Reward Poisoning Attacks on Offline Multi-Agent Reinforcement Learning

Young Wu
Department of Computer Science
University of Wisconsin-Madison
Madison, WI 53706
yw@cs.wisc.edu

Jeremy McMahan
Department of Computer Science
University of Wisconsin-Madison
Madison, WI 53706
jmcmahan@wisc.edu

Xiaojin Zhu
Department of Computer Science
University of Wisconsin-Madison
Madison, WI 53706
jerryzhu@cs.wisc.edu

Qiaomin Xie
Department of Computer Science
University of Wisconsin-Madison
Madison, WI 53706
qiaomin.xie@wisc.edu

Abstract

We expose the danger of reward poisoning in offline multi-agent reinforcement learning (MARL), whereby an attacker can modify the reward vectors to different learners in an offline data set while incurring a poisoning cost. Based on the poisoned data set, all rational learners using some confidence-bound-based MARL algorithm will infer that a target policy – chosen by the attacker and not necessarily a solution concept originally – is the Markov perfect dominant strategy equilibrium for the underlying Markov Game, hence they will adopt this potentially damaging target policy in the future. We characterize the exact conditions under which the attacker can install a target policy. We further show how the attacker can formulate a linear program to minimize its poisoning cost. Our work shows the need for robust MARL against adversarial attacks.

1 Motivation

In offline Multi-Agent Reinforcement Learning (MARL), a batch data set $\mathcal{D}$ records historical plays of n agents in a Markov Game G under some behavior policies. We consider the episodic setting where each tuple in $\mathcal{D}$ has the form $(s, \boldsymbol{a}, \boldsymbol{r})$, with s being the state of an period in an episode, $\boldsymbol{a} = (a_1, \ldots, a_n)$ the joint action vector of the n learning agents (which is called learners), and $\boldsymbol{r} = (r_1, \ldots, r_n)$ the immediate reward vector for the n learners. $\mathcal{D}$ is the training data shared by n learners. These learners analyze their version of $\mathcal{D}$, and must each independently come up with a rational and strategic policy π_i. They then behave according to the resulting joint policy $\pi = (\pi_1, \ldots, \pi_n)$ in future deployment. For generality, we only ask that the learners be rational in that they will not take dominated actions [21].

Now imagine a threat model where an attacker can modify $\mathcal{D}$ before the learners see it. For instance, the attacker may bribe the historical players to record fake data, or hack the computer where $\mathcal{D}$ is stored. For simplicity, we assume the attacker can only change the reward values $\boldsymbol{r}$ in $\mathcal{D}$. This is known as a reward poisoning attack. The attacker's goal is to convince each learner $i \in [n]$ to follow a specific deterministic policy (pure strategy) $\pi_i^\dagger$ — chosen by the attacker with nefarious intention — by making it desirable. A natural idea is to make $\boldsymbol{\pi}^\dagger = (\pi_1^\dagger, \ldots, \pi_n^\dagger)$ a Nash equilibrium (NE). However, there could be multiple NEs, and additional knowledge on learners' rationality and

Preprint. Under review.

behavior is needed to ensure that $\boldsymbol{\pi}^\dagger$ is the unique policy that will be deployed. On the other hand, not taking a dominated strategy is the most basic rational behavior [21]. To make minimal assumptions on the learners, the attacker intends to make $\boldsymbol{\pi}^\dagger$ an apparent Dominant Strategy Equilibrium (DSE) in each stage of the Markov Game G. The attacker is allowed to choose any deterministic $\boldsymbol{\pi}^\dagger$, not necessarily an equilibrium of the original G. Of course, there will be a poisoning cost to the attacker for modifying $\mathcal{D}$; e.g., bribing takes money. Thus the attacker wants to minimize its poisoning cost while ensuring all agents use $\boldsymbol{\pi}^\dagger$ during deployment.

Poisoning attack has been known in supervised learning and single-agent RL, but not offline MARL. Our main contribution is to demonstrate the feasibility of this threat model until mild conditions. In fact, we show that the attacker can efficiently formulate the attack problem as a linear program (LP). Our results thus call for future studies on MARL defense.

2 Related Work

Online Reward Poisoning: Reward poisoning problem has been studied in various settings, including online single-agent reinforcement learners [2, 10, 15, 23, 24, 22, 25, 33], as well as online bandits [4, 6, 8, 12, 14, 16, 18, 28, 35]. Online reward poisoning for multiple learners is recently studied as a game redesign problem in [19]. We focus on the offline setting in which the attacker modifies the rewards in the training data set so it cannot influence the learners' decisions after their models are trained and deployed.

Offline Reward Poisoning: [20, 22, 23, 29, 30] focus on adversarial attack on offline single-agent reinforcement learners. In our setting, there can be multiple learners and the action of one learner affects the rewards of all other learners. Consequently, the learners cannot make decisions independently, so the results from the single-agent setting do not directly apply here. [7, 9] study the poisoning attack on multi-agent reinforcement learners, assuming that the attacker controls one of the learners. We do not model the attacker as one of the learners, instead, the attacker can poison the rewards of all learners at the same time.

Constrained Mechanism Design: Our paper is also related to the mechanism design literature, in particular, the K-implementation problem in [21, 1]. Our model mainly differs from constrained mechanism design in that the attacker, not a mechanism designer, does not modify the game directly, but instead modifies the training data in a way that the learners infer the underlying game and compute the optimal policy based on the estimated game.

Defense against Reinforcement Learning: There is also recent work on defending reward poisoning or adversarial attacks on reinforcement learning in general, examples including [3, 17, 26, 27, 31, 32]. They focus on defending poisoning attack or data corruption on single-agent reinforcement learning where attackers have limited ability to modify the training data. We are not aware of defenses against reward poisoning in our offline multi-agent reinforcement learning setting.

Robust Multi-Agent Reinforcement Learning: The types of learners our reward poisoning algorithm can successfully attack include naive offline multi-agent learners who compute the optimal policy based on the maximum likelihood estimate of the Markov game, as well as robust offline learners those leverage the principle of pessimism [5, 34].

3 Formal Setting

Markov Game G: A finite-horizon general-sum n-player Markov game is defined as a tuple $G = (\mathcal{S}, \mathcal{A}, P, R, H, \mu)$ [13]. Here $\mathcal{S}$ is the finite state space, and $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ is the finite joint action space. We use $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathcal{A}$ to represent a joint action of the n learners; we sometimes write $\boldsymbol{a} = (a_i, a_{-i})$ to emphasize learner i takes action a_i and the other $n-1$ learners take joint action a_{-i}. For each period $h \in [H-1]$, $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel, where $\Delta(\mathcal{S})$ denotes the probability simplex on $\mathcal{S}$, and $P_h(s'|s, \boldsymbol{a})$ is the probability that the state is s' in period $h+1$ given the state is s and the joint action is $\boldsymbol{a}$ in period h. $\boldsymbol{R}_h : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^n$ is the mean reward function for the n players, where $R_{i,h}(s, \boldsymbol{a})$ denotes the scalar mean reward for player i in state s period h when the joint action $\boldsymbol{a}$ is taken. μ is the initial state distribution in period 1.

We use π to denote a deterministic Markov policy for the game, where $\boldsymbol{\pi}_h : \mathcal{S} \to \mathcal{A}$ is the policy in period h and $\boldsymbol{\pi}_h(s)$ specifies the joint action in state s in period h. We write $\boldsymbol{\pi}_h = (\pi_{i,h}, \pi_{-i,h})$, where $\pi_{i,h}(s)$ is the action taken by learner i and $\pi_{-i,h}(s)$ is the joint action taken by learners other than i in state s period h. The value of a policy π is the expected cumulative reward of the game assuming learners take actions according to π. Formally, the Q value of learner i in state s during period h under joint action $\boldsymbol{a}$ is given recursively by

$$Q_{i,H}^{\pi}\left(s,\boldsymbol{a}\right)=R_{i,H}\left(s,\boldsymbol{a}\right);\; Q_{i,h}^{\pi}\left(s,\boldsymbol{a}\right)=R_{i,h}\left(s,\boldsymbol{a}\right)+\sum_{s'\in\mathcal{S}}P_h\left(s'|s,\boldsymbol{a}\right)V_{i,h+1}^{\pi}\left(s'\right),h\in[H-1].$$

The value to learner i in state s in period h under policy π is given by $V_{i,h}^{\pi}\left(s\right)=Q_{i,h}^{\pi}\left(s,\boldsymbol{\pi}_h\left(s\right)\right)$, and we use $\boldsymbol{V}_h^{\pi}\left(s\right)$ to denote the vector of values for all learners in state s in period h under policy π.

Data set $\mathcal{D}$: The offline data set $\mathcal{D}=\left\{\left(s_h^{(k)},\boldsymbol{a}_h^{(k)},\boldsymbol{r}_h^{0,(k)}\right)_{h=1}^{H}\right\}_{k=1}^{K}$ contains K episodes of length H. The data tuple in episode k period h consists of the state $s_h^{(k)}\in\mathcal{S}$, action profile $\boldsymbol{a}_h^{(k)}\in\mathcal{A}$, and reward vector $\boldsymbol{r}_h^{0,(k)}\in\mathbb{R}^n$ where the superscript 0 denotes original rewards before attack. The next state $s_{h+1}^{(k)}$ is in the next tuple. We make no assumption on the behavior policies that generated $\mathcal{D}$ other than the full coverage condition below. This condition is needed to allow the learners to verify the fake DSE. [1] Let $N_h\left(s,\boldsymbol{a}\right)=\sum_{k=1}^{K}\mathbb{1}_{\{s_h^{(k)}=s,\boldsymbol{a}_h^{(k)}=\boldsymbol{a}\}}$ be the count of $(s,\boldsymbol{a},\cdot)$ in period h.

Assumption 1 (Full Coverage)**.** Every $(s,\boldsymbol{a})$ pair appears at least once in every period h, i.e., $N_h\left(s,\boldsymbol{a}\right)>0,\forall\, h\in[H],s\in\mathcal{S},a\in\mathcal{A}$.

The Learners: The learners are uncertainty-aware. Given training set $\mathcal{D}$, they estimate high confidence bounds on what Markov games (P and R) are plausible. Conversely, the attacker must ensure its attacks work on any one of these plausible Markov games. This makes the attacks strong; in fact one can show such attacks work on state-of-the-art learners such as pessimistic offline MARL algorithms [5, 34]; see Appendix A for details.

