant, we only consider $k = 1$, since an adversary that controls more agents can lift the ant up, denying any forward movement.

Table 1: Comparison of ART with the proposed baselines. Fixed-K($k=2$) for Independent Spread is omitted, as it is equivalent to ART($k=2$). We report the mean and the standard error of the total reward achieved by an agent under attack over five runs with different seeds. In each run, we first train the benign agents and then train adversaries to minimize the total reward of one of the benign agents, considered to be a victim.

To train robust neural policies, we implemented ART as described in Algorithm 1. Both the adversarial and benign policies are updated using the PPO [Schulman et al., 2017] implementation of the ray library [Moritz et al., 2018]. The adversarial subset selection model $\omega$, as described in Section 4

<table>
<thead>
<tr>
<th>Environment</th>
<th>Algorithm</th>
<th>No Adversary</th>
<th>1 Adversary</th>
<th>2 Adversaries</th>
<th>4 Adversaries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent</td>
<td>Naive</td>
<td>$-34.0 \pm 1.9$</td>
<td>$-493.2 \pm 11.2$</td>
<td>$-480.8 \pm 16.1$</td>
<td>NA</td>
</tr>
<tr>
<td>Spread</td>
<td>Fixed-K($k=1$)</td>
<td>$-142.8 \pm 24.5$</td>
<td>$-271.2 \pm 31.0$</td>
<td>$-290.8 \pm 36.6$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ART($k=1$)</td>
<td>$-123.4 \pm 11.7$</td>
<td>$-190.2 \pm 13.3$</td>
<td>$-242.0 \pm 33.4$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ART($k=2$)</td>
<td>$-148.0 \pm 15.1$</td>
<td>$-201.4 \pm 11.7$</td>
<td>$-213.6 \pm 9.9$</td>
<td></td>
</tr>
<tr>
<td>Ant</td>
<td>Naive</td>
<td>$327.5 \pm 48.9$</td>
<td>$2.2 \pm 10.7$</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>Fixed-K($k=1$)</td>
<td>$229.0 \pm 30.0$</td>
<td>$29.5 \pm 14.8$</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>ART($k=1$)</td>
<td>$181.9 \pm 25.4$</td>
<td>$58.2 \pm 6.7$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naive</td>
<td>$40.7 \pm 31$</td>
<td>$34.4 \pm 1.2$</td>
<td>$32.3 \pm 1.9$</td>
<td>$-2.0 \pm 2.3$</td>
</tr>
<tr>
<td>Pursuit</td>
<td>Fixed-K($k=1$)</td>
<td>$37.6 \pm 2.8$</td>
<td>$35.7 \pm 4.1$</td>
<td>$30.9 \pm 2.8$</td>
<td>$8.5 \pm 2.7$</td>
</tr>
<tr>
<td></td>
<td>Fixed-K($k=2$)</td>
<td>$29.2 \pm 10.5$</td>
<td>$26.3 \pm 10.3$</td>
<td>$25.7 \pm 9.8$</td>
<td>$17.1 \pm 3.3$</td>
</tr>
<tr>
<td></td>
<td>Fixed-K($k=4$)</td>
<td>$25.89 \pm 7.24$</td>
<td>$23.82 \pm 7.50$</td>
<td>$16.97 \pm 8.12$</td>
<td>$6.21 \pm 6.27$</td>
</tr>
<tr>
<td></td>
<td>ART($k=1$)</td>
<td>$47.3 \pm 1.3$</td>
<td>$50.6 \pm 1.8$</td>
<td>$48.1 \pm 1.9$</td>
<td>$26.0 \pm 4.3$</td>
</tr>
<tr>
<td></td>
<td>ART($k=2$)</td>
<td>$42.6 \pm 3.7$</td>
<td>$37.4 \pm 1.7$</td>
<td>$36.4 \pm 2.4$</td>
<td>$21.0 \pm 4.0$</td>
</tr>
<tr>
<td></td>
<td>ART($k=4$)</td>
<td>$35.6 \pm 4.3$</td>
<td>$32.8 \pm 1.5$</td>
<td>$34.0 \pm 1.6$</td>
<td>$19.3 \pm 2.0$</td>
</tr>
</tbody>
</table>

Figure 2: Consecutive snapshots of agents under attack by one adversary in the Independent Spread environment: first row represents Naive Training and second row represents ART($k=1$). The green circle corresponds to the agent under attack, red to the adversary, blue to the benign agent not under attack, the black star is the target landmark of the attacked agent, and the black circles are the target landmarks of other agents. Naively trained agents do not avoid other agents and allow the adversary to infer the position of their landmark, resulting in adversarial agents, and maintain distance around their landmark, which does not allow the adversary to infer its exact location.

agents compared to conventional techniques. Additionally, we introduce the Fixed-K training baseline where the subset of agents controlled by the adversary is fixed during the training process. We aim to signify the importance of training with the worst-case subset of adversaries. Both baselines are trained with the same hyperparameters as ART, ensuring a fair comparison across the board.

5.2 Implementation Details

To train robust neural policies, we implemented ART as described in Algorithm 1. Both the adversarial and benign policies are updated using the PPO [Schulman et al., 2017] implementation of the ray library [Moritz et al., 2018]. The adversarial subset selection model $\omega$, as described in Section 4
Figure 3: Consecutive snapshots of agents under attack in the Ant environment: first row represents naive training and second row represents ART(k=1). The dotted red line in Ant serves as a stationary reference point. Naively trained agents struggle to move forward when under attack. Robustly trained ones are able to move forward when one leg is corrupted.

Figure 4: Consecutive snapshots of agents under attack by 4 adversaries in the Pursuit environment: the first row is a result of Naive Training and the second row is a result of ART(k=1). For better visibility, we zoomed in on the area of interest (indicated by the red dotted area). A full version can be found in the Appendix (see the Supplementary Material). Red circles correspond to adversarial agents, green to benign agents, and blue to evaders, the orange boxes to the field of view of agents. Yellow arrows show the movement of the adversary in the upper row and benign agents in the lower row. Naively trained agents expect cooperation, leading to agents being easily fooled. The adversary moves upward instead of left, resulting in the prey not being captured. Agents trained with ART learned to meet at a specific rendezvous point unknown to the adversary at the upper half of the environment, avoiding interacting with adversaries altogether.

is implemented using pytorch [Paszke et al., 2019] and utilizes REINFORCE [Sutton et al., 1999] update rules. To allow training $\omega$ with more data, we used additionally generated trajectories. Hyperparameters were selected according to the fine-tuned examples provided in the ray library, except for the learning rate of benign agents, which is set to be half of the learning rate of adversarial agents. This leads to aggressively updated adversaries, while still ensuring convergence in a reasonable time. All used hyperparameters can be found in the Appendix (see the Supplementary Material).

