Data Poisoning to Fake a Nash Equilibrium in Markov Games

Young Wu, Jeremy McMahan, Xiaojin Zhu, Qiaomin Xie

University of Wisconsin - Madison
yw@cs.wisc.edu, jmcmahan@wisc.edu, jerryzhu@cs.wisc.edu, qiaomin.xie@wisc.edu

Abstract

We characterize offline data poisoning attacks on Multi-Agent Reinforcement Learning (MARL), where an attacker may change a data set in an attempt to install a (potentially fictitious) unique Markov-perfect Nash equilibrium for a two-player zero-sum Markov game. We propose the unique Nash set, namely the set of games, specified by their Q functions, with a specific joint policy being the unique Nash equilibrium. The unique Nash set is central to poisoning attacks because the attack is successful if and only if data poisoning pushes all plausible games inside the set. The unique Nash set generalizes the reward polytope commonly used in inverse reinforcement learning to MARL. For zero-sum Markov games, both the inverse Nash set and the set of plausible games induced by data are polytopes in the Q function space. We exhibit a linear program to efficiently compute the optimal poisoning attack. Our work sheds light on the structure of data poisoning attacks on offline MARL, a necessary step before one can design more robust MARL algorithms.

1 Introduction

Data poisoning attacks have been well studied in supervised learning (intentionally forcing the learner to train a wrong classifier) and reinforcement learning (wrong policy) (Banihashem et al. 2022; Huang and Zhu 2019; Liu and Lai 2021; Rakhsha et al. 2021a,b, 2020; Sun, Huo, and Huang 2020; Zhang et al. 2020; Ma et al. 2019; Rangi et al. 2022; Zhang and Parkes 2008; Zhang, Parkes, and Chen 2009). Can data poisoning attacks be a threat to Markov Games, too? This paper answers this question in the affirmative: Under mild conditions, an attacker can force two game-playing agents to adopt any fictitious Nash Equilibrium (NE), which does not need to be a true NE of the original Markov Game. Furthermore, the attacker can achieve this goal while minimizing its attack cost, which we define below. Clearly, such power poses a threat to the security of Multi-Agent Reinforcement Learning (MARL).

Formally, we study two-player zero-sum Markov game offline data poisoning, stated as the following.

Problem Statement: Offline Data Poisoning. Let D be a dataset $\{(s^{(k)}, \mathbf{a}^{(k)}, r^{(k)})\}_{k=1}^{K}$ with K tuples of state s, joint action $\mathbf{a} = (a_1, a_2)$, rewards $(r, -r)$. The attacker's target NE is an arbitrary pure strategy pair $\pi^\dagger := (\pi_1^\dagger, \pi_2^\dagger)$. The attacker can poison D into another dataset $D^\dagger$ by paying cost $C(D, D^\dagger)$. Two MARL agents then receive $D^\dagger$ instead of D. The attacker aims to enforce that the agents learn the target NE $\pi^\dagger$ from $D^\dagger$ while minimizing C.

This problem is not well studied in the literature. Naive approaches – such as modifying all the actions in the dataset to those specified by the target policy $(\pi_1^\dagger, \pi_2^\dagger)$ – might not achieve the attack goal for MARL learners who assign penalties due to the lack of data coverage. Modifying all the rewards in the dataset that coincide with the target policy to the reward upper bound might be feasible, but would not be optimal in terms of attack cost C. Results on data poisoning against single-agent RL cannot be directly applied to the multi-agent case. In particular, there are no optimal policies in MARL, and equilibrium policies are computed instead. There could be multiple equilibria that are significantly different, and consequently, installing a target policy as the unique equilibrium is difficult. To resolve this issue, we provide a novel characterization of when a zero-sum Markov game has a unique Markov perfect Nash equilibrium.

Our framework can be summarized by the mnemonic "ToM moves to the UN". (i) UN stands for the Unique Nash set, which is the set of Q functions that make the target $\pi^\dagger$ the unique NE. Uniqueness is crucial for the attacker to ensure that MARL agents choose the target NE with certainty, without breaking ties arbitrarily among multiple NEs. (ii) ToM stands for the attacker's Theory of Mind of the MARL agents, namely the plausible set of Q functions that the attacker believes the agents will entertain upon receiving the poisoned dataset $D^\dagger$. (iii) The attack is successful if, by controlling $D^\dagger$, the ToM set is moved inside the UN set. A successful attack with the smallest cost $C(D, D^\dagger)$ is optimal.

Adversarial attacks on MARL have been studied in some recent work (Ma, Wu, and Zhu 2021; Gleave et al. 2019; Guo et al. 2021), but we are only aware of one previous work (Wu et al. 2023) on offline reward poisoning against MARL. Nonetheless, they require a strong assumption of full data coverage, and that the learners compute the Dominant Strategy Markov Perfect Equilibrium (DSMPE). In contrast, we do not require full coverage, and we consider a weaker solution concept, Markov Perfect Equilibrium (MPE). Our general attack framework also accommodates other forms of data poisoning.

Understanding adversarial attacks in the multi-agent setting is critical since many real-life applications of MARL problems are susceptible to adversarial attacks. Examples of two-player zero-sum games include board games such as GO and Chess (Silver et al. 2017, 2016), where the learners use historical game plays as training data and an attacker can potentially alter the data to change the behavior of the trained agents. In the case of competitive robotics, for example, robot soccer (Gu et al. 2017; Riedmiller et al. 2009; Kober, Bagnell, and Peters 2013), they are trained on offline datasets and the attacker can mislead the trained policies by modifying the training sets. For finance application, especially algorithmic or high-frequency stock or option trading (Lee et al. 2007; Lee and O 2002) that are usually trained on historical prices, if the database is corrupted by an attacker, the learned trading strategies can be suboptimal as well. There are also examples of multi-player games that have two-player games as special cases, for example, video games (Vinyals et al. 2019; Jaderberg et al. 2019; Berner et al. 2019), card games (Brown and Sandholm 2019; Brown, Sandholm, and Machine 2017), autonomous driving (Shalev-Shwartz, Shammah, and Shashua 2016), automated warehouses (Yang, Juntao, and Lingling 2020), and economic policymaking, which can all be trained on offline datasets and become vulnerable to adversarial attacks. In all of the above MARL applications, the threat of adversarial attacks has not been investigated.

Our contributions include a unified framework for offline data poisoning attacks, and in particular, a linear program formulation that efficiently solves the reward poisoning problem for two-player zero-sum Markov games. On the technical side, we present a geometric characterization of a deterministic policy being the unique Markov perfect Nash equilibrium of zero-sum Markov games. In addition, we demonstrate that for a class of MARL learners that compute equilibrium policies based on games within confidence regions around a point estimate of the Q function of the Markov game, an attack with appropriate parameters on these learners would success on most of the model-based and model-free offline MARL learners proposed in the literature.