Assumption 2. The learners' confidence set on transition $P_h\left(s,\boldsymbol{a}\right)$ has the form

$$\mathrm{CI}_h^P\left(s,\boldsymbol{a}\right):=\left\{P_h\left(s,\boldsymbol{a}\right)\in\Delta\left(\mathcal{A}\right):\left\|P_h\left(s,\boldsymbol{a}\right)-\hat{P}_h\left(s,\boldsymbol{a}\right)\right\|_1\leq\rho_h^P\left(s,\boldsymbol{a}\right)\right\},\tag{1}$$

where $\hat{P}_h\left(s'|s,\boldsymbol{a}\right):=\dfrac{1}{N_h\left(s,\boldsymbol{a}\right)}\sum_{k=1}^{K}\mathbb{1}_{\left\{s_{h+1}^{(k)}=s',s_h^{(k)}=s,\boldsymbol{a}_h^{(k)}=\boldsymbol{a}\right\}}$ is the maximum likelihood estimate (MLE) of the true transition probability. Similarly, the learners' confidence set on reward $R_{i,h}\left(s,\boldsymbol{a}\right)$ has the form

$$\mathrm{CI}_{i,h}^R\left(s,\boldsymbol{a}\right):=\left\{R_{i,h}\left(s,\boldsymbol{a}\right)\in[-b,b]:\left|R_{i,h}\left(s,\boldsymbol{a}\right)-\hat{R}_{i,h}\left(s,\boldsymbol{a}\right)\right|\leq\rho_h^R\left(s,\boldsymbol{a}\right)\right\},\tag{2}$$

where $\hat{R}_{i,h}\left(s,\boldsymbol{a}\right):=\dfrac{1}{N_h\left(s,\boldsymbol{a}\right)}\sum_{k=0}^{K}r_{i,h}^{0,(k)}\mathbb{1}_{\left\{s_h^{(k)}=s,\boldsymbol{a}_h^{(k)}=\boldsymbol{a}\right\}}$ is the MLE of the reward.

Note that both the attacker and the learners know that all rewards are bounded within $[-b,b]$ (we allow $b=\infty$). The values of $\rho_h^P\left(s,\boldsymbol{a}\right)$ and $\rho_h^R\left(s,\boldsymbol{a}\right)$ are typically given by concentration inequalities.

We denote all plausible Markov games consistent with $\mathcal{D}$ by CI^G:

$$\mathrm{CI}^G:=\left\{G=\left(\mathcal{S},\mathcal{A},P,R,H,\mu\right):P_h\left(s,\boldsymbol{a}\right)\in\mathrm{CI}_h^P\left(s,\boldsymbol{a}\right),R_{i,h}\left(s,\boldsymbol{a}\right)\in\mathrm{CI}_{i,h}^R\left(s,\boldsymbol{a}\right),\forall\, i,h,s,\boldsymbol{a}\right\}.$$

As stated earlier, the attacker makes minimal assumptions on learners rationality, namely they do not take dominated actions. For technical reasons, we strengthen this assumption slightly by introducing a small margin $\iota>0$ (representing the learners' numerical resolution).

Definition 1. A ι-strict Markov perfect dominant strategy equilibrium (MPDSE) of a Markov game G is a policy π satisfying that for all learners $i\in[n]$, periods $h\in[H]$, and states $s\in\mathcal{S}$,

$$Q_{i,h}^{\pi}\left(s,\left(\pi_{i,h}(s),a_{-i}\right)\right)\geq Q_{i,h}^{\pi}\left(s,\left(a_i,a_{-i}\right)\right)+\iota,\quad\forall\, a_i\in\mathcal{A}_i,a_i\neq\pi_{i,h}(s),a_{-i}\in\mathcal{A}_{-i}.\tag{3}$$

Assumption 3. The learners will play an ι-strict MPDSE should one exist.

[1] The coverage condition can be relaxed to unilateral concentration [5] or low relative uncertainty [34], *if* the attacker makes much stronger assumption on learners rationality such that they will agree on a unique NE.

The Attacker: The attacker has access to the original data $\mathcal{D}$ and knows the learners' $\rho_h^P(s, \boldsymbol{a})$ and $\rho_h^R(s, \boldsymbol{a})$ parameters, but not their exact learning algorithm. Nonetheless, the attacker is assured that the learners' computed policies are consistent with some game in CI^G. The attacker is allowed to modify the rewards in $\mathcal{D}$ before the learners observe $\mathcal{D}$. The attacker has a pre-specified target policy $\boldsymbol{\pi}^\dagger$ that it wants the learners to learn. The attacker wants to poison $\mathcal{D}$ to ensure $\boldsymbol{\pi}^\dagger$ is the unique ι-strict MPDSE in all of CI^G while minimizing its poisoning cost. Let the attacker's poison cost function be $C\left(r^0, r^\dagger\right)$, where $r^0 = \left\{\left(\boldsymbol{r}_h^{0,(k)}\right)_{h=1}^H\right\}_{k=1}^K$ are the original rewards and $r^\dagger = \left\{\left(\boldsymbol{r}_h^{\dagger,(k)}\right)_{h=1}^H\right\}_{k=1}^K$ are the poisoned rewards. We will mostly use the L^1-norm cost $C\left(r^0, r^\dagger\right) = \|r^0 - r^\dagger\|_1$.

4 Attack Formulation

4.1 Special Case: Bandit Games (H = |S| = 1)

To fix idea, we start with the special case with $|\mathcal{S}| = 1$ and $H = 1$, sometimes called a bandit game, which has only a single stage normal-form game.

Just for illustration, in this paragraph let us even pretend that the learners simply take an MLE point estimate $\hat{G}$ of the game from data. This is unrealistic, but it highlights the attacker's tactic to convince the learners that $\boldsymbol{\pi}^\dagger$ is a unique ι-strict DSE in $\hat{G}$. Concretely, the original data is $\mathcal{D} := \left\{\left(\boldsymbol{a}^{(k)}, \boldsymbol{r}^{0,(k)}\right)\right\}_{k=1}^K$ (recall we no longer have state/period). Define counts $N(\boldsymbol{a}) := \sum_{k=1}^K \mathbb{1}_{\left\{\boldsymbol{a}^{(k)}=\boldsymbol{a}\right\}}$. The attacker's problem can be formulated as an optimization problem given in (4)–(7).

$$\min_{r^\dagger} C\left(r^0, r^\dagger\right) \tag{4}$$

$$\text{s.t. } R^\dagger(\boldsymbol{a}) := \frac{1}{N(\boldsymbol{a})} \sum_{k=1}^K \boldsymbol{r}^{\dagger,(k)} \mathbb{1}_{\left\{\boldsymbol{a}^{(k)}=\boldsymbol{a}\right\}}, \quad \forall\, \boldsymbol{a}; \tag{5}$$

$$R_i^\dagger\left(\pi_i^\dagger, a_{-i}\right) \geq R_i^\dagger\left(a_i, a_{-i}\right) + \iota, \quad \forall\, i, a_{-i}, a_i \neq \pi_i^\dagger; \tag{6}$$

$$\boldsymbol{r}^{\dagger,(k)} \in [-b, b]^n, \forall\, k. \tag{7}$$

$$\min_{r^\dagger} C(r^0, r^\dagger) \tag{8}$$

$$\text{s.t. } R^\dagger(\boldsymbol{a}) := \frac{1}{N(a)} \sum_{k=1}^K \boldsymbol{r}^{\dagger,(k)} \mathbb{1}_{\left\{\boldsymbol{a}^{(k)}=a\right\}}, \forall\, \boldsymbol{a}; \tag{9}$$

$$\mathrm{CI}_i^{R^\dagger}(\boldsymbol{a}) := \left\{R_i(\boldsymbol{a}) \in [-b, b] : \left|R_i(\boldsymbol{a}) - R_i^\dagger(\boldsymbol{a})\right| \leq \rho_h^R(\boldsymbol{a})\right\}, \forall\, i, \boldsymbol{a}; \tag{10}$$

$$\min_{R_i \in \mathrm{CI}_i^{R^\dagger}\left(\pi_i^\dagger, a_{-i}\right)} R_i \geq \max_{R_i \in \mathrm{CI}_i^{R^\dagger}\left(a_i, a_{-i}\right)} R_i + \iota, \quad \forall\, i, a_{-i}, a_i \neq \pi_i^\dagger; \tag{11}$$

$$\boldsymbol{r}^{\dagger,(k)} \in [-b, b]^n, \forall\, k. \tag{12}$$

The first constraint simulates the learners MLE $\hat{G}$ after poisoning; the second constraint enforces ι-strict DSE. We observe that: (i) The problem is feasible if $\iota \leq 2b$ since the attacker can always ensure the reward for the target action is b and the reward for all other actions is $-b$; (ii) For L^1-norm poison cost the problem is a linear program with nK variables and $(A-1)A^{n-1} + 2nK$ inequality constraints, assuming each learner has $|\mathcal{A}_i| = A$ actions; (iii) After attack, learner i only needs to see its own rewards to be convinced that $\pi_i^\dagger$ is a dominant strategy—it does not even need to observe other learners' rewards. Also note that this simple formulation serves as an asymptotic approximation of the attack problem for confidence bound learners—when $N(\boldsymbol{a})$ is large for all $\boldsymbol{a}$, the confidence intervals on P and R are usually small.

We now revert back to the more realistic learners who maintains an uncertainty game set CI^G from data, but still with $|\mathcal{S}| = 1, H = 1$. In this case, the attacker enforces an ι separation between the lower bound of the target action's reward and the upper bounds of the rewards from all other actions (similar to arm elimination in bandits). As a result, all plausible games in CI^G have the target action profile as the dominant strategy equilibrium. The attacker's optimization problem is only slightly more complex, as given in (8)–(12).

Proposition 1. *The attacker's problem* (8)–(12) *is feasible if* $\iota \leq 2b - 2\rho^R(\boldsymbol{a}), \forall\, \boldsymbol{a} \in \mathcal{A}$.

We present the proof in the Appendix C.1 as a special case of Theorem 1 when $H = |\mathcal{S}| = 1$.

When an L^1-norm cost function $C(\cdot,\cdot)$ is used, it is clear that (4)–(7) and (8)–(12) are linear programming problems, see Appendix B.1 and Appendix B.2.

Proposition 2. *With L^1-norm cost function $C(\cdot,\cdot)$, the problem* (8)–(12) *can be formulated as an linear program.*

4.2 General Case: Markov Games

We now present poisoning attack on a general Markov game. With multiple states and periods, there are two complications: (i) In period h the learners' decision depends on Q_h, which involves both the future Q_{h+1} value and the immediate reward R_h; (ii) The learners' uncertainty in Q_h also enlarges as it propagates backward in h. Accordingly, the attacker needs to design the poison recursively.

Our main technical innovation is a "Q confidence-bound backward induction" attack formulation. The attacker maintains confidence upper and lower bounds on the learners' Q function, $\overline{Q}$ and $\underline{Q}$, with backward induction. To install an ι-strict MPDSE target policy $\boldsymbol{\pi}^\dagger$, the attacker ensures ι separation between lower bound of the target action and upper bound of all other actions, at all states/periods.