5.3 Empirical Analysis

Quantitative analysis. To evaluate the robustness of the obtained agents, we fixed the benign policies and trained new adversaries from scratch that use the same hyperparameters. We used the latter to test if the robust policies generalize to new attackers. Table 1 reports the test-time results of ART and both baselines. We observe that in all environments, ART consistently outperforms Naive Training when under attack. Fixed-K training results in a more robust policy than naive training, but the benign agents tend to overfit to the adversaries present during training, resulting in worse performance when under attack by different adversarial agents. The importance of anticipating the correct number of adversaries during the training procedure is environment-specific. In Independent Spread, optimal performance of ART is achieved if the number of adversaries during training matches their number at deployment time. In pursuit, training with a smaller number of adversaries yields better performance in all scenarios. We conjecture that this may be due to the hardness of training with a larger number of adversarial agents. In the Independent Spread and Ant environment the resilience against adversarial attacks of ART and Fixed-K comes at the cost of decreased performance in a clean environment. In
Figure 5: Achievable increase in the performance of a single agent, i.e., victim, after we train the benign agents and adversaries as described in Table 1. We consider one fixed victim agent and 5 different runs with different seeds. In each run, we train the benign policies until convergence using ART. Afterwards, we select one agent and train an adversary attacking the selected agent. Then, training of the selected agent is continued for 5% of the original number of training steps, while keeping all other policies fixed. Depicted is the difference between the performance of the improved policy and the original performance. If the players have converged to an ARNEQ, then further training a given agent when the rest of the players keep playing fixed strategies should not imply increased performance for this player. Otherwise, this would mean there is room for improvement which would violate the very condition of an equilibrium. Note that, on average, performance tends to stay the same, except for some small oscillations which are due to the stochasticity of deep RL methods. This suggests that players are close to an ARNEQ.

Pursuit, adversarial training is beneficial even when there is no attack. This can, in part, be explained by examining the behavior of the trained agents, which we discuss in the qualitative analysis.

**Qualitative analysis.** Next, we analyze how the behavior of agents trained with ART differs from naive training in the examined environments. For Pursuit, Figure 4 depicts that agents trained using ART meet up at a specific point at the beginning of the episode. This strategic rendezvous not only allows them to navigate the environment in a trusted group, enabling them to bypass interactions with adversaries but seems to contribute to improved performance even when not under attack. For Independent Spread, Figure 2 shows that agents trained using ART opt for longer routes, strategically avoiding potential adversaries. This behavior explains the observed difference in performance when comparing ART and naive training in clean and adversarial environments. For Ant, Figure 3 depicts the ability of ART to train agents that are able to move the ant forward even if one leg is acting maliciously. As can be seen in the figure, we do not learn such a robust policy with Naive Training.

**Convergence.** Finally, we aim to evaluate the convergence of ART by analyzing the achievable increase in performance of a single agent. A small increase signifies the approximate convergence of ART to an ARNEQ. Figure 5 depicts this increase in performance for all evaluated environments and differing numbers of adversaries.

6 Conclusion

We considered the problem of defending against unknown adversarial policies in general-sum Markov games. We provided a theoretical characterization of our solution concept and further proposed a practical algorithm to approach it. Finally, we empirically demonstrated its efficiency. A key open question that is left in the paper is whether we can restrict the attack model so that we are able to provide theoretical convergence guarantees on independent gradient-based learning methods in more structure regimes, such as Markov potential games. Moreover, it would be interesting to see how scalable our method is for large games. Here, approaches such as mean-field multi-agent reinforcement learning might provide essential tools.
Acknowledgements

This research was, in part, funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 467367360.

References


Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, Multi-agent, Reinforcement Learning for Autonomous Driving. *CoRR*, abs/1610.03295, 2016.


Fan Wu, Linyi Li, Zijian Huang, Yevgeniy Vorobeychik, Ding Zhao, and Bo Li. CROP: Certifying Robust Policies for Reinforcement Learning through Functional Smoothing. CoRR, abs/2106.09292, 2021.

Fan Wu, Linyi Li, Chejian Xu, Huan Zhang, Bhavya Kailkhura, Krishnaram Kenthapadi, Ding Zhao, and Bo Li. COPA: Certifying Robust Policies for Offline Reinforcement Learning against Poisoning Attacks. CoRR, abs/2203.08398, 2022.


Appendix

Table of Contents

<table>
<thead>
<tr>
<th>A</th>
<th>Other Related Work</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1</td>
<td>Attacks and defenses in (MA)RL.</td>
<td>15</td>
</tr>
<tr>
<td>A.2</td>
<td>Robust (MA)RL.</td>
<td>16</td>
</tr>
<tr>
<td>A.3</td>
<td>Markov games.</td>
<td>16</td>
</tr>
<tr>
<td>B</td>
<td>Proof of Existence (Theorem 1)</td>
<td>16</td>
</tr>
<tr>
<td>B.1</td>
<td>Auxiliary Results</td>
<td>16</td>
</tr>
<tr>
<td>B.2</td>
<td>The Proof of the Theorem</td>
<td>20</td>
</tr>
<tr>
<td>C</td>
<td>Proof of Convergence of Adversarially Robust Nash Q-Learning (Theorem 2)</td>
<td>21</td>
</tr>
<tr>
<td>D</td>
<td>Experiments: Environments</td>
<td>24</td>
</tr>
<tr>
<td>D.1</td>
<td>Independent Spread</td>
<td>24</td>
</tr>
<tr>
<td>D.2</td>
<td>Multi-Agent Ant</td>
<td>24</td>
</tr>
<tr>
<td>D.3</td>
<td>Pursuit</td>
<td>24</td>
</tr>
<tr>
<td>E</td>
<td>Experiments: Hyperparameters</td>
<td>25</td>
</tr>
<tr>
<td>F</td>
<td>Experiments: Computing Infrastructure</td>
<td>25</td>
</tr>
<tr>
<td>G</td>
<td>Experiments: Runtime</td>
<td>25</td>
</tr>
</tbody>
</table>

A  Other Related Work

Our work is also related to the following strands of research.

A.1 Attacks and defenses in (MA)RL.