2 Offline Attack on a Normal-form Game

The Unique Nash Set (UN) of a Normal-form Game

We present the main components of our approach with a normal-form game, in particular, a two-player zero-sum game is a tuple $(\mathcal{A}, R)$, where $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$ is the joint action space and $R : \mathcal{A} \to [-b, b]$ is the mean reward function. We use $b = \infty$ in the case of unbounded rewards. Given $\mathcal{A}$, we denote the set of reward functions by $\mathcal{R} = \{R : \mathcal{A} \to \mathbb{R}\}$.

A pure strategy profile $\pi = (\pi_1, \pi_2)$ is a pair of actions, where $\pi_i \in \mathcal{A}_i$ specifies the action for agent $i \in \{1, 2\}$. We focus on pure strategies, but we allow mixed strategies in which case we use the notation $\pi_i(a_i)$ to represent the probability of i using the action $a_i \in \mathcal{A}_i$, and R computes the expected reward $R(\pi) := \sum_{a_1 \in \mathcal{A}_1, a_2 \in \mathcal{A}_2} \pi_1(a_1)\, \pi_2(a_2)\, R((a_1, a_2))$.

Definition 1 (Nash Equilibrium)**.** *A Nash equilibrium (NE) of a normal-form game $(\mathcal{A}, R)$ is a mixed strategy profile π that satisfies,*

$$\begin{aligned} R((\pi_1, a_2)) = R(\pi) = R((a_1, \pi_2)), \\ \forall\, a_1 : \pi_1(a_1) > 0, a_2 : \pi_2(a_2) > 0, \\ R((\pi_1, a_2)) \leqslant R(\pi) \leqslant R((a_1, \pi_2)), \\ \forall\, a_1 : \pi_2(a_1) = 0, a_2 : \pi_2(a_1) = 0, \end{aligned}$$

in particular, for a pure strategy profile π, it is a Nash equilibrium if,

$$\begin{aligned} R((\pi_1, a_2)) \leqslant R(\pi) \leqslant R((a_1, \pi_2)), \\ \forall\, a_1 \neq \pi_1, a_2 \neq \pi_2. \end{aligned} \tag{1}$$

We define $\mathcal{N}(R) := \{\pi : \pi \text{ is an NE of } (\mathcal{A}, R)\}$ to be the set of all Nash equilibria of a normal-form game $(\mathcal{A}, R)$.

Now, we define the inverse image of $\mathcal{N}$ from a single pure strategy profile π back to the space of reward functions to be the unique Nash set.

Definition 2 (Unique Nash)**.** *The unique Nash set of a pure strategy profile π is the set of reward functions R such that $(\mathcal{A}, R)$ has a unique Nash equilibrium π,*

$$\mathcal{U}(\pi) := \mathcal{N}^{-1}(\{\pi\}) = \{R \in \mathcal{R} : \mathcal{N}(R) = \{\pi\}\}. \tag{2}$$

To characterize $\mathcal{U}(\pi)$, we note that for normal-form games, a pure strategy profile π is the unique Nash equilibrium of a game if and only if it is a strict Nash equilibrium, which is defined as a policy π that satisfies (1) with strict inequalities.

Proposition 1 (Unique Nash Polytope)**.** *For any pure strategy profile π,*

$$\begin{aligned} \mathcal{U}(\pi) &= \{R \in \mathcal{R} : \pi \text{ is a strict NE of } (\mathcal{A}, R)\} \\ &= \{R \in \mathcal{R} : R((\pi_1, a_2)) < R(\pi) < R((a_1, \pi_2)), \\ &\qquad \forall\, a_1 \neq \pi_1, a_2 \neq \pi_2\}. \end{aligned} \tag{3}$$

Here, the uniqueness is among all Nash equilibria including mixed-strategy Nash equilibria. The proof of the equivalence between (2) and (3) is in the appendix. We restrict our attention to pure-strategy equilibria and defer the discussion of mixed strategy profiles to the last section.

To avoid working with strict inequalities, we define a closed subset of $\mathcal{U}(\pi)$ of reward functions that lead to strict Nash equilibria with an ι reward gap, which means all strict inequalities in (3) are satisfied with a gap of at least ι, for some $\iota > 0$.

Definition 3 (Iota Strict Unique Nash)**.** *For $\iota > 0$, the ι strict unique Nash set of a pure strategy profile π is, $\underline{\mathcal{U}}(\pi; \iota) :=$*

$$\begin{aligned} \{R \in \mathcal{R} : R((\pi_1, a_2)) + \iota \leqslant R(\pi) \leqslant R((a_1, \pi_2)) - \iota, \\ \forall\, a_1 \neq \pi_1, a_2 \neq \pi_2\}. \end{aligned} \tag{4}$$

For every pure strategy profile π and $\iota > 0$, we have $\underline{\mathcal{U}}(\pi; \iota) \subset \mathcal{U}(\pi)$, and the set is a polytope in $\mathcal{R}$.

The Attacker's Theory of Mind (ToM) for Offline Normal-form Game Learners

We provide a model of the attacker's theory of mind of the victim, which is the attacker's belief about the learning algorithm the victim uses. In particular, the attacker is not required to have complete knowledge of the victims' learning algorithms: only an approximation (of theory of mind) is needed. Formally, we define the theory-of-mind set as the set of plausible rewards that the victim uses based on the given training dataset, and we assume that the victims compute the Nash equilibria based on the reward functions estimated from a dataset $D \in \mathcal{D}$, where $\mathcal{D}$ is the set of possible datasets with K episodes in the form $\left\{\left(\mathbf{a}^{(k)}, r^{(k)}\right)\right\}_{k=1}^{K}$, with $\mathbf{a}^{(k)} \in \mathcal{A}$ and $r^{(k)} \in [-b, b]$ for every $k \in [K]$.

Definition 4 (Theory of Mind)**.** *Given a dataset $D \in \mathcal{D}$, the theory-of-mind set $\mathcal{T}(D) \subseteq \mathcal{R}$ is the set of plausible reward functions that the victims estimate based on D to compute their equilibria. In particular, if the victims learn an action profile π, then $\pi \in \bigcup_{R \in \mathcal{T}(D)} \mathcal{N}(R)$.*

The theory-of-mind sets can be arbitrary and could be difficult to work with. We define an outer approximation the set that is a hypercube in $\mathcal{R}$.