Recall Assumption 2 that given the training data $\mathcal{D}$, the learners would compute the MLEs $\boldsymbol{R}_h$ and corresponding confidence sets $\mathrm{CI}_{i,h}^{R}$ for the reward. The attacker aims to poison $\mathcal{D}$ so that these MLEs and confident sets become $\boldsymbol{R}_h^\dagger$ and $\mathrm{CI}_{i,h}^{R^\dagger}$, under which $\boldsymbol{\pi}^\dagger$ is the unique ι-strict MPDSE for all plausible games in the corresponding confidence game set. The attacker finds the minimum cost way of doing so by solving the following Q confidence-bound backward induction optimization problem:

$$\min_{r^\dagger} C\left(r^0, r^\dagger\right) \tag{13}$$

$$\text{s.t. } R_{i,h}^\dagger(s,\boldsymbol{a}) := \frac{1}{N_h(s,\boldsymbol{a})}\sum_{k=0}^{K} r_{i,h}^{\dagger,(k)} \mathbb{1}_{\left\{s_h^{(k)}=s,\boldsymbol{a}_h^{(k)}=\boldsymbol{a}\right\}}, \forall\, h,s,i,\boldsymbol{a} \tag{14}$$

$$\mathrm{CI}_{i,h}^{R^\dagger}(s,\boldsymbol{a}) := \left\{R_{i,h}(s,\boldsymbol{a}) \in [-b,b] : \left|R_{i,h}(s,\boldsymbol{a}) - R_{i,h}^\dagger(s,\boldsymbol{a})\right| \le \rho_h^R(s,\boldsymbol{a})\right\}, \forall\, h,s,i,\boldsymbol{a} \tag{15}$$

$$\underline{Q}_{i,H}(s,\boldsymbol{a}) := \min_{R_{i,H}\in \mathrm{CI}_{i,H}^{R^\dagger}(s,\boldsymbol{a})} R_{i,H}, \forall\, s,i,\boldsymbol{a} \tag{16}$$

$$\underline{Q}_{i,h}(s,\boldsymbol{a}) := \min_{R_{i,h}\in \mathrm{CI}_{i,h}^{R^\dagger}(s,\boldsymbol{a})} R_{i,h} + \min_{P_h\in \mathrm{CI}_h^P(s,\boldsymbol{a})} \sum_{s'\in\mathcal{S}} P_h(s')\, \underline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^\dagger(s')\right), \forall\, h<H,s,i,\boldsymbol{a} \tag{17}$$

$$\overline{Q}_{i,H}(s,\boldsymbol{a}) := \max_{R_{i,H}\in \mathrm{CI}_{i,H}^{R^\dagger}(s,\boldsymbol{a})} R_{i,H}, \forall\, s,i,\boldsymbol{a} \tag{18}$$

$$\overline{Q}_{i,h}(s,\boldsymbol{a}) := \max_{R_{i,h}\in \mathrm{CI}_{i,h}^{R^\dagger}(s,\boldsymbol{a})} R_{i,h} + \max_{P_h\in \mathrm{CI}_h^P(s,\boldsymbol{a})} \sum_{s'\in\mathcal{S}} P_h(s')\, \overline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^\dagger(s')\right), \forall\, h<H,s,i,\boldsymbol{a} \tag{19}$$

$$\underline{Q}_{i,h}\left(s,\left(\pi_{i,h}^\dagger(s), a_{-i}\right)\right) \ge \overline{Q}_{i,h}\left(s,(a_i,a_{-i})\right) + \iota, \forall\, h,s,i,a_{-i}, a_i \ne \pi_{i,h}^\dagger(s) \tag{20}$$

$$\boldsymbol{r}_h^{\dagger,(k)} \in [-b,b]^n, \forall\, h,k. \tag{21}$$

The backward induction steps (17) and (19) ensure that $\underline{Q}$ and $\overline{Q}$ are valid lower and upper bounds for the Q function for all plausible Markov Games in CI^G, for all periods. The margin constraints (20) enforces ι-separation between the target action and other actions at all states and periods. The following lemma, proved in Appendix C.1, shows that the attack is successful in the sense that all plausible games in CI^G will have $\boldsymbol{\pi}^\dagger$ as the unique ι-strict MPDSE.

Lemma 1. *After the poison attack, $\boldsymbol{\pi}^\dagger$ is the unique ι-strict MPDSE of every Markov game $G \in \mathrm{CI}^G$.*

The following theorem shows that the above optimization problem is indeed feasible under a mild condition. The proof is given in Appendix C.1.

Theorem 1. *The attacker's problem* (13)–(21) *is feasible if the following condition holds:*

$$\iota \leq 2b - (H+1)\,\rho_h^R(s, \boldsymbol{a}), \quad \forall\, h \in [H], s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}. \tag{22}$$

Despite the inner min and max, the problem (13)–(21) can be formulated as an LP, thanks to LP duality. We again have the following LP representability result, proved in Appendix B.2.

Theorem 2. *With L^1-norm cost function $C(\cdot,\cdot)$, problem* (13)–(21) *can be formulated as an LP.*

5 Attack Cost Lower Bound

Up to this point, things have gone very well for the attacker. Not only can the attacker always make the target policy $\boldsymbol{\pi}^\dagger$ a Markov perfect dominant strategy equilibrium under mild conditions, but the attacker can conduct the poisoning efficiently using an LP formulation. However, forcing $\boldsymbol{\pi}^\dagger$ is not the only goal of the attacker. The attacker also desires a small attack cost. Unfortunately for the attacker, this is not always possible. We show an attack cost lower bound, with the implication that some games require high costs to poison.

To simplify notation, we first define ι-dominance gaps.

Definition 2. For every $h \in [H], s \in \mathcal{S}, i \in [n]$ and $a_{-i} \in \mathcal{A}_{-i}$, the ι-dominance gap is,

$$\begin{aligned} d_{i,h}^\iota(s, a_{-i}) := \max\Big\{0, \max_{a_i \neq \pi_{i,h}^\dagger(s)} \Big[\hat{R}_{i,h}\big(s, (a_i, a_{-i})\big) + \rho_h^R\big(s, (a_i, a_{-i})\big)\Big] \\ - \hat{R}_{i,h}\Big(s, \big(\pi_{i,h}^\dagger(s), a_{-i}\big)\Big) + \rho_h^R\Big(s, \big(\pi_{i,h}^\dagger(s), a_{-i}\big)\Big) + \iota\Big\}, \end{aligned} \tag{23}$$

where $\hat{R}$ is the MLE w.r.t. the original data set $\mathcal{D}$.

Intuitively, this is how much the attacker would have to increase the mean reward for learner i while others are playing a_{-i}, so that the action $\pi_{i,h}^\dagger(s)$ becomes the ι-dominant strategy for learner i. In addition, we define the minimum count for any period h in $\mathcal{D}$ as $\underline{N}_h := \min_{s \in \mathcal{S}} \min_{\boldsymbol{a} \in \mathcal{A}} N_h(s, \boldsymbol{a})$, and the minimal over all periods as $\underline{N} := \min_{h \in H} \underline{N}_h$. With these definitions in hand, we present a lower bound on the attack cost given a specific full coverage data set $\mathcal{D}$.

Lemma 2. *For any dataset $\mathcal{D}$ with full coverage, and for any $\iota > 0$, the optimal attack cost with respect to L^1-norm is at least $\underline{N}_H \sum_{s \in \mathcal{S}} \sum_{i=1}^n \sum_{a_{-i} \in \mathcal{A}_{-i}} d_{i,H}^\iota(s, a_{-i})$.*

We provide the proof in Appendix C.2. This lower bound is valid for all Markov games, but it is weak in that it only uses the last period cost (it is exact for bandit games with $b = \infty$). However, this is also the most general lower bound we can obtain without assuming more about the structure of the game.

We show next if we assume some structure on the dataset, then this lower bound can be extended beyond the last period showing that the attack must have higher cost.

Lemma 3. *For any dataset $\mathcal{D}$ with full coverage and uniform MLE transitions (i.e., $\hat{P}_h(s'|s, \boldsymbol{a}) = 1/|\mathcal{S}|, \forall h, s', s, \boldsymbol{a}$), and for any $\iota > 0$, the optimal attack cost with respect to L^1-norm is at least $\sum_{h=1}^H \underline{N}_h \sum_{s \in \mathcal{S}} \sum_{i=1}^n \sum_{a_{-i} \in \mathcal{A}_{-i}} d_{i,h}^\iota(s, a_{-i})$.*

The proof can be found in Appendix C.2.

The lower bounds in both Lemmas 2 and 3 expose an exponential dependency on n, the number of players, for some extreme data sets $\mathcal{D}$, for which the attacker needs to modify $\hat{R}_{i,h}(s, \boldsymbol{a})$ for essentially every $\boldsymbol{a} \in \mathcal{A}$. The following dataset illustrates this dependency.

Consider an original data set $\mathcal{D}$ with uniform MLE transitions and the MLE of the rewards is given by $\hat{R}_{i,h}(s, a) = -b$ if $a_i = \pi_{i,h}^\dagger(s)$ and $\hat{R}_{i,h}(s, a) = b$ otherwise, for all $s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}$. The original rewards are the opposite of what the attacker wants. For example, with two learners and $\pi_{i,h}^\dagger(s) = (1, 1)$, the MLE $\hat{\boldsymbol{R}}_h(s, \cdot)$ from the original $\mathcal{D}$ forms the reward matrix in Table 1.

$\mathcal{A}_1/\mathcal{A}_2$	1	2	...	$\lvert\mathcal{A}_2\rvert$
1	$-b,-b$	$-b,b$	...	$-b,b$
2	$b,-b$	b,b	...	b,b
...	...	...	...	...
$\lvert\mathcal{A}_1\rvert$	$b,-b$	b,b	...	b,b

Table 1: MLE $\hat{\boldsymbol{R}}_h(s,\cdot)$ before attack

$\mathcal{A}_1/\mathcal{A}_2$	1	2	...	$\lvert\mathcal{A}_2\rvert$
1	b,b	$b,b-2\rho-\iota$	...	$b,b-2\rho-\iota$
2	$b-2\rho-\iota,b$	$b-2\rho-\iota,b-2\rho-\iota$	...	$b-2\rho-\iota,b-2\rho-\iota$
...	...	...	...	...
$\lvert\mathcal{A}_1\rvert$	$b-2\rho-\iota,b$	$b-2\rho-\iota,b-2\rho-\iota$	...	$b-2\rho-\iota,b-2\rho-\iota$

Table 2: MLE $\hat{\boldsymbol{R}}_h(s,\cdot)$ after attack

For simplicity suppose the learners use a unique confidence parameter $\rho_H^R(s,\boldsymbol{a})=\rho$ for all $s,\boldsymbol{a}$. The attacker needs to flip all reward matrices in period H to be at least as shown in Table 2. The situation is the same for $n\geq 2$ learners. For this data set the ι-dominance gap is $d_{i,H}^{\iota}(s,a_{-i})=2b+2\rho+\iota$. A direct application of Lemma 3 gives the foolowing more transparent lower bound:

Theorem 3. *Suppose $|\mathcal{A}_i|=|\mathcal{A}_1|$ for all $i\in[n]$. There exists a data set $\mathcal{D}$ with full coverage, such that the optimal attack cost with respect to L^1-norm cost is at least $H\,|\mathcal{S}|\,\underline{N}n|\mathcal{A}_1|^{n-1}(2b+2\rho+\iota)$.*

Recall the attacker wants to assume little about the learners, and therefore chooses to install a MPDSE (instead of making stronger assumptions on the learners and installing a Nash equilibrium or a non-Markov perfect equilibrium). On some data sets $\mathcal{D}$, the exponential poison cost is the price the attacker pays for this flexibility.