Adversarial attacks on RL systems have been extensively studied in recent years Kiourti et al. [2020], Lin et al. [2017], Huang et al. [2017], Mohammadi et al. [2023], Sun et al. [2020b,a]. The main types of these attacks include training-time attacks Rakhsha et al. [2020], Xu et al. [2021], test-time attacks Behzadan and Munir [2017], Huang et al. [2017], Kos and Song [2017], Sun et al. [2020a], and backdoor attacks Kiourti et al. [2020], Wang et al. [2021], Yang et al. [2019], where the adversarial perturbations happen at the environment level. More recently, however, there has been a growing interest in the so-called adversarial policies in (MA)RL Gleave et al. [2020], Guo et al. [2021], Liu et al. [2022], Wang et al. [2023], Mohammadi et al. [2023], Li et al. [2019] where attackers assume the identity of an agent, and thus attack by inducing natural observations. Our work belongs to this class of attack type, but generalizes the framework of adversarial policies in MARL for general $n$-player games and focuses on the defense front. Various types of defenses have been proposed against test-time attacks Zhang et al. [2020], Pattanaik et al. [2017], Zhang et al. [2021a], Wu et al. [2021], training-time attacks Banihashem et al. [2023], Kumar et al. [2021], Lykouris et al. [2021], Wu et al. [2022], Nika et al. [2023], Zhang et al. [2021a, 2022] and backdoor attacks Bharti et al. [2022]. When it comes to adversarial policies, previous work has mainly focused on heuristic defenses against them. For two-player zero-sum Markov games, Gleave et al. [2020] propose sequential fine-tuning procedures against various adversaries, which seems to robustify the trained policies against such adversaries, but not against a more general attack. Furthermore, apart from issues such as catastrophic forgetting and myopic robustness against a specific class of adversaries, the aforementioned defense strategy becomes combinatorially expensive in $n$-player Markov games. If the identities of the controlled agents and the objectives of the attacker were known, then a natural solution concept
would be the Nash equilibrium (NE) strategy of the new game. In our setting, this information is not
given. As a consequence, the problem is no longer a standard Markov game, and thus there is no
NE defined in its solution space. Recently Liu et al. [2023] consider the problem of defense in the
two-player regime with one adversarial policy and take a game-theoretic approach. They define a new
game with modified utilities and then solve the game as if it were a zero-sum game. We generalize
their setting and study the defense problem against any up to $k$ out of $n$ adversarial policies in MARL.

As we mentioned earlier, this formulation renders the classical NE notion inappropriate. Thus, a new
solution concept is necessary.

A.2 Robust (MA)RL.

Another line of research related to our work is that of robustness in (MA)RL under model uncertainty
Wang and Zou [2021], Zhang et al. [2020], Russel et al. [2020], Mankowitz et al. [2018]. In this line
of work, the focus is on designing algorithms that learn under worst-case environment assumptions.
Instead, we consider the robustness against the worst-case subset of adversarial policies at test time.
A recent work from Li et al. [2023] studies a similar problem to ours. They consider defenses against
unknown Byzantine attacks in a cooperative regime. However, they take a different approach to the
defense problem. They capture the uncertainty about the adversaries as uncertainty in transitions,
which are characterized by types, thus taking a Bayesian game approach, while allowing only for
one adversary. We take a direct robust approach on each individual agent while allowing for a more
general attack of up to $k$ adversaries. Furthermore, their notion of robust equilibrium pertains to only
one joint adversarial policy, while our robust equilibrium associates each robust policy with a specific
joint adversarial strategy. This substantially changes the setting since we are, in effect, solving $n$
games simultaneously.

A.3 Markov games.

We formalize our study using the notion of Markov games [Shapley, 1953]. Our goal is to design
decentralized robust training procedures. There has been a lot of research in the past decade on
model-based centralized Hu and Wellman [2003] and model-free decentralized methods of computing
the celebrated Nash equilibrium of such games [Daskalakis et al., 2020, Leonardos et al., 2021, Ding
et al., 2022, Zeng et al., 2022, Giannou et al., 2022, Zhang et al., 2021b, Wang and Zou, 2022, Wei
et al., 2021]. However, all efficient methods are proposed for particular types of Markov games,
such as zero-sum or potential Markov games. The notion of robust Markov games has also been
previously considered. For instance, Aghassi and Bertsimas [2006] propose the notion of ex-post
equilibrium to handle payoff uncertainty in normal form games, Kardeš et al. [2011] consider robust
Markov games with model uncertainty. In contrast, our notion of uncertainty is related to adversarial
policies. Finally, Kalogiannis et al. [2023] consider cooperative Markov games with a fixed adversary.
Our work differs from them in that we consider a general-sum Markov game setting and assume the
presence of unknown adversarial policies. This substantially changes the nature of the game since the
agents cannot cooperatively team up against the adversaries.

B Proof of Existence (Theorem 1)

For the existence proof, we will use the Kakutani’s fixed point Theorem. This result has been
classically used in proving the existence of Nash equilibria in various types of Markov games. It
requires the careful construction of a set-valued function whose fixed point would represent the
equilibrium of the game of interest. Once the construction is made, the theorem states that, if such
a function satisfies some technical conditions, then the existence of its fixed point is guaranteed,
thus effectively proving the existence of an equilibrium of the game. This will be our approach in
the following. First, we will prove some auxiliary results related to properties such as contraction,
continuity, and convexity. Then, we will construct a set-valued function whose fixed point would
represent an ARNEQ and further show that it satisfies the technical conditions of Kakutani’s fixed
point theorem.

B.1 Auxiliary Results

In this section we prove some auxiliary results which will be needed in the proof of Theorem 1.
We will use the following definitions only for this section. Fix an arbitrary state $s$ and player $i$. Let $x^i_s$ denote a benign policy $\pi_i(\cdot|s)$ of player $i$ at state $s$, and let $y^{-i}_s$ denote the joint policy $\pi_{-i}(\cdot|s)$ of the rest of the benign players. Moreover, let $z^i_s$ denote an adversarial joint policy $\pi_i(\cdot|s)$ and $\omega^i_s$ denote the randomized strategy $\omega_i$, but only applied on state $s$. We also denote by $\mathcal{X}$ the policy space $\Delta(A_i)$ and by $\mathcal{Y}$ the joint policy space $\times_{j \neq i} \Delta(A_j)$. Note that we do not assume action spaces that depend on the state. Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ denote the whole policy space $\Pi$.

Now, given a vector of value functions $(V^i)_{i \in N}$ in some compact value function space $\mathcal{V} = \mathcal{V}_1 \times \ldots \times \mathcal{V}_n$, let us define, for each player $i \in N$ and state $s \in S$, the function

$$f^i_s(x^i_s, y^{-i}_s, z^i_s, \omega^i_s, V^i) := \mathbb{E}_{a \sim (x^i_s, y^{-i}_s, z^i_s), \omega^i_s} \left[ R(s, a) + \gamma \langle P(s, a), V^i \rangle \right].$$

Next, let us define

$$\phi^i_s(x^i_s, y^{-i}_s, V^i) := \min_{z^i_s \in \mathcal{Y}} \min_{\omega^i_s \in \Omega_i} f^i_s(x^i_s, y^{-i}_s, z^i_s, \omega^i_s, V^i),$$

and

$$\psi^i_s(y^{-i}_s, V^i) := \max_{x^i_s \in \mathcal{X}} \phi^i_s(x^i_s, y^{-i}_s, V^i).$$

Note that, the spaces $\mathcal{X}$, $\mathcal{Y}$, and $\Omega_i$ are compact spaces (closed and bounded) as policy spaces. Our first result shows that the mapping $\psi$ is a contraction mapping on the value function spaces.