Definition 5 (Outer Approximation of Theory of Mind)**.** *An outer approximation of $\mathcal{T}(D)$ is a set denoted by $\overline{\mathcal{T}}(D)$ that satisfies $\mathcal{T}(D) \subseteq \overline{\mathcal{T}}(D)$ for every $D \in \mathcal{D}$, and can be written in the form, $\overline{\mathcal{T}}(D) :=$*

$$\left\{R \in \mathcal{R} : \left|R(\boldsymbol{a}) - \hat{R}(\boldsymbol{a})\right| \leqslant \rho^{(R)}(\boldsymbol{a}), \forall\, \boldsymbol{a} \in \mathcal{A}\right\}, \quad (5)$$

for some point estimate $\hat{R}$ and radius $\rho^{(R)}$.
We call $\overline{\mathcal{T}}(D)$ a linear outer approximation if $\hat{R}$ is linear in $\left\{r^{(k)}\right\}_{k=1}^{K.}$

We present a few examples of the theory-of-mind sets as follows.

Example 1 (Theory of Mind for Maximum Likelihood Victims)**.** *Given a dataset $D \in \mathcal{D}$, if the attacker believes the victims are maximum likelihood learners, then $\mathcal{T}(D)$ is a singleton R^{MLE}, where, for every $\boldsymbol{a} \in \mathcal{A}$,*

$$R^{MLE}(\boldsymbol{a}|r) := \begin{cases} \dfrac{1}{N(\boldsymbol{a})} \displaystyle\sum_{k=1}^{K} r^{(k)} \mathbb{I}_{\left\{\boldsymbol{a}^{(k)} = \boldsymbol{a}\right\}} & \text{if } N(\boldsymbol{a}) > 0 \\ 0 & \text{if } N(\boldsymbol{a}) = 0 \end{cases}$$

$$N(\boldsymbol{a}) := \sum_{k=1}^{K} \mathbb{I}_{\left\{\boldsymbol{a}^{(k)} = \boldsymbol{a}\right\}}. \quad (6)$$

The smallest outer approximation $\overline{\mathcal{T}}(D)$ can be specified using $\hat{R} = R^{MLE}$ and $\rho^{(R)} = 0$, and $\overline{\mathcal{T}}$ is linear since (6) *is linear in $\left\{r^{(k)}\right\}_{k=1}^{K}$.*

Example 2 (Theory of Mind for Pessimistic Optimistic Victims)**.** *Given a dataset $D \in \mathcal{D}$, if the attacker believes the victims are learners that use pessimism and optimism by adding and subtracting bonus terms and estimating one or two games, as in (Cui and Du 2022), then $\mathcal{T}(D)$ may contain two reward functions $\underline{R}$ and $\overline{R}$, where for every $\boldsymbol{a} \in \mathcal{A}$,*

$$\underline{R}(\boldsymbol{a}|r) := R^{MLE}(\boldsymbol{a}|r) - \beta(\boldsymbol{a})$$
$$\overline{R}(\boldsymbol{a}|r) := R^{MLE}(\boldsymbol{a}|r) + \beta(\boldsymbol{a}), \quad (7)$$

with $\beta(\boldsymbol{a}) = \dfrac{c}{\sqrt{N(\boldsymbol{a})}}$ being the bonus term, for some constant c.
The smallest outer approximation $\overline{\mathcal{T}}(D)$ can be specified using $\hat{R} = R^{MLE}$ and $\rho^{(R)}(\boldsymbol{a}) = \beta(\boldsymbol{a})$ for every $\boldsymbol{a} \in \mathcal{A}$, and $\overline{\mathcal{T}}$ is linear since (6) *and* (7) *are both linear in $\left\{r^{(k)}\right\}_{k=1}^{K}$.*

Example 3 (Theory of Mind for Data Splitting Victims)**.** *Given a dataset $D \in \mathcal{D}$, if the attacker believes the victims use maximum likelihood estimates on a subsample of the D, similar to the data-splitting procedure in (Cui and Du 2022), then $\overline{\mathcal{T}}(D)$ could be viewed as a high-probability set of rewards that the victims are estimating and $\rho^{(R)}$ would be half of the confidence interval width for the mean of the subsample around the mean of the complete dataset R^{MLE}.*

The Cheapest Way to Move ToM into UN for Normal-form Games

The goal of the attacker is to install a specific action profile as the unique Nash equilibrium of the game learned by the victim while minimally modifying the training data. We consider a general attacker's cost as a function $C : \mathcal{D} \times \mathcal{D} \to \mathbb{R}^{+}$ where $C\left(D, D^{\dagger}\right)$ is the cost of modifying the dataset from D to $D^{\dagger}$. Given the original data set $D \in \mathcal{D}$, the attacker's attack modality $\mathcal{D}(D)$ is the set of datasets the attacker is allowed to modify the original dataset to. For the reward poisoning problem, where $\mathcal{D}^{(R)}(D)$ is all possible datasets in which only rewards are modified from $r^{(k)}$ to $r^{\dagger,(k)}$, we consider the following cost function.

Example 4 (L_1 Cost Function)**.** *For reward poisoning problems, we define the L_1 cost of modifying the dataset from $D = \left\{\left(\boldsymbol{a}^{(k)}, r^{(k)}\right)\right\}_{k=1}^{K}$ to $D^{\dagger} = \left\{\left(\boldsymbol{a}^{(k)}, r^{\dagger,(k)}\right)\right\}_{k=1}^{K}$ by $C^{(1)}\left(D, D^{\dagger}\right) := \displaystyle\sum_{k=1}^{K} \left|r^{(k)} - r^{\dagger,(k)}\right|$.*

Remark 1. *In our framework, the attacker's cost function can be an arbitrary convex function, which can accommodate various settings, for example, when the attacker has a limited budget or when the attacker can only change a limited number of entries: the optimization will remain a convex program with linear constraints. L_1 loss is used for simplicity so that our attack optimization is a linear program, it could be relaxed, although then the optimization would be harder to solve.*

Now, given the original dataset D and the attacker's target action profile $\pi^{\dagger}$, we formally state the attacker's problem as finding the cheapest (minimal cost) way to move $\mathcal{T}(D)$ into $\mathcal{U}\left(\pi^{\dagger}\right)$.

Definition 6 (Attacker's Problem)**.** *The attacker's problem with the target action profile $\pi^{\dagger}$ is,*

$$\inf_{D^{\dagger} \in \mathcal{D}(D)} C\left(D, D^{\dagger}\right) \quad (8)$$

$$s.t.\ \mathcal{T}\left(D^{\dagger}\right) \subseteq \mathcal{U}\left(\pi^{\dagger}\right).$$

In general, (8) cannot be solved efficiently, but for reward poisoning problems with L_1 cost objective, we can relax the attacker's problem using ι strict unique Nash sets, which is a polytope described by (4), and a linear outer approximation of the theory-of-mind set, a hypercube described by (5), which can be converted into a linear program and solved efficiently. We state this observation as the following proposition and depict the relationship between the sets in Figure 1.