6 A Greedy Backward-Induction Attack Algorithm

While the general attack formulation (13) can be converted to a linear program, the problem may be too large to solve as a single linear program. If all learners have the same number of actions $|\mathcal{A}_i|=A$, then there are $2nKH+nH\,|\mathcal{S}|^2\,A^n$ variables and $4nKH+6nH\,|\mathcal{S}|^2\,A^n+nH\,|\mathcal{S}|\,(A-1)\,A^n$ constraints. Note that the attack problem is naturally separable over learners, but usually not separable over states and periods as the $\underline{Q},\overline{Q}$ variables couple the stage games.

One heuristic to speed up the optimization is to intentionally decouple them, so that $\underline{Q}_{i,h+1},\overline{Q}_{i,h+1}$ are computed and fixed in period $h+1$, then act as constants in period h. Algorithm 1 is such a greedy backward induction attack algorithm that breaks down (13) into $nH\,|\mathcal{S}|$ smaller linear programs, one for each learner i in one stage game for a specific period h and a specific state s. As a result, each small linear program has less than $2K+|\mathcal{S}|\,A^n$ variables and fewer than $4K+6\,|\mathcal{S}|\,A^n+(A-1)\,A^n$ constraints. More importantly, the constraint matrix is relatively sparse due to the removal of the interactions across the periods and states, and thus significantly easier to solve. The disadvantage is that $\underline{Q},\overline{Q}$ are no longer jointly optimized over all periods, but only stage-wise (hence greedy). This still results in a valid attack, albeit with higher poison cost.

Algorithm 1 Greedy Backward-Induction Attack

Require: The data set $\mathcal{D}$, the target policy $\boldsymbol{\pi}^{\dagger}$, and attack parameters including C,ι,b,ρ^R,ρ^P.
Ensure: $0\leq\iota\leq 2b-(H+1)\,\rho_h^R(s,\boldsymbol{a})\,,\forall\,h\in[H]\,,s\in\mathcal{S},\boldsymbol{a}\in\mathcal{A}$.
  for $i=1,2,\ldots,n$ **do**
    for $h=H,H-1,\ldots,1$ **do**
      for $s\in\mathcal{S}$ **do**
        Solve attack subproblem (24)- (30). Store variables $\underline{Q}_{i,h}(s,\boldsymbol{a})$, $\overline{Q}_{i,h}(s,\boldsymbol{a})$ for each $\boldsymbol{a}\in\mathcal{A}$.

Since the subproblem only involves a subset of the data, we use the shorthand notation $r_{i,h}^{0,(k)}(s)$ to denote the set of rewards for learner i in the training items $\left(s_h^{(k)},\boldsymbol{a}_h^{(k)},\boldsymbol{r}_h^{0,(k)}\right)$ with $s_h^{(k)}=s$. In addition, we assume the cost function is separable in the sense $C\left(r^{\dagger},r^0\right)=\sum_{k=1}^{K}\sum_{i=1}^{n}\sum_{h=1}^{H}\sum_{s\in\mathcal{S}}C_{i,h,(s)}^{(k)}\left(r_{i,h}^{\dagger,(k)}(s)\,,r_{i,h}^{0,(k)}(s)\right)$. Now we define the attack subproblem in period $h\in[H]$ and state $s\in\mathcal{S}$ in Algorithm 1:

$$\min_{r_{i,h}^{0,(k)}(s)}\sum_{k=1}^{K}C_{i,h,(s)}^{(k)}\left(r_{i,h}^{\dagger,(k)}(s)\,,r_{i,h}^{0,(k)}(s)\right)\tag{24}$$

$$\text{s.t. } R^{\dagger}_{i,h}(s,\boldsymbol{a}) := \frac{1}{N_h(s,\boldsymbol{a})}\sum_{k=1}^{K} r^{\dagger,(k)}_{i,h}\mathbb{1}_{\left\{s^{(k)}_h = s, \boldsymbol{a}^{(k)}_h = \boldsymbol{a}\right\}}, \forall\, \boldsymbol{a} \tag{25}$$

$$\text{CI}^{R^{\dagger}}_{i,h}(s,\boldsymbol{a}) := \left\{R_{i,h}(s,\boldsymbol{a}) \in [-b,b] : \left|R_{i,h}(s,\boldsymbol{a}) = R^{\dagger}_{i,h}(s,\boldsymbol{a})\right| \le \rho^R_h(s,\boldsymbol{a})\right\}, \forall\, \boldsymbol{a} \tag{26}$$

$$\underline{Q}_{i,h}(s,\boldsymbol{a}) := \min_{R_{i,h}\in \text{CI}^{R^{\dagger}}_{i,h}(s,\boldsymbol{a})} R_{i,h} + \begin{cases} 0 & \text{if } h = H \\ \min\limits_{P_h \in \text{CI}^P_h(s,\boldsymbol{a})} P_h(s')\,\underline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}^{\dagger}_{h+1}(s')\right) & \text{if } h < H \end{cases}, \forall\, \boldsymbol{a} \tag{27}$$

$$\overline{Q}_{i,h}(s,\boldsymbol{a}) := \max_{R_{i,h}\in \text{CI}^{R^{\dagger}}_{i,h}(s,\boldsymbol{a})} R_{i,h} + \begin{cases} 0 & \text{if } h = H \\ \max\limits_{P_h \in \text{CI}^P_h(s,\boldsymbol{a})} P_h(s')\,\overline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}^{\dagger}_{h+1}(s')\right) & \text{if } h < H \end{cases}, \forall\, \boldsymbol{a} \tag{28}$$

$$\underline{Q}_{i,h}\left(s, \left(\pi^{\dagger}_{i,h}(s), a_{-i}\right)\right) \ge \overline{Q}_{i,h}(s,(a_i,a_{-i})) + \iota, \forall\, a_{-i}, a_i \ne \pi^{\dagger}_{i,h}(s) \tag{29}$$

$$r^{\dagger,(k)}_{i,h} \in [-b,b], \forall\, k \tag{30}$$

The key difference between this subproblem and the original attack problem (13) is that $\underline{Q}_{i,h+1}(s,\boldsymbol{a})$ and $\overline{Q}_{i,h+1}(s,\boldsymbol{a})$ are constants; they were computed and stored by the $h+1$ subproblems and used in the h subproblems. Conversely, the current subproblem computes and stores $\underline{Q}_{i,h}(s,\boldsymbol{a})$ by (27) and $\overline{Q}_{i,h}(s,\boldsymbol{a})$ by (28) for period $h-1$. The subproblem can be converted to a linear program and the derivation is presented in Appendix B.

Lemma 4. *The solution to the fast backward-induction attack described by Algorithm 1 is feasible in the attacker's problem* (13)–(21).

Although the greedy algorithm retains feasibility, the simpler implementation and faster computation time come at a price. Specifically, the attack cost of the greedy algorithm need not be optimal.

Lemma 5. *There exists a full coverage data set $\mathcal{D}^1$ on which the solution to the fast backward-induction attack described by Algorithm 1 is not cost-optimal in the attacker's problem* (13)–(21).

We present the proof of Lemma 4 and provide an example of $\mathcal{D}^1$ in Lemma 5 in the Appendix C.3.

7 Experiments

7.1 Biased Prisoner's Dilemma

Suppose Bonnie and Clyde are in a Prisoner's Dilemma, in which they can either defect (D) or confess (C). The NE is for both players to defect. Suppose the attacker is a relative of Bonnie who has an inside man at the police department. When the police department makes offers to criminals, they allow them to see past offers with their outcomes. The attacker can influence both criminals by paying off the inside man to modify the past offer data. The attacker wants to ensure Bonnie defects and Clyde confesses; thus, completely saving Bonnie. However, Bonnie and Clyde inevitably go back to crime after Clyde is released. Once they get caught again, another prisoner's dilemma arises, but this time Clyde is more suspicious of confessing and Bonnie is more prone to confessing due to guilt. We assume this process happens only twice and then both give up crime forever.

We describe this scenario as a 2-period Markov Game. Let $\mathcal{S} = \{0, +1, -1\}$, where state $+1$ denotes Bonnie defected and Clyde confessed previously, -1 denotes the opposite situation, and 0 denotes they both took the same action previously so remain trusting. The game starts in state 0 since they trust each other initially. The transitions, denoted $\hat{P}(a)$, are deterministic and defined to be consistent with the definitions of the states above. The mean rewards of the game are,

$\hat{R}_h(0,\cdot)$

$\mathcal{A}_1/\mathcal{A}_2$	C	D
C	$-5,-5$	$-10,0$
D	$0,-10$	$-1,-1$

$\hat{R}_1(+1,\cdot)$

$\mathcal{A}_1/\mathcal{A}_2$	C	D
C	$-4,-6$	$-11,1$
D	$1,-11$	$0,-2$

$\hat{R}_1(-1,\cdot)$

$\mathcal{A}_1/\mathcal{A}_2$	C	D
C	$-6,-4$	$1,-11$
D	$-11,1$	$-2,0$

So we see it is harder to convince Clyde to confess the second time and harder to make Bonnie defect. The target is $\pi_h^\dagger(s) = (D, C)$ for all states and time periods; we have highlighted this cell for clarity in all reward matrices. If the attack is successful then the second period state will always be $+1$. We use $K = 16$ episodes. Each episode is of the form $\{(0, \boldsymbol{a}, \hat{R}_0(0, \boldsymbol{a})), (\hat{P}(a), \boldsymbol{a}', \hat{R}_1(\hat{P}(a), \boldsymbol{a}')\}$ for some joint actions $\boldsymbol{a}$ and $\boldsymbol{a}'$. We set $\rho_h^R(s, \boldsymbol{a}) = .1$ for $s \in \{+1, -1\}$ and the distrustful actions $a \in \{(C, D), (D, C)\}$ to represent the heightened uncertainty for these unfavorable actions, and define $\rho_h^R(s, \boldsymbol{a}) = 0$ otherwise. We further let $\rho_h^P(s, \boldsymbol{a}) = .1$ for all state-action pairs to represent an unbiased uncertainty in the transitions. We lastly set the parameters $\iota = .01$ and $b = 12$. After using the greedy algorithm to poison the data, the poisoned (rounded) MLE reward matrices are,

$R_0^\dagger(0, \cdot) =$

$\mathcal{A}_1/\mathcal{A}_2$	C	D
C	$-5, .62$	$-10, -5.39$
D	$0, -4.34$	$-1, -5.69$

$R_1^\dagger(0, \cdot) =$

$\mathcal{A}_1/\mathcal{A}_2$	C	D
C	$-5, -2.41$	$-10, -2.42$
D	$0, -4.82$	$-1, -4.83$

$R_1^\dagger(+1, \cdot) =$

$\mathcal{A}_1/\mathcal{A}_2$	C	D
C	$-4, -2.25$	$-11, -2.36$
D	$1, -5.47$	$0, -5.58$

$R_1^\dagger(-1, \cdot) =$

$\mathcal{A}_1/\mathcal{A}_2$	C	D
C	$-7.97, -4$	$-0.56, -11$
D	$-7.86, 1$	$-0.45, 0$

The total attack cost is approximately 82. If we say each unit of attack cost corresponds to \$100 the attacker has to pay the inside man, then the attacker can ensure Bonnie goes free in both stages by influencing the players into playing (D, C), but at the high cost of $\$8,200$. All the attacker needs to assume is that Bonnie and Clyde use very basic rationality and corresponding confidence bounds to make their decisions.