**Lemma 1** Given joint policy $x$, let $\psi_x : \mathcal{V} \to \mathcal{V}$ be such that, for each $i, s$ we have $\psi_x(V) = (\psi^i_s(y^{-i}_s, V^i))_{i, s}$. Then, the mapping $\psi$ is a contraction mapping.

**Proof:** Let $V, U \in \mathcal{V}$. Fix a player $i$ and state $s$. Note that we have

$$|\psi^i_s(y^{-i}_s, V^i) - \psi^i_s(y^{-i}_s, U^i)| = \left| \max_{x^i_s \in \mathcal{X}} \phi^i_s(x^i_s, y^{-i}_s, V^i) - \max_{x^i_s \in \mathcal{X}} \phi^i_s(x^i_s, y^{-i}_s, U^i) \right|$$

$$\leq \left| \phi^i_s(x^i_s, y^{-i}_s, V^i) - \phi^i_s(x^i_s, y^{-i}_s, U^i) \right|$$

$$\leq \min_{z^i_s \in \mathcal{Y}} \min_{\omega^i_s \in \Omega_i} f^i_s(x^i_s, y^{-i}_s, z^i_s, \omega^i_s, V^i) - \min_{z^i_s \in \mathcal{Y}} \min_{\omega^i_s \in \Omega_i} f^i_s(x^i_s, y^{-i}_s, z^i_s, \omega^i_s, U^i)$$

$$\leq f^i_s(x^i_s, y^{-i}_s, z^i_s, \omega^i_s, V^i) - f^i_s(x^i_s, y^{-i}_s, z^i_s, \omega^i_s, U^i)$$

$$\leq \mathbb{E}_{a \sim (x^i_s, y^{-i}_s, z^i_s), \omega^i_s} \left[ R(s, a) + \gamma \langle P(s, a), V^i \rangle - (R(s, a) + \gamma \langle P(s, a), U^i \rangle) \right]$$

$$\leq \mathbb{E}_{a \sim (x^i_s, y^{-i}_s, z^i_s), \omega^i_s} \left[ \gamma \langle P(s, a), V^i - U^i \rangle \right]$$

$$\leq \|V^i - U^i\|_{\infty},$$

where for the first inequality, we let $\bar{x}^i_s$ be the maximizer of $\phi^i_s(x^i_s, y^{-i}_s, V^i)$, for the second inequality we let $(\bar{z}^i_s, \bar{\omega}^i_s)$ be the minimizer of $f^i_s(x^i_s, y^{-i}_s, z^i_s, \omega^i_s, U^i)$, and for the last inequality we use Hölder. \(\square\)

Next, we will show that there exists a unique robust value vector of the above mapping, a result that follows from Banach’s contraction theorem which we state below.

**Theorem 3 (Banach’s Contraction Mapping Theorem)** Let $(\mathcal{V}, \| \cdot \|)$ be a complete metric space and let $\psi : \mathcal{V} \to \mathcal{V}$ be a contraction mapping. Then there exists a unique fixed point of the function $\psi$.

**Lemma 2** For any given joint policy $x$, player $i \in N$ and state $s \in S$, there exists a unique value vector $V^i_s$ such that

$$V^i_s = \max_{x^i_s \in \mathcal{X}} \min_{z^i_s \in \mathcal{Y}} \min_{\omega^i_s \in \Omega_i} f^i_s(x^i_s, y^{-i}_s, z^i_s, \omega^i_s, V^i) = \psi^i_s(y^{-i}_s, V^i).$$

**Proof:** Note that, the fact that the metric space $(\mathcal{V}, \| \cdot \|_{\infty})$ is complete, since $\mathcal{V}$ is compact, together with Lemma 1 above, implies the desired result as a consequence of Banach’s contraction mapping point theorem (Theorem 3). \(\square\)
where the second inequality follows from Hölder’s inequality, where \( x_s \) denotes the joint policy vector for state \( s \), \( x^i_{sa} \) denotes the probability of player \( i \) taking action \( a \in A_i \) in state \( s \), while \( V^i_s \) denotes the value of state \( s \) for player \( i \) with respect to value vector \( V \).

**Lemma 3** Fix \( s \in S \). Given \( \epsilon > 0 \), there exists \( \delta(\epsilon) > 0 \), such that, if for any \( p = (x_s, V) \) and \( q = (y_s, U) \), we have \( d(p, q) < \delta(\epsilon) \), then, for all \( z^i_s \in Y \) and \( \omega_s \in \Omega \), we have

\[
|f^i_s(x_s, z^i_s, \omega_s, V^i) - f^i_s(y_s, z^i_s, \omega_s, U^i)| < \epsilon.
\]

**Proof:** First, note that, since rewards are in the unit interval, value functions are bounded by \( 1/(1 - \gamma) \). Letting \( p = (x_s, V) \) and \( q = (y_s, U) \), we have

\[
|f^i_s(x_s, z^i_s, \omega_s, V^i) - f^i_s(y_s, z^i_s, \omega_s, U^i)|
= |E_{a \sim (x_s, z^i_s, \omega_s)} [R(s, a) + \gamma \langle P(s, a), V^i \rangle] - E_{a \sim (y_s, z^i_s, \omega_s)} [R(s, a) + \gamma \langle P(s, a), U^i \rangle]|
\]

\[
\leq E_{\omega \sim Y} \left[ \sum_{a \in A} \prod_{j \in K^*} z^i_{sa_j} \prod_{l \in K^*} x^i_{la_j} (R(s, a) + \gamma \langle P(s, a), V^i \rangle) - \sum_{a \in A} \prod_{j \in K^*} z^i_{sa_j} \prod_{l \in K^*} y^i_{la_j} (R(s, a) + \gamma \langle P(s, a), U^i \rangle) \right]
\]

\[
\leq \sum_{a \in A} \left| \prod_{j \in K^*} z^i_{sa_j} \left( \prod_{l \in K^*} x^i_{la_j} - \prod_{l \in K^*} y^i_{la_j} \right) R(s, a) \right|
+ \gamma \sum_{a \in A} \left| \prod_{j \in K^*} z^i_{sa_j} \left( \prod_{l \in K^*} x^i_{la_j} - \prod_{l \in K^*} y^i_{la_j} \right) \langle P(s, a), V^i \rangle \right|
+ \gamma \sum_{a \in A} \left| \prod_{j \in K^*} z^i_{sa_j} \prod_{l \in K^*} y^i_{la_j} \langle P(s, a), V^i - U^i \rangle \right|
\]