Proposition 2 (Reward Poisoning Linear Program)**.** *Given $\iota > 0$ and a linear $\overline{\mathcal{T}}$, the following problem is a relaxation of the attacker's reward poisoning problem and can be converted into a linear program,*

$$\min_{D^{\dagger} \in \mathcal{D}^{(R)}(D)} C^{(1)}\left(D, D^{\dagger}\right) \quad (9)$$
$$s.t.\ \overline{\mathcal{T}}\left(D^{\dagger}\right) \subseteq \underline{\mathcal{U}}\left(\pi^{\dagger}; \iota\right).$$

In Figure 1, given a dataset D, the general attacker's problem (8) of moving $\mathcal{T}(D)$ (light green) to $\mathcal{T}\left(D^{\dagger}\right)$ (light red) such that it is inside $\mathcal{U}\left(\pi^{\dagger}\right)$ (light blue) while minimizing the distance from D to $D^{\dagger}$ is often intractable. We construct a relaxed problem (9) of moving $\overline{\mathcal{T}}(D)$ (green) to $\overline{\mathcal{T}}\left(D^{\dagger}\right)$ (red) such that it is inside $\underline{\mathcal{U}}\left(\pi^{\dagger}\right)$ (blue), in which all sets are polytopes and thus can be converted to a linear program for linear costs and linear theory-of-mind mappings.

In the appendix, we provide the complete linear program and show that the solution of (9) is feasible for (8). The optimality of the linear program solution depends on how close the outer approximation of the theory-of-mind set is, and in the case when the theory-of-mind set is already a hypercube, the infimum in (8) can be achieved by taking the limit as $\iota \to 0$.

Example 5 (Maximum Likelihood Centered Linear Program)**.** *In the case $\hat{R} = R^{MLE}$ in the theory-of-mind set,* (9) *is given by,*

$$\min_{r^{\dagger} \in [-b,b]^K} \sum_{k=1}^{K} \left| r^{(k)} - r^{\dagger,(k)} \right| \quad (10)$$
$$s.t.\ R^{MLE}\left(r^{\dagger}\right) \text{ is linear in } r^{\dagger} \text{ satisfying } (6)$$
$$\overline{R}\left(r^{\dagger}\right) \text{ and } \underline{R}\left(r^{\dagger}\right) \text{ satisfying } (5)$$
$$\text{are upper and lower bounds of } \overline{\mathcal{T}}\left(r^{\dagger}\right)$$
$$\left[\overline{R}\left(r^{\dagger}\right), \underline{R}\left(r^{\dagger}\right)\right] \text{ is in } \underline{\mathcal{U}}\left(\pi^{\dagger}\right) \text{ satisfying } (4)$$

Since $\overline{\mathcal{T}}\left(r^{\dagger}\right)$ is a hypercube and $\underline{\mathcal{U}}\left(\pi^{\dagger}\right)$ is a polytope, the fact that the corners of the hypercube are inside the unique Nash set if and only if every element in the hypercube is in the unique Nash set implies that the constraint in (9) *is satisfied. Technically, we only require one corner of the hypercube to be inside the unique Nash polytope, as shown in Figure 1, and we leave the details to the proof of Proposition 2 in the appendix. Then, because the objective and all of the constraints in* (10) *are linear in $r^{\dagger}, \overline{R}, \underline{R}$ and R^{MLE}, this problem is a linear program.*

3 Offline Attack on a Markov Game

The Unique Nash Set (UN) of a Markov Game

We now consider the attacker's problem for Markov games. A finite-horizon two-player zero-sum Markov game G is a tuple $(\mathcal{S}, \mathcal{A}, P, R, H)$, where $\mathcal{S}$ is the finite state space; $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$ is the joint action space; $P = \{P_h : \mathcal{S} \times \mathcal{A} \to \Delta\mathcal{S}\}_{h=1}^{H}$ is the transition function with the initial state distribution $P_0 \in \Delta\mathcal{S}$; and $R = \{R_h : \mathcal{S} \times \mathcal{A} \to [-b, b]\}_{h=1}^{H}$ is the mean reward function; and H is the finite time horizon.

A deterministic Markovian policy $\pi = (\pi_1, \pi_2)$ is a pair of policies, where $\pi_i = \{\pi_{i,h} : \mathcal{S} \to \mathcal{A}_i\}_{h=1}^{H}$ for $i \in \{1, 2\}$, and $\pi_{i,h}(s)$ specifies the action used in period h and state s. Again, we focus on deterministic policies, but we allow stochastic policies in which case we use the notation $\pi_i = \{\pi_{i,h} : \mathcal{S} \to \Delta\mathcal{A}_i\}_{h=1}^{H}$ for $i \in \{1, 2\}$, and $\pi_{i,h}(s)(a_i)$ represent the probability of i using the action $a_i \in \mathcal{A}_i$ in period h state s.

The Q function is defined as, for every $h \in [H], s \in \mathcal{S}, \mathbf{a} \in \mathcal{A}$, we write

$$Q_h(s, \mathbf{a}) := R_h(s, \mathbf{a}) + \sum_{s' \in \mathcal{S}} P_h\left(s' | s, \mathbf{a}\right) \max_{\pi_1 \in \Delta\mathcal{A}_1} \min_{\pi_2 \in \Delta\mathcal{A}_2} Q_{h+1}\left(s', \pi\right), \quad (11)$$

with the convention $Q_{H+1}(s, \mathbf{a}) = 0$, and in the case π is stochastic, we write, $Q_h(s, \pi_h(s)) :=$

$$\sum_{a_1 \in \mathcal{A}_1} \sum_{a_2 \in \mathcal{A}_2} \pi_{1,h}(s)(a_1)\, \pi_{2,h}(s)(a_2)\, Q_h(s, (a_1, a_2)).$$

Given $\mathcal{S}, \mathcal{A}, H$, we denote the set of Q functions by $\mathcal{Q} = \left\{\{Q_h : \mathcal{S} \times \mathcal{A} \to \mathbb{R}\}_{h=1}^{H}\right\}$. Technically, $\mathcal{Q}$ is not the set of proper Q functions of Markov games since both the reward functions and the transition functions do not have to be proper, and given $Q \in \mathcal{Q}$, we may not be able to construct a Markov game that induces Q. This choice is made to accommodate both model-based and model-free victims who may or may not estimate the rewards and transitions explicitly from the dataset.

A stage game of a Markov game G in period $h \in [H]$, state $s \in \mathcal{S}$ under policy π is a normal form game $(\mathcal{A}, Q_h(s))$, where $\mathcal{A}$ is the joint action space of G; and $Q_h(s)$ is the mean reward function, meaning the reward from action profile $\mathbf{a} \in \mathcal{A}$ is $Q_h(s, \mathbf{a})$. We define Markov perfect equilibria as policies in which the action profile used in every stage game is a Nash equilibrium.