7.2 Public Good Game

Consider the iterated public goods game in which all participants need to volunteer in order to have a public good provided. Examples of such games include the donations to environmental causes, private investment for startups, and voluntary vaccination to achieve herd immunity. In all of these examples, each participant can choose between one of two actions: volunteer (V) and free ride (N). We assume that the public good is provided only when all participants choose to volunteer, and in that case, every participant gets a positive reward; and if the public good is not provided, the participants who volunteered incur a cost. To be concrete, suppose there are n participants and n_V is the number of them who volunteered, then the reward matrix is given by,

$\mathcal{A}_i/\mathcal{A}_{-i}$	$n_V = n$	$n_V < n$
V	$\frac{b}{n}(n-s)$	$-\frac{b}{n}s$
N	0	0

$$\mathbb{P}\left\{s'|s, \boldsymbol{a}\right\} = \begin{cases} 1 & \text{if } s' = n - n_V \\ 0 & \text{if } s' \neq n - n_V \end{cases}, \forall\, s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}, \tag{31}$$

$$\mu\,(s = 0) = 1. \tag{32}$$

where b is maximum value of the public good and the maximum cost of volunteering, and s is a social distrust parameter, in which the higher the social distrust, the lower the value of the public good and the higher the cost of volunteering. We consider $n + 1$ states $\mathcal{S} = \{0, 1, ..., n\}$ each representing a different value of s, and we consider the transition probabilities described by (31) and (32).

The original game has a Pareto dominant Markov perfect Nash equilibrium in which all participants volunteer in all stages and the public good is always provided. Suppose the attacker wants to sow social distrust and prevent the public good, the attacker can produce fake training data to misinform the participants about the benefits and costs and as a result incentivize them to choose not to volunteer.

A simple two learner two period three state example would have the following reward matrix in state $s \in \{0, 1, 2\}$. We take $b = 1, H = 2$, and before the attack, the stage games look like (weakly dominant strategies are shaded),

$\hat{R}_1(0, \cdot)$

$\mathcal{A}_1/\mathcal{A}_2$	V	N
V	$1, 1$	$0, 0$
N	$0, 0$	$0, 0$

$\hat{R}_2(0, \cdot)$

V	N
$1, 1$	$0, 0$
$0, 0$	$0, 0$

$\hat{R}_2(1, \cdot)$

V	N
$0.5, 0.5$	-0.5, 0
$0, -0.5$	$0, 0$

$\hat{R}_2(2, \cdot)$

V	N
$0, 0$	-1, 0
$0, -1$	$0, 0$

In the original Markov game, choosing V in both periods in all states is a weakly dominant strategy, and it is strictly dominant in period 1. We use $K = 16$ episodes in similar format as before. We set $\rho_h^R(s, \boldsymbol{a}) = 0$ for all $h, s, \boldsymbol{a}$ to represent an unbiased confidence in each reward. We set $\rho_h^P(s, \boldsymbol{a}) = 0.001$ for all h, s, a to represent a high confidence in the transitions. Now, given the target policy

$\boldsymbol{\pi}^\dagger = (N, N)$ in all stages, and the attack parameters $\iota = 0.01$, after the greedy attack, the maximum likelihood estimate of the stage games look like (strictly dominant strategies are shaded),

$R_1^\dagger(0, \cdot)$

$\mathcal{A}_1/\mathcal{A}_2$	V	N
V	$0.344, 0.344$	$-0.005, 0.359$
N	$0.359, -0.005$	$0.005, 0.005$

$R_2^\dagger(0, \cdot)$

V	N
$0.346, 0.346$	$-0.005, 0.356$
$0.356, -0.005$	$0.005, 0.005$

$R_2^\dagger(1, \cdot)$

$\mathcal{A}_1/\mathcal{A}_2$	V	N
V	$0.213, 0.213$	$-0.5, 0.223$
N	$0.223, -0.5$	$0{,}0$

$R_2^\dagger(2, \cdot)$

V	N
$-0.005, -0.005$	$-1, 0.005$
$0.005, -1$	$0, 0$

The attack cost is approximately 10.1. In the poisoned game, action V is now strictly dominated in all periods in all states, and learners using MARL algorithms using estimates with complete confidence $\rho = 0$ will always choose the action N.

8 Conclusion

We studied a security threat to offline MARL where an attacker can force learners into executing an arbitrary Dominant Strategy Equilibrium by minimally poisoning historical data. We showed that the attack problem can be formulated as a linear program, and provided analysis on the attack feasibility and cost. This paper thus helps to raise awareness on the trustworthiness of multi-agent learning. We encourage the community to study defense against such attacks, e.g. via robust statistics and reinforcement learning.

References

[1] Ashton Anderson, Yoav Shoham, and Alon Altman. Internal implementation. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 191–198. Citeseer, 2010.

[2] Kiarash Banihashem, Adish Singla, Jiarui Gan, and Goran Radanovic. Admissible policy teaching through reward design. *arXiv preprint arXiv:2201.02185*, 2022.

[3] Kiarash Banihashem, Adish Singla, and Goran Radanovic. Defense against reward poisoning attacks in reinforcement learning. *arXiv preprint arXiv:2102.05776*, 2021.

[4] Ilija Bogunovic, Arpan Losalka, Andreas Krause, and Jonathan Scarlett. Stochastic linear bandits robust to adversarial attacks. In *International Conference on Artificial Intelligence and Statistics*, pages 991–999. PMLR, 2021.

[5] Qiwen Cui and Simon S Du. When is offline two-player zero-sum Markov game solvable? *arXiv preprint arXiv:2201.03522*, 2022.

[6] Evrard Garcelon, Baptiste Roziere, Laurent Meunier, Olivier Teytaud, Alessandro Lazaric, and Matteo Pirotta. Adversarial attacks on linear contextual bandits. *arXiv preprint arXiv:2002.03839*, 2020.

[7] Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. *arXiv preprint arXiv:1905.10615*, 2019.

[8] Ziwei Guan, Kaiyi Ji, Donald J Bucci Jr, Timothy Y Hu, Joseph Palombo, Michael Liston, and Yingbin Liang. Robust stochastic bandit algorithms under probabilistic unbounded adversarial attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4036–4043, 2020.

[9] Wenbo Guo, Xian Wu, Sui Huang, and Xinyu Xing. Adversarial policy learning in two-player competitive games. In *International Conference on Machine Learning*, pages 3910–3919. PMLR, 2021.

[10] Yunhan Huang and Quanyan Zhu. Deceptive reinforcement learning under adversarial manipulations on cost signals. In *International Conference on Decision and Game Theory for Security*, pages 217–237. Springer, 2019.

[11] Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pages 5084–5096. PMLR, 2021.

[12] Kwang-Sung Jun, Lihong Li, Yuzhe Ma, and Jerry Zhu. Adversarial attacks on stochastic bandits. *Advances in Neural Information Processing Systems*, 31:3640–3649, 2018.

[13] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

[14] Fang Liu and Ness Shroff. Data poisoning attacks on stochastic bandits. In *International Conference on Machine Learning*, pages 4042–4050. PMLR, 2019.

[15] Guanlin Liu and Lifeng Lai. Provably efficient black-box action poisoning attacks against reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[16] Shiyin Lu, Guanghui Wang, and Lijun Zhang. Stochastic graphical bandits with adversarial corruptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8749–8757, 2021.

[17] Thodoris Lykouris, Max Simchowitz, Alex Slivkins, and Wen Sun. Corruption-robust exploration in episodic reinforcement learning. In *Conference on Learning Theory*, pages 3242–3245. PMLR, 2021.

[18] Yuzhe Ma, Kwang-Sung Jun, Lihong Li, and Xiaojin Zhu. Data poisoning attacks in contextual bandits. In *International Conference on Decision and Game Theory for Security*, pages 186–204. Springer, 2018.

[19] Yuzhe Ma, Young Wu, and Xiaojin Zhu. Game redesign in no-regret game playing. *arXiv preprint arXiv:2110.11763*, 2021.

[20] Yuzhe Ma, Xuezhou Zhang, Wen Sun, and Jerry Zhu. Policy poisoning in batch reinforcement learning and control. *Advances in Neural Information Processing Systems*, 32:14570–14580, 2019.

[21] Dov Monderer and Moshe Tennenholtz. k-implementation. *Journal of Artificial Intelligence Research*, 21:37–62, 2004.

[22] Amin Rakhsha, Goran Radanovic, Rati Devidze, Xiaojin Zhu, and Adish Singla. Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning. In *International Conference on Machine Learning*, pages 7974–7984. PMLR, 2020.

[23] Amin Rakhsha, Goran Radanovic, Rati Devidze, Xiaojin Zhu, and Adish Singla. Policy teaching in reinforcement learning via environment poisoning attacks. *Journal of Machine Learning Research*, 22(210):1–45, 2021.

[24] Amin Rakhsha, Xuezhou Zhang, Xiaojin Zhu, and Adish Singla. Reward poisoning in reinforcement learning: Attacks against unknown learners in unknown environments. *arXiv preprint arXiv:2102.08492*, 2021.

[25] Yanchao Sun, Da Huo, and Furong Huang. Vulnerability-aware poisoning mechanism for online rl with unknown dynamics. *arXiv preprint arXiv:2009.00774*, 2020.

[26] Chen-Yu Wei, Christoph Dann, and Julian Zimmert. A model selection approach for corruption robust reinforcement learning. In *International Conference on Algorithmic Learning Theory*, pages 1043–1096. PMLR, 2022.

[27] 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. *arXiv preprint arXiv:2203.08398*, 2022.

[28] Lin Yang, Mohammad Hajiesmaili, Mohammad Sadegh Talebi, John Lui, and Wing Shing Wong. Adversarial bandits with corruptions: Regret lower bound and no-regret algorithm. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[29] Haoqi Zhang and David C Parkes. Value-based policy teaching with active indirect elicitation. In *AAAI*, volume 8, pages 208–214, 2008.