\[
\leq \frac{1}{1 - \gamma} \sum_{a \in A} \left| \prod_{j \in K^*} z^i_{sa_j} \left( \prod_{l \in K^*} x^i_{la_j} - \prod_{l \in K^*} y^i_{la_j} \right) + \frac{\gamma}{1 - \gamma} \sum_{a \in A} \prod_{j \in K^*} z^i_{sa_j} \left( \prod_{l \in K^*} x^i_{la_j} - \prod_{l \in K^*} y^i_{la_j} \right) \right|
+ \gamma A^n \|V^i - U^i\|_\infty,
\]

where the second inequality follows from Hölder’s inequality, where \( K^* \) denotes the set that maximizes the difference inside the absolute value; for the third inequality we have just used an algebraic artifice, adding and subtracting a term related to \( y^i \) and \( V^i \), and then rearranging; the third inequality uses the fact that the rewards are in the unit interval, and Hölder’s inequality.
Now, let us define
\[
\delta_1(\epsilon) = \frac{(1 - \gamma) \min\{\epsilon, 1\}}{2(2^{n-k} - 1)A^n}, \quad \text{and} \quad \delta_2(\epsilon) = \frac{\min\{\epsilon, 1\}}{2\gamma A^n}.
\]

Moreover, let \(\alpha_{s,a}^i = y_{sa}^i - x_{sa}^i\). Assume that \(|\alpha_{s,a}^i| < \min\{\delta_1(\epsilon), \delta_2(\epsilon)\}\), and \(|V^i_s - U^i_s| < \min\{\delta_1(\epsilon), \delta_2(\epsilon)\}\) for any player \(i \in \mathcal{N}\), state \(s \in S\) and action \(a \in A\). For the first term on the right-hand-side of the last equality above, we have
\[
\frac{1}{1 - \gamma} \sum_{a \in A} \prod_{j \in K^*} z_{sa}^{i,j} \left( \prod_{j \notin K^*} x_{sa}^i - \prod_{j \notin K^*} y_{sa}^i \right) \leq \frac{1}{1 - \gamma} \sum_{a \in A} \prod_{j \in K^*} z_{sa}^{i,j} \sum_{M \subseteq \mathcal{N} \backslash K^*} \left| \prod_{m \in M} \alpha_{s,a}^m \right| \prod_{m \notin M} y_{sa}^m \leq \frac{1}{1 - \gamma} \sum_{a \in A} \sum_{M \subseteq \mathcal{N} \backslash K^*} \left| \alpha_{s,a}^m \right| \leq \frac{\epsilon}{2},
\]
where the first inequality follows from the algebraic identity
\[
\left| \prod_{m=1}^{n-k} (y_{sa}^m + \alpha_{s,a}^m) - \prod_{m=1}^{n-k} y_{sa}^m \right| = \sum_{M \subseteq \mathcal{N} \backslash K^*} \left( \prod_{m \in M} \alpha_{s,a}^m \right) \left( \prod_{m \notin M} y_{sa}^m \right),
\]
the second inequality follows from the fact that the product of numbers in the unit interval is no greater than 1; the third inequality uses a similar argument on the policies \(z_{sa}^{i,j}\), and the rest of the differences \(\alpha_{s,a}^m\), except \(\alpha_{s,a}^m\), where we note that \(\alpha_{s,a}^m\) are also in the unit interval; the last inequality follows by definition of \(\delta_1(\epsilon)\).

On the other hand, note that we also have
\[
\gamma A^n \|V^i - U^i\|_\infty \leq \frac{\epsilon}{2},
\]
by definition of \(\delta_2(\epsilon)\). Thus, we obtain that
\[
|f^i_s(x_s, z_s^i, \omega_s^i, V^i) - f^i_s(y_s, z_s^i, \omega_s^i, U^i)| < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.
\]

\(\square\)

Next, we will state the following lemmas, which will be the rest of the necessary ingredients for the existence proof. We omit their proofs, since they follow the same lines as in [Kardeş et al., 2011, Fink, 1964]. First, let us define the unique best response for player \(i\) as
\[
\tau_i(y^{-i}) = \left\{ V^i : V^i_s = \max \min f^i_s(x^i_s, y_{s}^{-i}, z_s^i, \omega_s^i, V^i), \forall s \in S \right\}.
\]

**Lemma 4** The function \(\phi^i_s(x_s, y_{s}^{-i}, V^i)\) is continuous in all its variables, for all \(i \in \mathcal{N}\) and \(s \in S\).

**Proof:** The result immediately follows from the fact that the pointwise minimum of a family of equicontinuous functions is continuous. \(\square\)

**Lemma 5** The function \(\phi^i_s(x_s, y_{s}^{-i}, V^i)\) is concave in \(x_s^i\), for a fixed \(y_{s}^{-i}\) and \(V^i\).
With this, we can now state Kakutani’s fixed point theorem. We will construct a correspondence that satisfies the conditions of the theorem, and show that its variables. This, together with the fact that its domain is compact, implies that \( \phi_i \) achieves its maximum, that is \( \arg \max \phi_i(x^i, x_s^{-i}, V^i) \), for all \( s \in S \) and \( i \in N \).

Theorem 1 for convenience, and then proceed to its proof.

**Proof:** The result immediately follows by definition of \( \phi_i \).

**Lemma 6** The function \( \psi_i(y_s^{-i}, V^i) \) is continuous in \( y_s^{-i} \). Furthermore, the set \{ \( \psi_i(y_s^{-i}, V^i) | V^i \) is bounded \} is equicontinuous.

**Proof:** The result follows from Lemma 3 and Lemma 3 of Fink [1964].

**Lemma 7** If the sequence \( y^{-i,n} \) goes to \( y^{-i} \) and \( \tau_s(y^{-i,n}) \) goes to \( V_s \), as \( n \) goes to \( \infty \), then

\[
\tau_s(y^{-i}) = V_s.
\]

**Proof:** Observe that

\[
|V_s - \tau_s(y_s^{-i,n}, V^i)| \leq |V_s - \tau_s(y_s^{-i,n})| + |\tau_s(y_s^{-i,n}) - \psi_i(y_s^{-i}, \tau_s(y_s^{-i,n}))| + |\psi_i(y_s^{-i}, \tau_s(y_s^{-i,n}) - \psi_i(y_s^{-i,n}, V^i)|,
\]

for every \( n \geq 1 \), \( i \in N \) and \( s \in S \). Now note that \( |V_s - \tau_s(y_s^{-i,n})| \to 0 \) and \( \psi_i(y_s^{-i}, \tau_s(y_s^{-i,n}) - \psi(y_s^{-i,n}, V^i)) \to 0 \), as \( n \to \infty \), by assumption. Moreover, Lemma 6 implies that \( \tau_s(y_s^{-i,n}) - \psi_i(y_s^{-i}, \tau_s(y_s^{-i,n}) - \psi_i(y_s^{-i,n}, V^i)) \to 0 \), as \( n \to \infty \). Hence, \( |V_s - \tau_s(y_s^{-i,n})| \to 0 \), as \( n \to \infty \).