Definition 7 (Markov Perfect Equilibrium)**.** *A Markov perfect equilibrium (MPE) policy π is a policy such that $\pi_h(s)$ is a Nash equilibrium in the stage game $(\mathcal{A}, Q_h(s))$.*
We define the set of all Markov perfect equilibria policies of a Markov game that induces $Q \in \mathcal{Q}$ by $\mathcal{M}(Q) = \{\pi : \pi$ is an MPE of a Markov game with Q function $Q\}$.

We note that Nash equilibria for Markov games can also be defined by converting the Markov game into a single normal-form game, but we only consider Markov perfect equilibria since Nash equilibria that are not Markov perfect

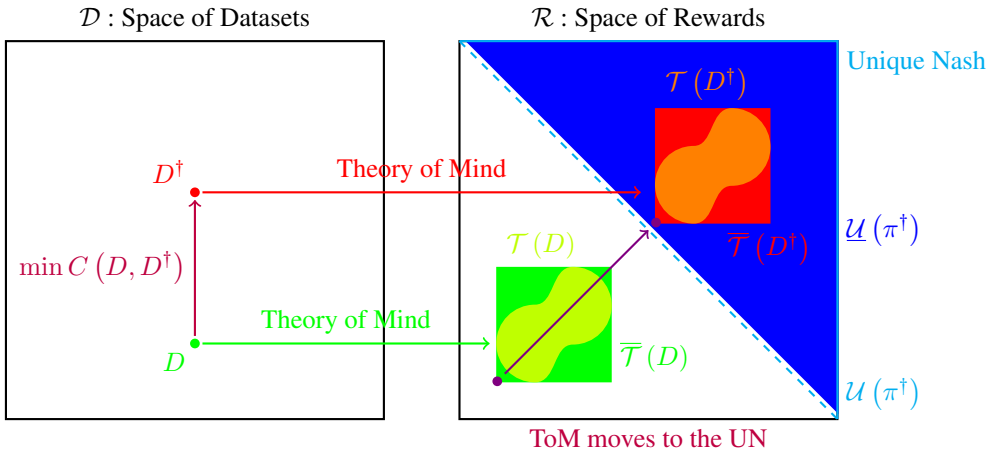


Figure 1: Attacker's Problem

require coordination and commitment to policies in stage games that are not visited along equilibrium paths, which is not realistic in the MARL setting.

We define the unique Nash set for Markov games as follows.

Definition 8 (Unique Nash)**.** *The unique Nash set of a deterministic Markovian policy π for a Markov game G is the set of Q functions such that π is the unique Markov perfect equilibrium under policy π,*

$$\mathcal{U}(\pi) := \mathcal{M}^{-1}(\{\pi\}) = \{Q \in \mathcal{Q} : \mathcal{M}(Q) = \{\pi\}\}. \quad (12)$$

Next, we extend the characterization of the unique Nash set for normal-form games to the Markov game setting.

Theorem 1 (Unique Nash Polytope)**.** *For any deterministic policy π,*

$$\begin{aligned}\mathcal{U}(\pi) &= \{Q \in \mathcal{Q} : \pi_h(s) \text{ is a strict NE of } (\mathcal{A}, Q_h(s)), \\ &\quad \forall\, h \in [H], s \in \mathcal{S}\} \\ &= \{Q \in \mathcal{Q} : Q_h(s,(\pi_{1,h}(s), a_2)) < Q_h(s, \pi(s)) \\ &\quad < Q_h(s,(a_1, \pi_{2,h}(s))), \forall\, a_1 \neq \pi_{1,h}(s), \\ &\quad , a_2 \neq \pi_{2,h}(s), h \in [H], s \in \mathcal{S}\}, \end{aligned} \quad (13)$$

We show the equivalence between (12) and (13) in the proof of Theorem 1 in the appendix. To avoid working with strict inequalities in (13), we again define the ι strict version of the unique Nash polytope.

Definition 9 (Iota Strict Unique Nash)**.** *For $\iota > 0$, the ι strict unique Nash set of a deterministic policy π is, $\underline{\mathcal{U}}(\pi;\iota) :=$*

$$\begin{aligned}&:= \{Q \in \mathcal{Q} : Q_h(s,(\pi_{1,h}(s), a_2)) + \iota \leqslant Q_h(s, \pi(s)) \\ &\quad \leqslant Q_h(s,(a_1, \pi_{2,h}(s))) - \iota, \forall\, a_1 \neq \pi_{1,h}(s), \\ &\quad a_2 \neq \pi_{2,h}(s), h \in [H], s \in \mathcal{S}\}. \end{aligned} \quad (14)$$

For every deterministic policy π and $\iota > 0$, we have $\underline{\mathcal{U}}(\pi;\iota) \subset \mathcal{U}(\pi)$, and the set is a polytope in $\mathcal{Q}$.

The Attacker's Theory of Mind (ToM) for Offline Multi-Agent Reinforcement Learners

Similar to the theory-of-mind set for normal-form game learners, we define the set for Markov game learners in the $\mathcal{Q}$ space. Here, $\mathcal{D}$ is the set of datasets with K episodes in the form $\left\{\left\{\left(s_h^{(k)}, \mathbf{a}_h^{(k)}, r_h^{(k)}\right)\right\}_{h=1}^{H}\right\}_{k=1}^{K}$ with $s_h^{(k)} \in \mathcal{S}, \mathbf{a}_h^{(k)} \in \mathcal{A}$ and $r_h^{(k)} \in [-b, b]$ for every $k \in [K]$, and the victims compute the Markov perfect equilibria based on the Q functions estimated from such datasets.

Definition 10 (Theory of Mind)**.** *Given a dataset $D \in \mathcal{D}$, the theory-of-mind set $\mathcal{T}(D) \subseteq \mathcal{Q}$ is the set of Q functions that the victims estimate based on D to compute their equilibria. In particular, if the victims learn a policy π, then $\pi \in \bigcup_{Q \in \mathcal{T}(D)} \mathcal{M}(Q)$.*

Example 6 (Theory of Mind for Maximum Likelihood Victims)**.** *To extend Example 1 in the Markov game setting, we define R^{MLE} the same way and P^{MLE} as follows, if $N_h(s,\boldsymbol{a}) := \sum_{k=1}^{K} \mathbb{I}_{\left\{s_h^{(k)}=s, \boldsymbol{a}_h^{(k)}=\boldsymbol{a}\right\}} > 0$,*

$$R_h^{MLE}(s,\boldsymbol{a}|r) := \frac{\sum_{k=1}^{K} r_h^{(k)} \mathbb{I}_{\left\{s_h^{(k)}=s, \boldsymbol{a}_h^{(k)}=\boldsymbol{a}\right\}}}{N_h(s,\boldsymbol{a})} \quad (15)$$

$$P_h^{MLE}(s'|s,\boldsymbol{a}) := \frac{\sum_{k=1}^{K} \mathbb{I}_{\left\{s_{h+1}^{(k)}=s', s_h^{(k)}=s, \boldsymbol{a}_h^{(k)}=\boldsymbol{a}\right\}}}{N_h(s,\boldsymbol{a})} \quad (16)$$

$$P_0^{MLE}(s) := \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}_{\left\{s_1^{(k)}=s\right\}},$$

and if $N_h(s,\boldsymbol{a}) = 0$, we define $R_h^{MLE}(s,\boldsymbol{a}|r) := 0$ and $P_h^{MLE}(s'|s,\boldsymbol{a}) := \frac{1}{|\mathcal{S}|}$.