[30] Haoqi Zhang, David C Parkes, and Yiling Chen. Policy teaching through reward function learning. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 295–304, 2009.

[31] Xuezhou Zhang, Yiding Chen, Jerry Zhu, and Wen Sun. Corruption-robust offline reinforcement learning. *arXiv preprint arXiv:2106.06630*, 2021.

[32] Xuezhou Zhang, Yiding Chen, Xiaojin Zhu, and Wen Sun. Robust policy gradient against strong data corruption. In *International Conference on Machine Learning*, pages 12391–12401. PMLR, 2021.

[33] Xuezhou Zhang, Yuzhe Ma, Adish Singla, and Xiaojin Zhu. Adaptive reward-poisoning attacks against reinforcement learning. In *International Conference on Machine Learning*, pages 11225–11234. PMLR, 2020.

[34] Han Zhong, Wei Xiong, Jiyuan Tan, Liwei Wang, Tong Zhang, Zhaoran Wang, and Zhuoran Yang. Pessimistic minimax value iteration: Provably efficient equilibrium learning from offline datasets. *arXiv preprint arXiv:2202.07511*, 2022.

[35] Shiliang Zuo. Near optimal adversarial attack on ucb bandits. *arXiv preprint arXiv:2008.09312*, 2020.

Appendices

A Compatibility with Pessimistic/Optimistic Offline MARL Algorithms

There is a growing literature on offline RL with theoretical guarantees [11, 5, 34]. In particular, to address the uncertainty due to limited coverage of the offline dataset, prior work leverages the principle of pessimism — it uses uncertainty based confidence bounds to penalize the value function on states/actions less covered — to design offline RL algorithms. This principle has been implemented and analyzed for single-agent offline RL [11]. Recent work extends the pessimism principle to offline multi-agent RL, focusing on finding the NE in *two-player zero-sum* MG: [5] considers tabular setting, while [34] considers linear MG.

While we are not aware of existing work on provably efficient offline algorithms for the general setting considered in this paper, namely *multi-player general-sum* MGs, we expect that an appropriate form of pessimism continue to apply in such settings. We note that the above algorithms are model-free approaches, i.e., the confidence bounds are applied to the value functions. In comparison, our attack formulation is developed under the assumption that the learners build confidence bounds for the MG model (i.e., the reward and transition kernel).

In this section, we show that our formulation is in fact compatible with existing model-free offline algorithms and hence our attack works on state-of-the-art learners that use such algorithms. To this end, we consider below a general class of model-free offline MARL algorithms, called Pessimistic-Optimistic Value Iteration (POVI), that is a generalization of the existing pessimistic algorithms from [11, 5, 34]. When specialized to two-player zero-sum MGs, this algorithm class recovers the pessimistic offline algorithms from [5, 34]. We emphasize that our goal is not to provide an theoretical analysis of POVI as an offline learning algorithm. Rather, we aim to show that our attack is guaranteed to be successful if the learners use any instantiation of POVI.

We now describe the POVI algorithm. Denote by $f : \mathcal{S} \to \mathbb{R}$ an arbitrary value function. Define the true Bellman operator $B^*_{i,h} : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ by

$$\left(B^*_{i,h} f\right)(s, \boldsymbol{a}) = R^*_{i,h}(s, \boldsymbol{a}) + \langle P^*_h(\cdot|s, \boldsymbol{a}), f(\cdot)\rangle\,, \tag{33}$$

where $R^*_{i,h}$ and P^*_h are the true reward function for agent i and the transition kernel at period h, respectively. Based on the offline dataset $\mathcal{D} = \left\{(s^k_h, \boldsymbol{a}^k_h, \boldsymbol{r}^k_h)\right\}_{h\in[H],k\in[K]}$, the learner constructs the empirical Bellman operator $\widehat{B}_{i,h} : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ by

$$\left(\widehat{B}_{i,h} f\right)(s, \boldsymbol{a}) = \widehat{R}_{i,h}(s, \boldsymbol{a}) + \left\langle \widehat{P}_h(\cdot|s, \boldsymbol{a}), f(\cdot)\right\rangle\,, \tag{34}$$

where $\widehat{R}_{i,h}$ and $\widehat{P}_h$ are the empirical estimates (i.e., MLEs) of the reward function of agent i and the transition kernel at period h, respectively. With these notations, the POVI algorithm, independently run by each agent i, is given below.

Algorithm 2 Pessimistic-Optimistic Value Iteration (POVI)

Require: Offline dataset $\mathcal{D} = \left\{(s^k_h, \boldsymbol{a}^k_h, \boldsymbol{r}^k_h)\right\}_{h\in[H],k\in[K]}$

  Initialization: $\underline{V}_{i,H+1}(\cdot) \leftarrow 0$ for all i

  for $h = H, H-1, \ldots, 1$ **do**

    Let $\widetilde{Q}_{i,h}(s, \boldsymbol{a}) \leftarrow \left(\widehat{B}_{i,h}\underline{V}_{i,h+1}\right)(s, \boldsymbol{a}), \quad i \in [n]$ `// empirical Q estimate`

    Let $\underline{Q}_{i,h}(s, \boldsymbol{a}) \leftarrow \widetilde{Q}_{i,h}(s, \boldsymbol{a}) - \Gamma_{i,h}(s, \boldsymbol{a}), \quad i \in [n]$ `// bonus`

    Let $\underline{\boldsymbol{\pi}}_h(\cdot|s) \leftarrow \text{NE}\left(\underline{Q}_{1,h}(s,\cdot), \ldots, \underline{Q}_{n,h}(s,\cdot)\right)$ `// NE policy`

    Let $\underline{V}_{i,h}(s) \leftarrow \langle \underline{\boldsymbol{\pi}}_h(\cdot|s), \underline{Q}_{i,h}(s,\cdot)\rangle, \quad i \in [n]$ `// V function`

  return $\underline{\boldsymbol{\pi}} = (\underline{\boldsymbol{\pi}}_h)_{i=1}^H$

In the above algorithm, $\Gamma_{i,h}(s, \boldsymbol{a})$ is a bonus term (a.k.a. uncertainty quantifier) that plays the role of confidence width for the value function. A typical choice of this bonus, suggested by concentration inequalities, takes the form $|\Gamma_{i,h}(s,\boldsymbol{a})| \propto \frac{1}{\sqrt{N_h(s,\boldsymbol{a})+1}}$, where we recall that $N_h(s,\boldsymbol{a})$ is the visit count of the state-action pair $(s,\boldsymbol{a})$. Note that we allow $\Gamma_{i,h}(s,\boldsymbol{a})$ to take an arbitrary sign, with a positive (resp., negative) $\Gamma_{i,h}(s,\boldsymbol{a})$ corresponding to pessimism (resp., optimism).

Recall that in our attack formulation, confidence intervals $\mathrm{CI}^R_{i,h}(s,\boldsymbol{a})$ and $\mathrm{CI}^P_h(s,\boldsymbol{a})$, with confidence widths $\rho^R_h(s,\boldsymbol{a})$ and $\rho^P_h(s,\boldsymbol{a})$, are constructed for the reward and transition, respectively (cf. Assumption 2). We impose the following assumption on the relationship between these confidence widths and the bonus $\Gamma_{i,h}(s,\boldsymbol{a})$ used in POVI.

Assumption 4 (Relationship between CIs)**.** The above CIs satisfy

$$|\Gamma_{i,h}(s,\boldsymbol{a})| \le \rho^R_h(s,\boldsymbol{a}) + \max_{\substack{U\in\mathbb{R}^{|\mathcal{S}|}:\sum_{s'}U(s')=0\\ \|U\|_1\le\rho^P_h(s,\boldsymbol{a})}} \left\langle U, \underline{V}_{i,h+1}\right\rangle$$

for all $(i,s,\boldsymbol{a},h) \in [n]\times\mathcal{S}\times\mathcal{A}\times[H]$.

Under this assumption, we have the following theorem, which states that the Q/value functions computed by POVI correspond to the NE of some plausible Markov game in the confidence set of our attack formulation. Recall our attack is guaranteed to be successful in installing the target policy $\boldsymbol{\pi}^\dagger$ as the unique ι-strict MPDSE (hence also the unique NE) for *all* plausible games in the confidence set (see Lemma 1). Combined with Theorem 4, we conclude that our attack will also successfully install $\boldsymbol{\pi}^\dagger$ as the output of any instantiation of POVI.

Theorem 4 (Compatibility)**.** *Under Assumption 4, there exist some reward function and transition kernel $(\boldsymbol{R}_h, P_h)_{h\in[H]}$ for which the following hold:*

1. *For all $(i,s,\boldsymbol{a},h) \in [n]\times\mathcal{S}\times\mathcal{A}\times[H]$:*
$$R_{i,h}(s,\boldsymbol{a}) \in \mathrm{CI}^R_{i,h}(s,\boldsymbol{a}),$$
$$P_h(\cdot \mid s,\boldsymbol{a}) \in \mathrm{CI}^P_h(s,\boldsymbol{a}).$$
2. *For all $(i,s,\boldsymbol{a},h) \in [n]\times\mathcal{S}\times\mathcal{A}\times[H]$:*
$$\underline{Q}_{i,h}(s,\boldsymbol{a}) = R_{i,h}(s,\boldsymbol{a}) + \left\langle P_h(\cdot \mid s,\boldsymbol{a}), \underline{V}_{i,h+1}(\cdot)\right\rangle,$$
$$\underline{V}_{i,h}(s) = \langle \underline{\boldsymbol{\pi}}_h(\cdot|s), \underline{Q}_{i,h}(s,\cdot)\rangle,$$
$$\text{where } \underline{\boldsymbol{\pi}}_h(\cdot|s) = \mathrm{NE}\left(\underline{Q}_{1,h}(s,\cdot),\ldots,\underline{Q}_{n,h}(s,\cdot)\right).$$
That is, $\underline{\boldsymbol{Q}}_h$ and $\underline{\boldsymbol{V}}_h$ are the Q and value functions of the NE of the Markov game $G = (\mathcal{S},\mathcal{A},\boldsymbol{R},P,H)$.