**B.2 The Proof of the Theorem**

In this section, we will conclude the proof of Theorem 1. In order to do that, we will make use of the famous Kakutani’s fixed point theorem which we state below. First, let us define the notion of upper semi-continuous functions, which is a precondition of this result.

**Definition 4** A correspondence \( \kappa : \mathcal{Z} \to 2^\mathbb{Z} \) is said to be upper semicontinuous if \( y_n \in \kappa(x_n) \), for all \( n \geq 1 \), \( \lim_{n \to \infty} x_n = x \) and \( \lim_{n \to \infty} y_n = y \) imply that \( y \in \kappa(x) \).

With this, we can now state Kakutani’s fixed point theorem.

**Theorem 4** (Kakutani’s fixed point theorem) If \( \mathcal{Z} \) is a closed, bounded and convex set in a Euclidean space, and \( \kappa \) is an upper semicontinuous correspondence mapping \( \mathcal{Z} \) into the family of closed convex subsets of \( \mathcal{Z} \), then there exists \( x \in \mathcal{Z} \) such that \( x = \kappa(x) \).

We will construct a correspondence that satisfies the conditions of the theorem, and show that its fixed point is an equilibrium point. The correspondence we need is the following:

\[
\kappa(x) = \big\{ y \in \mathcal{Z} : y_s^i \in \arg \max_{u_s^i \in \mathcal{X}} \phi_i^s(u_s^i, x_s^{-i}, V^i), \ V_s^i = \max_{u_s^i \in \mathcal{X}} \phi_i^s(u_s^i, x_s^{-i}, V^i), \forall s \in S, i \in N \big\}
\]

In the previous section, we have shown that the functions \( \phi_i \) satisfy continuity. We will further show that the defined set function \( \kappa \) satisfies the conditions of Kakutani’s theorem. Let us first restate Theorem 1 for convenience, and then proceed to its proof.

**Statement 1** Let \( \mathcal{G} \) be a finite game with \( n \) players and let \( k < n \). Assume that an arbitrary subset of at most \( k \) players use arbitrarily modified utilities, and assume that the mixed strategies of all players lie in compact sets. Also, assume that, for each player \( i \in N \), the sets \( \Omega_i \) are compact. Then, a \( k \)-ARNEG, as given in Definition 2 exists.

**Proof:** We will use the same argument as in the proof of Theorem 4 of [Kardeš et al., 2011]. We repeat the argument here with our notation for completion.

Given \( i \in N \), \( s \in S \), Lemma 4 shows that the function \( \phi_i^s(x_s^i, y_s^{-i}, V^i) \) is continuous in all of its variables. This, together with the fact that its domain is compact, implies that \( \phi_i^s \) achieves its maximum, that is \( \arg \max_{u_s^i \in \mathcal{X}} \phi_i^s(u_s^i, x_s^{-i}, V^i) \neq \emptyset \). On the other hand, Theorem 2 implies that \( V_s^i = \max_{x_s^i \in \mathcal{X}} \phi_i^s(x_s^i, y_s^{-i}, V^i) \). Thus, we have that \( \kappa(x) \neq \emptyset \), by definition of \( \kappa \) above.
Next, we show that $\kappa(x)$ is a convex set. Suppose that $u, v \in \kappa(x)$, for some $u = (u^1, \ldots, u^n)$ and $v = (v^1, \ldots, v^n)$. Then, for any $y \in S$ and $s \in S$, we have, by definition of $\kappa(x)$, that

$$V^i_s = \phi^i_s(u^i_s, x^{-i}_s, V^i) = \phi^i_s(v^i_s, x^{-i}_s, V^i) \geq \phi^i_s(y^i_s, x^{-i}_s, V^i),$$

for any $y \in S$. Thus, for any $\lambda \in [0, 1], i \in N, s \in S$, and by concavity of $\phi^i_s(y^i_s, x^{-i}_s, V^i)$, we have

$$\phi^i_s(y^i_s, x^{-i}_s, V^i) \leq V^i_s = \lambda \phi^i_s(u^i_s, x^{-i}_s, V^i) + (1 - \lambda) \phi^i_s(v^i_s, x^{-i}_s, V^i) \leq \phi^i_s((\lambda u^i_s + (1 - \lambda) v^i_s), x^{-i}_s, V^i) \leq V^i_s,$$

which implies that $\lambda u^i_s + (1 - \lambda) v^i_s \in \kappa(x)$. Thus, the image of $\kappa$ is convex.

Next, we will show that $\kappa$ is upper semicontinuous. To that end, suppose that $x^n \to x$ and $y^n \to y$, and that $y^n \in \kappa(x^n)$, for $n \geq 1$. Note that $\tau^i_s(x^{-i}, n) \to V^i_s$, since $x^{-i,n} \to x^{-i}$, as a subsequence of $(x^n)_{n\geq 1}$. Using the triangle inequality, we obtain

$$\left| \phi^i_s(y^i_s, x^{-i}_s, V^i) - V^i \right| \leq \left| \phi^i_s(y^i_s, x^{-i}_s, V^i) - \phi^i_s(y^{i,n}_s, x^{-i}_s, \tau^i_s(x^{-i,n})) \right| + \left| \phi^i_s(y^{i,n}_s, x^{-i}_s, \tau^i_s(x^{-i,n})) - V^i \right|$$

$$= \left| \phi^i_s(y^i_s, x^{-i}_s, V^i) - \phi^i_s(y^{i,n}_s, x^{-i}_s, \tau^i_s(x^{-i,n})) \right| + \left| \tau^i_s(x^{-i,n}) - V^i \right| \to 0, \text{ as } n \to \infty$$

Thus, $V^i_s = \phi^i_s(y^i_s, x^{-i}_s, V^i)$. Lemma 7 also implies that $\tau^i_s(x^{-i}) = V^i_s$. Thus, we obtain

$$V^i_s = \phi^i_s(y^i_s, x^{-i}_s, V^i) = \tau^i_s(x^{-i}) = \max_{u^i_s \in X^i_s} \phi^i_s(u^i_s, x^{-i}_s, V^i).$$