We can construct Q^{MLE} based on R^{MLE} and P^{MLE} according to (11)*, and since all Nash equilibria have the same value for zero-sum games, Q^{MLE} is unique for every Markov perfect equilibrium of the Markov game with rewards R^{MLE} and transitions P^{MLE}. Then we have that $\mathcal{T}(D)$ is a singleton Q^{MLE}.*

Example 7 (Theory of Mind for Confidence Bound Victims)**.** *Given a dataset $D \in \mathcal{D}$, if the attacker believes the victims estimate the Markov game by estimating the rewards and transitions within some confidence region around some point estimates such as the maximum likelihood estimates, as described in (Wu et al. 2023), then $\mathcal{T}(D)$ would be a polytope with Q functions induced by the Markov games $(\mathcal{S}, \mathcal{A}, P, R, H)$ with P and R satisfying, for every $h \in [H], s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}$,*

$$R_h(s,\boldsymbol{a}|r) \in \mathcal{C}_h^{(R)}(s,\boldsymbol{a}|r) \tag{17}$$
$$\mathcal{C}_h^{(R)}(s,\boldsymbol{a}|r) := \left\{R \in \mathbb{R} : \left|R - \hat{R}_h(s,\boldsymbol{a}|r)\right| \leqslant \rho_h^{(R)}(s,\boldsymbol{a})\right\},$$
$$P_h(s,\boldsymbol{a}) \in \mathcal{C}_h^{(P)}(s,\boldsymbol{a}) \tag{18}$$
$$\mathcal{C}_h^{(P)}(s,\boldsymbol{a}) := \left\{P \in \Delta\mathcal{S} : \left\|P - \hat{P}_h(s,\boldsymbol{a})\right\|_1 \leqslant \rho_h^{(P)}(s,\boldsymbol{a})\right\},$$

for some point estimates $\hat{P}, \hat{R}$, and radii $\rho^{(R)}$ and $\rho^{(P)}$. We note that $\mathcal{T}(D)$ is a polytope in $\mathcal{Q}$, but it has an exponential number of vertices. We can construct a tight hypercube around this polytope and call it the outer approximation of $\mathcal{T}(D)$. It contains all the Q functions in the following set, for every $h \in [H], s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}$,

$$Q_h(s,\boldsymbol{a}|r) \in \left[\underline{Q}_h(s,\boldsymbol{a}|r), \overline{Q}_h(s,\boldsymbol{a}|r)\right], \tag{19}$$
$$\underline{Q}_h(s,\boldsymbol{a}|r) := \min_{R \in \mathcal{C}_h^{(R)}(s,\boldsymbol{a}|r)} R + \min_{P \in \mathcal{C}_h^{(P)}(s,\boldsymbol{a})} \sum_{s' \in \mathcal{S}} P(s') \max_{\pi_1 \in \Delta\mathcal{A}_1} \min_{\pi_2 \in \Delta\mathcal{A}_2} \underline{Q}_{h+1}(s', \pi),$$
$$\overline{Q}_h(s,\boldsymbol{a}|r) := \max_{R \in \mathcal{C}_h^{(R)}(s,\boldsymbol{a}|r)} R + \max_{P \in \mathcal{C}_h^{(P)}(s,\boldsymbol{a})} \sum_{s' \in \mathcal{S}} P(s') \max_{\pi_1 \in \Delta\mathcal{A}_1} \min_{\pi_2 \in \Delta\mathcal{A}_2} \overline{Q}_{h+1}(s', \pi).$$

We omit Example 2 and Example 3 for Markov games since the constructions are identical, except it is done for every stage game. As described in Example 7, we define $\hat{Q}_h(s,\mathbf{a}|r) := \frac{1}{2}\left(\overline{Q}_h(s,\mathbf{a}|r) + \underline{Q}_h(s,\mathbf{a}|r)\right)$ and $\rho_h^{(Q)}(s,\mathbf{a}|r) := \frac{1}{2}\left(\overline{Q}_h(s,\mathbf{a}|r) - \underline{Q}_h(s,\mathbf{a}|r)\right)$, and we formally define the outer approximation of the theory-of-mind set for Markov games as follows.

Definition 11 (Outer Approximation of Theory of Mind)**.** *An outer approximation of $\mathcal{T}(D)$ is a set denoted by $\overline{\mathcal{T}}(D)$ that satisfies $\mathcal{T}(D) \subseteq \overline{\mathcal{T}}(D)$ for every $D \in \mathcal{D}$, and can be written in the form,*

$$\overline{\mathcal{T}}(D) = \left\{Q \in \mathcal{Q} : \left|Q_h(s,\boldsymbol{a}) - \hat{Q}_h(s,\boldsymbol{a}|r)\right| \leqslant \rho_h^{(Q)}(s,\boldsymbol{a}|r), \quad \forall\, \boldsymbol{a} \in \mathcal{A}, h \in [H], s \in \mathcal{S}\right\}, \tag{20}$$

for some point estimate $\hat{Q}$ and radius $\rho^{(Q)}$.
We call $\overline{\mathcal{T}}(D)$ a linear outer approximation if $\hat{Q}$ is linear in $\left\{\left\{r_h^{(k)}\right\}_{h=1}^{H}\right\}_{k=1}^{K}$.

The Cheapest Way to Move ToM into UN for Markov Games

In this subsection, we restate the attacker's problem for multi-agent reinforcement learners.

Definition 12 (Attacker's Problem)**.** *The attacker's problem with target policy $\pi^\dagger$ is,*

$$\inf_{D^\dagger \in \mathcal{D}(D)} C\left(D, D^\dagger\right) \quad s.t.\ \mathcal{T}\left(D^\dagger\right) \subseteq \mathcal{U}\left(\pi^\dagger\right). \tag{21}$$

For reward poisoning problems, we consider the following L_1 cost.

Example 8 (L_1 Cost Function)**.** *For reward poisoning problem, where $\mathcal{D}^{(R)}(D)$ is all possible datasets in the form $D^\dagger = \left\{\left\{\left(s_h^{(k)}, \boldsymbol{a}_h^{(k)}, r_h^{\dagger,(k)}\right)\right\}_{h=1}^{H}\right\}_{k=1}^{K}$ that are modified from $D = \left\{\left\{\left(s_h^{(k)}, \boldsymbol{a}_h^{(k)}, r_h^{(k)}\right)\right\}_{h=1}^{H}\right\}_{k=1}^{K}$, we define the L_1 cost by $C^{(1)}\left(D, D^\dagger\right) = \sum_{k=1}^{K}\sum_{h=1}^{H}\left|r_h^{(k)} - r_h^{\dagger,(k)}\right|$.*

We use the same ι strictness relaxation of the unique Nash set and the linear outer approximation of the theory-of-mind set to convert (21) into a linear program, which can be solved efficiently. We state this observation as the following theorem.