Proof of Theorem 4. Fix an arbitrary tuple $(i,s,\boldsymbol{a},h) \in [n]\times\mathcal{S}\times\mathcal{A}\times[H]$. By Assumption 4, there exists a $u\in\mathbb{R}$ and a $U\in\mathbb{R}^{|\mathcal{S}|}$ satisfying $|u| \le \rho^R_h(s,\boldsymbol{a})$, $\sum_{s'}U(s')=0$, $\|U\|_1 \le \rho^P_h(s,\boldsymbol{a})$ and

$$\Gamma_{i,h}(s,\boldsymbol{a}) = u + \left\langle U, \underline{V}_{i,h+1}\right\rangle. \tag{35}$$

Let

$$R_{i,h}(s,\boldsymbol{a}) = \widehat{R}_{i,h}(s,\boldsymbol{a}) - u,$$
$$P_h(\cdot \mid s,\boldsymbol{a}) = \widehat{P}_h(\cdot \mid s,\boldsymbol{a}) - U.$$

By construction it is clear that $R_{i,h}(s,\boldsymbol{a}) \in \mathrm{CI}^R_{i,h}(s,\boldsymbol{a})$ and $P_h(\cdot \mid s,\boldsymbol{a}) \in \mathrm{CI}^P_h(s,\boldsymbol{a})$, hence part 1 of the theorem holds. Moreover, we have

$$\begin{aligned}\underline{Q}_{i,h}(s,\boldsymbol{a}) &\overset{\text{(i)}}{=} \left(\widehat{B}_{i,h}\underline{V}_{i,h+1}\right)(s,\boldsymbol{a}) - \Gamma_{i,h}(s,\boldsymbol{a})\\ &\overset{\text{(ii)}}{=} \widehat{R}_{i,h}(s,\boldsymbol{a}) + \left\langle \widehat{P}_h(\cdot \mid s,\boldsymbol{a}), \underline{V}_{i,h+1}(\cdot)\right\rangle - u - \left\langle U, \underline{V}_{i,h+1}\right\rangle\\ &\overset{\text{(iii)}}{=} R_{i,h}(s,\boldsymbol{a}) + \left\langle P_h(\cdot \mid s,\boldsymbol{a}), \underline{V}_{i,h+1}(\cdot)\right\rangle,\end{aligned}$$

where step (i) follows from Line 4 in Algorithm 2, step (ii) follows from the definition of $\widehat{B}_{i,h}$ in (34) and the expression of $\Gamma_{i,h}(s,\boldsymbol{a})$ in (35), and step (iii) follows from the above construction of $R_{i,h}(s,\boldsymbol{a})$ and $P_h(\cdot \mid s,\boldsymbol{a})$. This proves the first equation in Part 2 of the theorem. The remaining equations in Part 2 are from the POVI algorithm specification. $\square$

Remark 1. Below we discuss when Assumption 4 holds and how it is related to common choices of the confidence widths $\Gamma_h, \rho_h^R, \rho_h^P$.

- A sufficient condition for Assumption 4 is

$$|\Gamma_{i,h}(s, \boldsymbol{a})| \leq \rho_h^R(s, \boldsymbol{a}), \qquad \forall i, s, \boldsymbol{a}, h. \tag{36}$$

  This condition becomes equivalent to Assumption 4 when $\underline{V}_{i,h+1}$ is a constant function, i.e., $\underline{V}_{i,h+1}(s') = \underline{V}_{i,h+1}(s''), \forall s', s'' \in \mathcal{S}$.

- The sufficient condition (36) and in turn Assumption 4 are satisfied for the following choices of the bonus term and CI widths:

$$|\Gamma_{i,h}(s, \boldsymbol{a})| = H\sqrt{\frac{\beta}{N_h(s, \boldsymbol{a}) + 1}},$$
$$\rho_h^R(s, \boldsymbol{a}) = H\sqrt{\frac{\beta}{N_h(s, \boldsymbol{a}) + 1}},$$
$$\rho_h^P(s, \boldsymbol{a}) = \sqrt{\frac{|\mathcal{S}|\,\beta}{N_h(s, \boldsymbol{a}) + 1}}.$$

  where β denotes a logarithmic term of the form $\beta := c\log(|\mathcal{S}|\,|\mathcal{A}|\,HN\delta^{-1})$, with c being a universal constant, $N := |\mathcal{D}| = \sum_{s,a,h} N_h(s, \boldsymbol{a})$ and δ the desired failure probability. Note that the above choice of $\Gamma_{i,h}(s, \boldsymbol{a})$ is similar to those used in existing work on offline MARL [5, 34]. The above choices of $\rho_h^R(s, \boldsymbol{a})$ and $\rho_h^P(s, \boldsymbol{a})$, given by Hoeffding-type concentration inequalities, are also similar to those typically used in existing model-based RL algorithms.

B Linear Program Formulations

B.1 Bandit Game Maximum Likelihood Learners

The problem (4)– (7) can be converted into the following linear program,

$$\min_{r^\dagger, t, R^\dagger} \sum_{i=1}^{n} \sum_{k=1}^{K} t_i^{(k)} \tag{37}$$
$$\text{such that } r_i^{\dagger,(k)} - r_i^{0,(k)} \leq t_i^{(k)}, \forall\, k, i \tag{38}$$
$$r_i^{0,(k)} - r_i^{\dagger,(k)} \leq -t_i^{(k)}, \forall\, k, i \tag{39}$$
$$R_i^\dagger(\boldsymbol{a}) = \frac{1}{N(\boldsymbol{a})} \sum_{k=1}^{K} r_i^{\dagger,(k)} \mathbb{1}_{\{\boldsymbol{a}^{(k)} = \boldsymbol{a}\}}, \forall\, \boldsymbol{a}, i \tag{40}$$
$$R_i^\dagger(a_i, a_{-i}) - R_i^\dagger\left(\pi_i^\dagger, a_{-i}\right) \leq -\iota, \forall\, i, a_{-i}, a_i \neq \pi_i^\dagger \tag{41}$$
$$r_i^{\dagger,(k)} \leq b, \forall\, k, i \tag{42}$$
$$- r_i^{\dagger,(k)} \leq -b, \forall\, k, i. \tag{43}$$

To linearize the L^1 -norm, we introduce slack variables t, and write,

$$\min_{r^\dagger} \left\| r^\dagger - r^0 \right\|_1, \tag{44}$$

as,

$$\min_{t, r^\dagger} e^T t \tag{45}$$
$$\text{such that } -t \leq r^\dagger - r^0 \leq t. \tag{46}$$

B.2 Confidence Bound Learners

B.2.1 Section 4.1 Bandit Game Confidence Bound Learners

The problem (8)– (12) can be converted into the following linear program,

$$\min_{r^\dagger, R^\dagger, t, m} \sum_{i=1}^{n} \sum_{k=1}^{K} t_i^{(k)} \tag{47}$$

$$\text{such that } r_i^{\dagger,(k)} - r_i^{0,(k)} \leq t_i^{(k)}, \forall\, k, i \tag{48}$$

$$r_i^{0,(k)} - r_i^{\dagger,(k)} \leq -t_i^{(k)}, \forall\, k, i \tag{49}$$

$$R_i^\dagger(\boldsymbol{a}) = \frac{1}{N(\boldsymbol{a})} \sum_{k=1}^{K} r_i^{\dagger,(k)} \mathbb{1}_{\{\boldsymbol{a}^{(k)} = \boldsymbol{a}\}}, \forall\, \boldsymbol{a}, i \tag{50}$$

$$\begin{matrix} -\overline{m}_i^- \left(\pi_i^\dagger, a_{-i}\right) - \overline{m}_i^+ \left(\pi_i^\dagger, a_{-i}\right) - \underline{m}_i^- (a_i, a_{-i}) - \underline{m}_i^+ (a_i, a_{-i}) \\ \leq -2b - \iota \end{matrix}, \forall\, i, a_{-i}, a_i \neq \pi_i^\dagger \tag{51}$$

$$\begin{matrix} -\overline{m}_i^- \left(\pi_i^\dagger, a_{-i}\right) - \overline{m}_i^+ \left(\pi_i^\dagger, a_{-i}\right) - \underline{m}_i^- (a_i, a_{-i}) - \underline{m}_i^+ (a_i, a_{-i}) \\ + R_i^\dagger \left(\pi_i^\dagger, a_{-i}\right) - R_i^\dagger (a_i, a_{-i}) \leq -\rho^R \left(\pi_i^\dagger, a_{-i}\right) - \rho^R (a_i, a_{-i}) - \iota \end{matrix}, \forall\, i, a_{-i}, a_i \neq \pi_i^\dagger \tag{52}$$

$$\begin{matrix} -\overline{m}_i^- \left(\pi_i^\dagger, a_{-i}\right) - \overline{m}_i^+ \left(\pi_i^\dagger, a_{-i}\right) - \underline{m}_i^- (a_i, a_{-i}) - \underline{m}_i^+ (a_i, a_{-i}) \\ + R_i^\dagger \left(\pi_i^\dagger, a_{-i}\right) \leq -b - \rho^R \left(\pi_i^\dagger, a_{-i}\right) - \iota \end{matrix}, \forall\, i, a_{-i}, a_i \neq \pi_i^\dagger \tag{53}$$

$$\begin{matrix} -\overline{m}_i^- \left(\pi_i^\dagger, a_{-i}\right) - \overline{m}_i^+ \left(\pi_i^\dagger, a_{-i}\right) - \underline{m}_i^- (a_i, a_{-i}) - \underline{m}_i^+ (a_i, a_{-i}) \\ - R_i^\dagger (a_i, a_{-i}) \leq -\rho^R (a_i, a_{-i}) - b - \iota \end{matrix}, \forall\, i, a_{-i}, a_i \neq \pi_i^\dagger \tag{54}$$

$$-\overline{m}_i^- \left(\pi_i^\dagger, a_{-i}\right) - R_i^\dagger \left(\pi_i^\dagger, a_{-i}\right) \leq \rho^R \left(\pi_i^\dagger, a_{-i}\right) - b, \forall\, i, a_{-i}, a_i \neq \pi_i^\dagger \tag{55}$$

$$-\overline{m}_i^+ \left(\pi_i^\dagger, a_{-i}\right) + R_i^\dagger \left(\pi_i^\dagger, a_{-i}\right) \leq -\rho^R \left(\pi_i^\dagger, a_{-i}\right) + b, \forall\, i, a_{-i}, a_i \neq \pi_i^\dagger \tag{56}$$

$$-\underline{m}_i^- (a_i, a_{-i}) + R_i^\dagger \left(\pi_i^\dagger, a_{-i}\right) \leq -\rho^R (a_i, a_{-i}) + b, \forall\, i, a_{-i}, a_i \neq \pi_i^\dagger \tag{57}$$

$$-\underline{m}_i^+ (a_i, a_{-i}) - R_i^\dagger \left(\pi_i^\dagger, a_{-i}\right) \leq \rho^R (a_i, a_{-i}) - b, \forall\, i, a_{-i}, a_i \neq \pi_i^\dagger \tag{58}$$

$$r_i^{\dagger,(k)} \leq b, \forall\, k, i \tag{59}$$

$$-r_i^{\dagger,(k)} \leq -b, \forall\, k, i \tag{60}$$

$$\overline{m}_i^- \left(\pi_i^\dagger, a_{-i}\right), \overline{m}_i^+ \left(\pi_i^\dagger, a_{-i}\right), \underline{m}_i^- (a_i, a_{-i}), \underline{m}_i^+ (a_i, a_{-i}) \geq 0, \forall\, i, a_{-i}, a_i \neq \pi_i^\dagger. \tag{61}$$