Therefore, $y \in \kappa(x)$, which is what we needed, in order to prove that $\kappa$ is upper semicontinuous. The fact that $\kappa(x)$ is a closed set for any $x \in X$ follows by definition of upper semicontinuity. Thus, $\kappa$ satisfies the conditions of Theorem 4, which implies that its fixed point exists, that is, there exists joint policies $x = (x^1, \ldots, x^n)$ and value functions $V^1, \ldots, V^n$, such that, for any $i \in N$ and $s \in S$, we have

$$x^i_s \in \arg \max_{u^i_s \in X^i_s} \min_{\omega^i_s \in \Omega^i_s, z^i_s \in Y^i_s} \mathbb{E}_{\alpha \sim x^i_s, x^{-i}_s, z^i_s, \omega^i_s} [R(s, a) + \gamma \langle P(s, a), V^i \rangle],$$

and

$$V^i_s = \max_{u^i_s \in X^i_s} \min_{\omega^i_s \in \Omega^i_s, z^i_s \in Y^i_s} \mathbb{E}_{\alpha \sim x^i_s, x^{-i}_s, z^i_s, \omega^i_s} [R(s, a) + \gamma \langle P(s, a), V^i \rangle].$$

Thus, an ARNEQ of $G$ exists. 

\[ \square \]

C Proof of Convergence of Adversarially Robust Nash Q-Learning (Theorem 2)

In this section, we will formally prove that the Nash Q-Learning approach to Adversarially Robust Training convergence to an ARNEQ under certain technical assumptions. We first define some necessary additional notions and then state the assumptions.

Given $t \geq 1$, let $(Q^i_t(s))_{i \in N}$ denote the stage game at state $s$, where $Q^i_t(s) = [Q^i_t(s, a)]_{a \in A}$ is the game matrix of player $i$ comprised of the $Q$-value estimates at time $t$. Recall that a stage game ARNEQ $\pi^i(\cdot | s), \pi^{j \downarrow} (\cdot | s), \omega^i)_{i \in N}$ for state $s$ is defined as the joint policy that satisfies, for each $i$, the following:

$$\pi^i(\cdot | s), \pi^{j \downarrow} (\cdot | s), \omega^i) \in \arg \max_{\pi^i \in \Pi^i} \min_{\pi^j \in \Pi^j} \mathbb{E}_{K \sim \omega^i} \left[ \sum_{a \in A} \pi^i(a^i | s) \prod_{i \neq j \in K} \pi^j(a^j | s) \prod_{l \in K} \pi^l(a^l | s) \mathbb{Q}^i_t(s, a) \right].$$

We now formally state Assumption 3.

**Statement 2** For each stage game at time $t$ and state $s \in S$, one of the following conditions holds.
A stage ARNEQ is also a global optimum, that is, for any \( i \in N \) and \( \pi(\cdot|s) \), we have
\[
E_{K \sim \omega_i^t} \left[ \sum_{a \in A} \pi_T^t(a_{-K}|s) \bar{\pi}_K^t(a_K|s) \bar{Q}_i^t(s, a) \right] \geq E_{K \sim \omega_i^t} \left[ \sum_{a \in A} \pi_T^t(a_{-K}|s) \bar{\pi}_K^t(a_K|s) \bar{Q}_i^t(s, a) \right],
\]
for any \( \pi \in \Pi \), \( \pi^t \in \Pi_{-i} \) and \( \omega_i \in \Omega_i \).

A given player's payoff is increased if other benign players or its attacker deviate, that is, for any \( i \in N \) we have
\[
E_{K \sim \omega_i^t} \left[ \sum_{a \in A} \pi_T^t(a_{-K}|s) \bar{\pi}_K^t(a_K|s) \bar{Q}_i^t(s, a) \right] \leq E_{K \sim \omega_i^t} \left[ \sum_{a \in A} \pi_T^t(a_{-K}|s) \pi_T^t(a_K|s) \bar{Q}_i^t(s, a) \right],
\]
for all \( \pi' \in \Pi \), \( \pi^t \in \Pi_{-i} \) and \( \omega_i \in \Omega_i \).

Given that Assumptions 1, 2 and 3 hold, we can proceed with our argument. We will prove the following result.

**Lemma 8** For every player \( i \) and round \( t \), and for any given stage game matrix of \( Q \)-values \( \bar{Q}_i(s) \), let \( \mathcal{B}_i : \mathcal{Q} \to \mathcal{Q} \) be an operator on the \( Q \)-value function space \( \mathcal{Q} \), defined as
\[
\mathcal{B}_i^t \bar{Q}_i(s, a) := \mathcal{R}_i(s, a) + \gamma E_{K \sim \omega_i^t} \left[ \sum_{a' \in A} \pi_T^t(a'_{-K}|s) \bar{\pi}_K^t(a_K|s) \bar{Q}_i^t(s, a') \right],
\]
for any given \((s, a)\)-tuple. Then, \( \mathcal{B}_i^t \) is a contraction operator with respect to the \( l_\infty \) norm.

In order to prove the above lemma, we will make use of the following standard result on pseudo-contractions.

**Lemma 9 (Lemma 8 of Hu and Wellman [2003])** Assume that \( \alpha_t \) satisfy Assumption 2 and that the mapping \( \mathcal{B}^t : \mathcal{Q} \to \mathcal{Q} \) satisfies the following condition: there exists a number \( 0 < \gamma < 1 \) and a sequence \( \gamma_t \geq 0 \) converging to zero with probability 1 such that
\[
\|\mathcal{B}^t Q - \mathcal{B}^t Q^*\| \leq \gamma \|Q - Q^*\| + \gamma_t,
\]
for all \( Q \in \mathcal{Q} \) and \( Q^* \) such that \( Q^* = E[\mathcal{B}^t Q^*] \), then the iteration defined by
\[
Q_{t+1} = (1 - \alpha_t)Q_t + \alpha_t \mathcal{B}^t Q_t
\]
converges to \( Q^* \) with probability 1.

Now we are ready to prove Lemma 8. In order to do so, we need to show that its conditions are satisfied for our setting.

**Proof:** Let \( (\bar{Q}_i)^t \) and \( (\bar{Q}'_i)^t \) be two given sets of \( Q \)-value functions. Denote by \( (\pi_T^t, \pi^t, \omega_i^t) \) and \( (\pi_T^t, \pi^t, \omega_i^t) \) their corresponding ARNEQ policies, respectively. Let \((s, a)\) be a given state-action tuple and suppose we have
\[
\mathcal{B}_i^t \bar{Q}_i(s, a) \geq \mathcal{B}_i^t \bar{Q}'_i(s, a).
\]
If the first part of Assumption 3 holds, then we have
\[
0 \leq \left| \mathcal{B}_i^t \bar{Q}_i(s, a) - \mathcal{B}_i^t \bar{Q}'_i(s, a) \right| = \gamma E_{K \sim \omega_i^t} \left[ \sum_{a' \in A} \pi_T^t(a'_{-K}|s) \bar{\pi}_K^t(a_K|s) \bar{Q}_i(s, a') \right].
\]
A similar argument is used for the case when we have $i$'s policy

\[ \pi_i(s,a) \]

and the third inequality follows by the second part of Assumption 3.