Theorem 2 (Reward Poisoning Linear Program)**.** *Given $\iota > 0$ and a linear $\overline{\mathcal{T}}$, the following problem is a relaxation of the attacker's reward poisoning problem and can be converted into a linear program,*

$$\min_{D^\dagger \in \mathcal{D}^{(R)}(D)} C^{(1)}\left(D, D^\dagger\right) \quad s.t.\ \overline{\mathcal{T}}\left(D^\dagger\right) \subseteq \underline{\mathcal{U}}\left(\pi^\dagger; \iota\right). \tag{22}$$

Example 9 (Maximum Likelihood Centered Linear Program)**.** *In the case $\hat{R} = R^{MLE}$ and $\hat{P} = P^{MLE}$, and we construct $\overline{\mathcal{T}}(D)$ as described in Example 7,* (22) *can be converted into a linear program even without explicitly constructing the $\overline{\mathcal{T}}(D)$ set. We provide an intuition here and the formal construction in the proof of Theorem 2,*

$$\min_{r^\dagger \in [-b,b]^K} \sum_{k=1}^{K}\sum_{h=1}^{H}\left|r_h^{(k)} - r_h^{\dagger,(k)}\right| \tag{23}$$
$$s.t.\ R^{MLE}\left(r^\dagger\right) \text{ is linear in } r^\dagger \text{ satisfying (15)}$$
$$P^{MLE} \text{ is independent of } r^\dagger \text{ satisfying (16)}$$
$$Q^{MLE}\left(r^\dagger\right) \text{ satisfying (11) is linear in } R^{MLE}\left(r^\dagger\right) \text{ thus } r^\dagger$$
$$\overline{Q}\left(r^\dagger\right) \text{ and } \underline{Q}\left(r^\dagger\right) \text{ satisfying (19) are upper and lower bounds of } \overline{\mathcal{T}}\left(r^\dagger\right)$$
$$\left[\overline{Q}\left(r^\dagger\right), \underline{Q}\left(r^\dagger\right)\right] \text{ is in } \underline{\mathcal{U}}\left(\pi^\dagger\right) \text{ satisfying (14)}$$

We move the hypercube $\overline{\mathcal{T}}\left(r^\dagger\right)$ into the polytope $\underline{\mathcal{U}}\left(\pi^\dagger\right)$ by moving one of the corners into the polytope. Note that if $\overline{Q}$

$\mathcal{A}_1 \backslash \mathcal{A}_2$	$1^\dagger$	2	3
$1^\dagger$	0	b	b
2	$-b$	-	-
3	$-b$	-	-

Table 1: A Feasible Attack

$\mathcal{A}_1 \backslash \mathcal{A}_2$	H	T
H	$U[0,1]$	$U[-1,0]$
T	$U[-1,0]$	$U[0,1]$

Table 2: The original dataset generation distributions

and Q are not constructed directly as linear functions of $r^\dagger$, and are computed by (19)*, then these constraints are not linear in $r^\dagger$. We avoid this problem by using the dual linear program of* (19)*. We present the details in the appendix in the proof of Theorem 2. All other constraints are linear in $r^\dagger$, and as a result,* (23) *is a linear program.*

In the end, we present a sufficient but not necessary condition for the feasibility of (22) and (21). This condition applies directly to normal-form games with $H = 1$.

Theorem 3 (Reward Poisoning Linear Program Feasibility)**.** *For $\iota > 0$, $\mathcal{T}(D)$ with $\hat{Q} = Q^{MLE}$, and $N_h(s, \boldsymbol{a}) > 0$ for every $h \in [H], s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}$ where either $a_1 = \pi^\dagger_{1,h}(s)$ or $a_2 = \pi^\dagger_{2,h}(s)$, the attacker's reward poisoning problem is feasible if for every $h \in [H], s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}$,*

$$\rho_h^{(R)}(s, \boldsymbol{a}) \leqslant \frac{b - \iota}{4H}. \tag{24}$$

To construct a feasible attack under (24), we use the poisoned rewards similar to the one shown in Table 1, which is an example where each agent has three actions and the target action profile being action $(1, 1)$. With this $r^\dagger$, the maximum likelihood estimate of the game has a unique Nash equilibrium $\pi^\dagger_h(s)$ with a value of 0 in every stage (h, s). Furthermore, if either the radius of rewards or the radius of Q functions for the theory-of-mind set is less than $\frac{b-\iota}{4H}$, we can show inductively that $\pi^\dagger_h(s)$ remains the unique Nash equilibrium in every stage (h, s), thus showing that every Q function in the theory-of-mind set is also in the unique Nash set, which means the attack is feasible. The complete proof is in the appendix.

4 Experiments

Rock Paper Scissors

We start with a simple toy dataset for the Rock Paper Scissors (RPS) game, shown in Table 3 with partial coverage, where each entry appears once in the dataset, and the target action profile is $\pi^\dagger = (R, R)$, leading to a tie.

	R	P	S
R	0	-1	1
P	1	0	-1
S	-1	1	0

Table 3: RPS Game

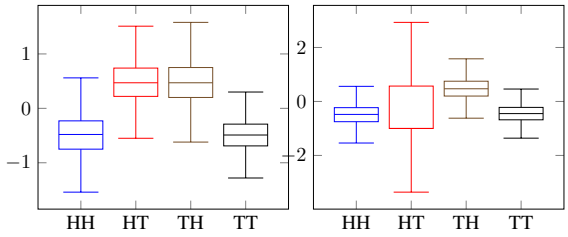
R	P	S
0	-1	1
1	-	-
-1	-	-

Table 4: Original

R	P	S
0	0.01	1
-0.01	-	-
-1	-	-

Table 5: Poisoned

Given the original dataset with 5 entries described in Table 4, our algorithm with $\rho = 0$ and $\iota = 0.01$ leads to the poisoned dataset described in Table 5. The attack cost is 2.02, whereas the attack cost from the feasible attack described in Table 1 with $b = 1$ is 4. In addition, note that given the partial coverage, the attack described in (Wu et al. 2023) is not feasible due to their full coverage requirement.