If the second part of Assumption 3 holds, then, instead we have

\[ \max_{K \in \mathcal{K}} \left| \sum_{a' \in A} \pi_K(a_{-K}'|s)\pi_i(a_K'|s)Q_i(s,a') - \pi_K(a_{-K}'|s)\pi_i(a_K'|s)Q_i(s,a') \right| = \max_{K \in \mathcal{K}} \left| \sum_{a' \in A} \pi_K(a_{-K}'|s)\pi_i(a_K'|s)Q_i(s,a') - \pi_K(a_{-K}'|s)\pi_i(a_K'|s)Q_i(s,a') \right| \]

\[ \leq \max_{a' \in A} \left| Q_i(s,a') - Q_i(s,a') \right| \]

\[ \leq \|Q_i - Q_i\|_\infty. \]

If the second part of Assumption 3 holds, then, instead we have

\[ 0 \leq \mathcal{E}(Q_i(s,a) - Q_i(s,a)) \]

\[ = \max_{K \in \mathcal{K}} \left| \sum_{a' \in A} \pi_K(a_{-K}'|s)\pi_i(a_K'|s)Q_i(s,a') - \pi_K(a_{-K}'|s)\pi_i(a_K'|s)Q_i(s,a') \right| \]

\[ \leq \max_{a' \in A} \left| Q_i(s,a') - Q_i(s,a') \right| \]

\[ \leq \|Q_i - Q_i\|_\infty, \]

where the second inequality follows from the equilibrium definition with respect to player $i$'s policy and the third inequality follows by the second part of Assumption 3.

A similar argument is used for the case when we have

\[ \mathcal{E}(Q_i(s,a) - Q_i(s,a)). \]

Since the chosen $(s,a)$-tuple was arbitrary, we conclude that

\[ \|Q_i - Q_i\|_\infty \leq \gamma \|Q_i - Q_i\|_\infty. \]
Thus, we have shown that the operator $B_t^l$ is a contraction. Recall that this operators has $Q^*_t$ as a fixed point. Furthermore, the rates $\alpha_t$, $t \geq 1$, satisfy the requirements of Lemma 9. Hence, the result of Theorem 2 follows.

D Experiments: Environments

D.1 Independent Spread

This environment modifies the Spread environment from Lowe et al. [2017] in such a way, that all agents pursue an individual goal, instead of cooperating toward one shared objective. Each agent strives to reduce the distance to a landmark that is designated to them while simultaneously avoiding collisions with other agents. Therefore, the reward of each agent consists of the negative $L_2$-distance to their designated landmark, coupled with a large penalty of 10 for colliding with agents. Agents observe their position, as well as the relative position of their designated landmark and all peer agents. The action space consists of no-action as well as increasing the velocity in one of four directions.

D.2 Multi-Agent Ant

We consider a cooperative multi-agent extension of the MuJoCo Ant environment [Todorov et al., 2012] as proposed in Peng et al. [2021]. Agents control one leg of the ant, and need to cooperate with the other legs to maximize the distance traveled forward, while maintaining balance. Consequentially, each agent receives a shared reward, which consists of the forward momentum compared to the preceding frame and a small incentive of 0.1 reward for not falling over. We removed the penalty for large actions and high external contact, as the adversary could easily exploit this reward function without actually impacting the ant’s objective and there is defence mechanism for the other agents. Agents observe the position and orientation of all joints and can apply torque to the joint between the torso and upper leg and between the upper and lower leg of the leg they control.

D.3 Pursuit

We finally consider the cooperative Pursuit environment Gupta et al. [2017]. Each agent assumes the role of one of the eight pursuers aiming to catch 30 randomly moving evaders. Evaders are captured if it is enclosed on all sides by pursuers or walls. Pursuers receive the reward of 5 for every successful capture, as well as an urgency reward of $-0.1$ every time step there is still prey left to be captured. Observations consist of all pursuers, evaders, and walls within a 7x7 grid around the agent. The available actions are staying at the current position or moving one field to the left, right, up, or down.
E Experiments: Hyperparameters

Hyperparameters for the PPO algorithm for all environments are given in Table 7.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Value for Spread</th>
<th>Value for Ant</th>
<th>Value for Pursuit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total timesteps</td>
<td>800K</td>
<td>13M</td>
<td>10M</td>
</tr>
<tr>
<td>Batch size</td>
<td>4000</td>
<td>65536</td>
<td>5000</td>
</tr>
<tr>
<td>Minibatch size</td>
<td>128</td>
<td>4096</td>
<td>500</td>
</tr>
<tr>
<td>Epochs per update</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Learning rate $\alpha$</td>
<td>0.0003</td>
<td>$5 \times 10^{-5}$</td>
<td>$5 \times 10^{-5}$</td>
</tr>
<tr>
<td>Discount Factor $\gamma$</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>Clipping parameter $A$</td>
<td>0.3</td>
<td>0.2</td>
<td>0.3</td>
</tr>
<tr>
<td>Advantage estimation discount $\lambda$</td>
<td>1.0</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>Entropy coefficient</td>
<td>0.0</td>
<td>0.0</td>
<td>0.01</td>
</tr>
<tr>
<td>Value function loss coefficient</td>
<td>1.0</td>
<td>0.5</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Figure 7: Hyperparameters for Spread, Ant, and Pursuit

F Experiments: Computing Infrastructure

Computing infrastructure for the Independent Spread and Pursuit environment:

- **GPU:** None
- **CPU:** Intel Xeon E5-2667 v2
- **Memory:** 256GB, DDR3, 1866 MHz, ECC
- **Operating System:** Debian

The computing infrastructure for the Ant environment:

- **GPU:** V100 Nvidia Tesla GPU 32GB
- **CPU:** Intel Xeon Gold 6134M
- **Memory:** 768GB, DDR4 2666MT/s, ECC
- **Operating System:** Debian

G Experiments: Runtime

Runtime for all algorithms and environments are given in Table 8.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Runtime for Spread</th>
<th>Runtime for Ant</th>
<th>Runtime for Pursuit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent Learning</td>
<td>27m28s</td>
<td>1h45m58s</td>
<td>10h6m</td>
</tr>
<tr>
<td>fixed-k</td>
<td>22m36s</td>
<td>1h46m48s</td>
<td>10h9m</td>
</tr>
<tr>
<td>ART</td>
<td>34m38s</td>
<td>1h50m24s</td>
<td>10h55m59s</td>
</tr>
</tbody>
</table>

Figure 8: Runtime of algorithms for Spread, Ant and Pursuit