(a) Original (b) Poisoned

Figure 2: Distribution of rewards

Average costs	$n = 1$	$n = 10$	$n = 100$
Our attack	1.06	9.09	99.47
Feasible attack	2.12	16.08	250.46
DSE attack	2.06	18.31	198.38

Table 6: Cost comparison between different attacks

Stochastic Matching Penny

We follow up with the matching penny game, which is also the penalty kick game in soccer, and the rewards are usually estimated by random data points. We generate the datasets randomly with Uniform distributions summarized in Table 2. The attacker would like to install a target action profile of (H, H), and in the context of the penalty kick game, the attacker's motivation might be to increase or decrease the total number of goals.

We summarize the before-vs-after box plots in Figure 2a for the $n = 100$ case. The cost comparison of our attack, the feasible attack in Table 1 with $b = 1$, and the Dominant Strategy Equilibrium (DSE) attack in (Wu et al. 2023), is given in Table 6.

5 Discussions

We discuss a few extensions. Faking a unique mixed strategy Nash equilibrium is in general impossible due to the sensitivity of mixing probabilities from small perturbations of the reward function, and as long as the theory-of-mind set has non-zero volume, it is impossible to install a mixed strategy profile (or stochastic policy for Markov games) as the unique equilibrium. Faking a unique optimal policy for single-agent reinforcement learners can be easily adapted from our linear program (22). Faking a unique coarse correlated equilibrium in every stage game is equivalent to our problem as well since for a two-player zero-sum game, a policy is the unique Markov perfect coarse correlated equilibrium if and only if it is the unique Markov perfect Nash equilibrium.

6 Acknowledgments

This project is supported in part by NSF grants 1545481, 1704117, 1836978, 2023239, 2041428, 2202457, ARO MURI W911NF2110317, and AF CoE FA9550-18-1-0166. Xie is partially supported by NSF grant 1955997. We also thank Yudong Chen for his useful comments and discussions.

References

Banihashem, K.; Singla, A.; Gan, J.; and Radanovic, G. 2022. Admissible Policy Teaching through Reward Design. *arXiv preprint arXiv:2201.02185*.

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

Brown, N.; and Sandholm, T. 2019. Superhuman AI for multiplayer poker. *Science*, 365(6456): 885–890.

Brown, N.; Sandholm, T.; and Machine, S. 2017. Libratus: The Superhuman AI for No-Limit Poker. In *IJCAI*, 5226–5228.

Cui, Q.; and Du, S. S. 2022. When is Offline Two-Player Zero-Sum Markov Game Solvable? *arXiv preprint arXiv:2201.03522*.

Gleave, A.; Dennis, M.; Wild, C.; Kant, N.; Levine, S.; and Russell, S. 2019. Adversarial policies: Attacking deep reinforcement learning. *arXiv preprint arXiv:1905.10615*.

Gu, S.; Holly, E.; Lillicrap, T.; and Levine, S. 2017. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, 3389–3396. IEEE.

Guo, W.; Wu, X.; Huang, S.; and Xing, X. 2021. Adversarial policy learning in two-player competitive games. In *International Conference on Machine Learning*, 3910–3919. PMLR.

Huang, Y.; and Zhu, Q. 2019. Deceptive reinforcement learning under adversarial manipulations on cost signals. In *International Conference on Decision and Game Theory for Security*, 217–237. Springer.

Jaderberg, M.; Czarnecki, W. M.; Dunning, I.; Marris, L.; Lever, G.; Castaneda, A. G.; Beattie, C.; Rabinowitz, N. C.; Morcos, A. S.; Ruderman, A.; et al. 2019. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science*, 364(6443): 859–865.

Kober, J.; Bagnell, J. A.; and Peters, J. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11): 1238–1274.

Lee, J. W.; and O, J. 2002. A multi-agent Q-learning framework for optimizing stock trading systems. In *International Conference on Database and Expert Systems Applications*, 153–162. Springer.

Lee, J. W.; Park, J.; Jangmin, O.; Lee, J.; and Hong, E. 2007. A multiagent approach to q-learning for daily stock trading. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37(6): 864–877.

Liu, G.; and Lai, L. 2021. Provably Efficient Black-Box Action Poisoning Attacks Against Reinforcement Learning. *Advances in Neural Information Processing Systems*, 34.

Ma, Y.; Wu, Y.; and Zhu, X. 2021. Game Redesign in No-regret Game Playing. *arXiv preprint arXiv:2110.11763*.

Ma, Y.; Zhang, X.; Sun, W.; and Zhu, J. 2019. Policy poisoning in batch reinforcement learning and control. *Advances in Neural Information Processing Systems*, 32: 14570–14580.

Rakhsha, A.; Radanovic, G.; Devidze, R.; Zhu, X.; and Singla, A. 2020. Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning. In *International Conference on Machine Learning*, 7974–7984. PMLR.

Rakhsha, A.; Radanovic, G.; Devidze, R.; Zhu, X.; and Singla, A. 2021a. Policy teaching in reinforcement learning via environment poisoning attacks. *Journal of Machine Learning Research*, 22(210): 1–45.

Rakhsha, A.; Zhang, X.; Zhu, X.; and Singla, A. 2021b. Reward poisoning in reinforcement learning: Attacks against unknown learners in unknown environments. *arXiv preprint arXiv:2102.08492*.

Rangi, A.; Xu, H.; Tran-Thanh, L.; and Franceschetti, M. 2022. Understanding the Limits of Poisoning Attacks in Episodic Reinforcement Learning. In Raedt, L. D., ed., *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 3394–3400. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Riedmiller, M.; Gabel, T.; Hafner, R.; and Lange, S. 2009. Reinforcement learning for robot soccer. *Autonomous Robots*, 27: 55–73.

Shalev-Shwartz, S.; Shammah, S.; and Shashua, A. 2016. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; and Bolton, A. 2017. Mastering the game of go without human knowledge. *nature*, 550(7676): 354–359.

Sun, Y.; Huo, D.; and Huang, F. 2020. Vulnerability-aware poisoning mechanism for online rl with unknown dynamics. *arXiv preprint arXiv:2009.00774*.

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

Wu, Y.; McMahan, J.; Zhu, X.; and Xie, Q. 2023. Reward Poisoning Attacks on Offline Multi-Agent Reinforcement Learning. In *The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI)*.

Yang, Y.; Juntao, L.; and Lingling, P. 2020. Multi-robot path planning based on a deep reinforcement learning DQN algorithm. *CAAI Transactions on Intelligence Technology*, 5(3): 177–183.

Zhang, H.; and Parkes, D. C. 2008. Value-Based Policy Teaching with Active Indirect Elicitation. In *AAAI*, volume 8, 208–214.

Zhang, H.; Parkes, D. C.; and Chen, Y. 2009. Policy teaching through reward function learning. In *Proceedings of the 10th ACM conference on Electronic commerce*, 295–304.

Zhang, X.; Ma, Y.; Singla, A.; and Zhu, X. 2020. Adaptive reward-poisoning attacks against reinforcement learning. In *International Conference on Machine Learning*, 11225–11234. PMLR.