The same linearization is done to the L^1 -norm objective. To linearize the $\max$ and $\min$, we introduce slack variables $\overline{m}^-, \overline{m}^+, \underline{m}^-, \underline{m}^+$ to rewrite the constraints,

$$\max\{-b, R_1 - \rho_1\} \geq \min\{b, R_2 + \rho_2\} + \iota, \tag{62}$$

as,

$$\min\{-b, R_1 - \rho_1\} + |R_1 - \rho_1 + b| \geq \max\{b, R_2 + \rho_2\} - |R_2 + \rho_2 - b| + \iota, \tag{63}$$

which can then be converted to the following set of linear constraints,

$$-b + \overline{m}^- + \overline{m}^+ \geq b - \underline{m}^- - \underline{m}^+ + \iota \tag{64}$$

$$R_1 - \rho_1 + \overline{m}^- + \overline{m}^+ \geq R_2 + \rho_2 - \underline{m}^- - \underline{m}^+ + \iota \tag{65}$$
$$-b + \overline{m}^- + \overline{m}^+ \geq R_2 + \rho_2 - \underline{m}^- - \underline{m}^+ + \iota \tag{66}$$
$$R_1 - \rho_1 + \overline{m}^- + \overline{m}^+ \geq b - \underline{m}^- - \underline{m}^+ + \iota \tag{67}$$
$$\overline{m}^- \geq -R_1 - \rho_1 - b \tag{68}$$
$$\overline{m}^+ \geq R_1 - \rho_1 + b \tag{69}$$
$$\underline{m}^- \geq -R_2 - \rho_2 + b \tag{70}$$
$$\underline{m}^+ \geq R_2 + \rho_2 - b \tag{71}$$
$$\overline{m}^-, \overline{m}^+, \underline{m}^-, \underline{m}^+ \geq 0. \tag{72}$$

We do the same conversion for each $\boldsymbol{a} \in \mathcal{A}$ to obtain the above linear problem.

B.2.2 Section 4.2 Markov Game Confidence Bound Learners

The problem (13)$-$ (21) can be converted into the following linear program,

$$\min_{r^\dagger, t, u, v, w} \sum_{i=1}^{n} \sum_{k=1}^{K} \sum_{k=1}^{K} t_{i,h}^{(k)} \tag{73}$$
$$\text{such that } r_{i,h}^{\dagger,(k)} - r_{i,h}^{0,(k)} \leq t_{i,h}^{(k)}, \forall\, h, k, i \tag{74}$$
$$r_{i,h}^{0,(k)} \leq -t_{i,h}^{(k)}, \forall\, h, k, i \tag{75}$$
$$R_{i,h}^{\dagger}(s, \boldsymbol{a}) = \frac{1}{N_h(s, \boldsymbol{a})} \sum_{k=1}^{K} r_{i,h}^{\dagger,(k)} \mathbb{1}_{\left\{s_h^{(k)} = s, \boldsymbol{a}_h^{(k)} = \boldsymbol{a}\right\}}, \forall\, h, s, i, \boldsymbol{a} \tag{76}$$
$$\underline{Q}_{i,H}(s, \boldsymbol{a}) = R_{i,H}^{\dagger}(s, \boldsymbol{a}) - \rho_H^R(s, \boldsymbol{a}), \forall\, s, i, \boldsymbol{a} \tag{77}$$
$$\overline{Q}_{i,H}(s, \boldsymbol{a}) = R_{i,H}^{\dagger}(s, \boldsymbol{a}) + \rho_H^R(s, \boldsymbol{a}), \forall\, s, i, \boldsymbol{a} \tag{78}$$
$$\underline{Q}_{i,h}(s, \boldsymbol{a}) = \begin{array}{c} R_{i,h}^{\dagger}(s, \boldsymbol{a}) - \rho_h^R(s, \boldsymbol{a}) - \sum_{s' \in \mathcal{S}} \hat{P}_h(s'|s, \boldsymbol{a}) \left[\underline{u}_{i,h}(s, \boldsymbol{a}) - \underline{v}_{i,h}(s, \boldsymbol{a})\right]_{s'} \\ - \left\{\sum_{s' \in \mathcal{S}} \rho_h^P(s, \boldsymbol{a}) \left[\underline{u}_{i,h}(s, \boldsymbol{a}) + \underline{v}_{i,h}(s, \boldsymbol{a})\right]_{s'}\right\} - \underline{w}_{i,h}(s, \boldsymbol{a}) \end{array}, \forall\, \boldsymbol{a} \tag{79}$$
$$\overline{Q}_{i,h}(s, \boldsymbol{a}) = \begin{array}{c} R_{i,h}^{\dagger}(s, \boldsymbol{a}) + \rho_h^R(s, \boldsymbol{a}) + \sum_{s' \in \mathcal{S}} \hat{P}_h(s'|s, \boldsymbol{a}) \left[\overline{u}_{i,h}(s, \boldsymbol{a}) - \overline{v}_{i,h}(s, \boldsymbol{a})\right]_{s'} \\ + \left\{\sum_{s' \in \mathcal{S}} \rho_h^P(s, \boldsymbol{a}) \left[\overline{u}_{i,h}(s, \boldsymbol{a}) + \overline{v}_{i,h}(s, \boldsymbol{a})\right]_{s'}\right\} + \overline{w}_{i,h}(s, \boldsymbol{a}) \end{array}, \forall\, \boldsymbol{a} \tag{80}$$
$$-\underline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}(s')\right) \leq \left[\underline{u}_{i,h}(s, \boldsymbol{a}) - \underline{v}_{i,h}(s, \boldsymbol{a}) + \underline{w}_{i,h}(s, \boldsymbol{a})\right]_{s'}, \forall\, h, s, s', i, \boldsymbol{a} \tag{81}$$
$$\overline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}(s')\right) \leq \left[\overline{u}_{i,h}(s, \boldsymbol{a}) - \overline{v}_{i,h}(s, \boldsymbol{a}) + \overline{w}_{i,h}(s, \boldsymbol{a})\right]_{s'} \geq, \forall\, h, s, s', i, \boldsymbol{a} \tag{82}$$
$$\overline{Q}_{i,h}(s, (a_i, a_{-i})) - \underline{Q}_{i,h}\left(s, \left(\pi_{i,h}^{\dagger}(s), a_{-i}\right)\right) \leq -\iota, \forall\, h, s, i, a_{-i}, a_i \neq \pi_{i,h}^{\dagger}(s) \tag{83}$$
$$r_{i,h}^{\dagger,(k)} \leq b, \forall\, h, k, i \tag{84}$$
$$-r_{i,h}^{\dagger,(k)} \leq -b, \forall\, h, k, i \tag{85}$$
$$\left[\underline{u}_{i,h}(s, \boldsymbol{a})\right]_{s'}, \left[\underline{v}_{i,h}(s, \boldsymbol{a})\right]_{s'}, \left[\overline{u}_{i,h}(s, \boldsymbol{a})\right]_{s'}, \left[\overline{v}_{i,h}(s, \boldsymbol{a})\right]_{s'} \geq 0, \forall\, h, s, s', i, \boldsymbol{a} \tag{86}$$

The same linearization is done to the L^1 -norm objective. We ignore the boundary clipping on the confidence bounds for this problem to simplify the notations. To linearize the inner optimizations, we find the dual of the following problem and substitute it into the original optimization,

$$\max_{P_h \in \mathrm{CI}_h^P(s,\boldsymbol{a})} \sum_{s' \in \mathcal{S}} P_h\left(s'\right) \overline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}\left(s'\right)\right), \tag{87}$$

where,

$$\mathrm{CI}_h^P\left(s,\boldsymbol{a}\right) = \left\{P_h \in \Delta\left(\mathcal{A}\right) : \left\|P_h - \hat{P}_h\left(s,\boldsymbol{a}\right)\right\|_1 \leq \rho_h^P\left(s,\boldsymbol{a}\right)\right\}. \tag{88}$$

We treat P_h as a vector of size $|\mathcal{S}|$ with $\left[P_h\right]_{s'} = P_h\left(s'\right)$, and we define $\overline{Q}_{i,h+1}\left(\boldsymbol{\pi}_{h+1}^{\dagger}\right)$ as a vector of size $|\mathcal{S}|$ with $\left[\overline{Q}_{i,h+1}\left(\boldsymbol{\pi}_{h+1}^{\dagger}\right)\right]_{s'} = \overline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}\left(s'\right)\right)$, and write the constrained optimization as,

$$\begin{aligned}
&\max_{P_h} P_h \cdot \overline{Q}_{i,h+1}\left(\boldsymbol{\pi}_{h+1}^{\dagger}\right) && (89)\\
&\text{such that } P_h \leq \hat{P}_h\left(s,\boldsymbol{a}\right) + \rho_h^P\left(s,\boldsymbol{a}\right) && (90)\\
&P_h \geq \hat{P}_h\left(s,\boldsymbol{a}\right) - \rho_h^P\left(s,\boldsymbol{a}\right) && (91)\\
&P_h \cdot e_{|\mathcal{S}|} = 1 && (92)\\
&P_h \geq 0, && (93)
\end{aligned}$$

and in the standard form,

$$\begin{aligned}
&\min_{P_h} \overline{Q}_{i,h+1}^T\left(\boldsymbol{\pi}_{h+1}^{\dagger}\right) P_h && (94)\\
&\text{such that } \begin{bmatrix} I_{|\mathcal{S}|} \\ -I_{|\mathcal{S}|} \end{bmatrix} P_h \leq \begin{bmatrix} \hat{P}_h\left(s,\boldsymbol{a}\right) + \rho_h^P\left(s,\boldsymbol{a}\right) \\ -\hat{P}_h\left(s,\boldsymbol{a}\right) + \rho_h^P\left(s,\boldsymbol{a}\right) \end{bmatrix} && (95)\\
&e_{|\mathcal{S}|}^T P_h = 1 && (96)\\
&P_h \geq 0, && (97)
\end{aligned}$$

where the notation I_n is the $n \times n$ identity matrix and e_n is the vector of n ones.
We use the linear programming duality to get the dual problem,

$$\begin{aligned}
&\min_{u,v,w} \begin{bmatrix} \hat{P}_h\left(s,\boldsymbol{a}\right) + \rho_h^P\left(s,\boldsymbol{a}\right) \\ -\hat{P}_h\left(s,\boldsymbol{a}\right) + \rho_h^P\left(s,\boldsymbol{a}\right) 1^T \begin{bmatrix} u \\ v \\ w \end{bmatrix} \end{bmatrix} && (98)\\
&\text{such that } \begin{bmatrix} I_{|\mathcal{S}|} \\ -I_{|\mathcal{S}|} \\ e_{|\mathcal{S}|}^T \end{bmatrix}^T \begin{bmatrix} u \\ v \\ w \end{bmatrix} \geq \overline{Q}_{i,h+1}\left(\boldsymbol{\pi}_{h+1}^{\dagger}\right) && (99)\\
&\begin{bmatrix} u \\ v \end{bmatrix} \geq 0, && (100)
\end{aligned}$$

which is equivalent to,

$$\begin{aligned}
&\min_{u,v,w} \sum_{s' \in \mathcal{S}} \hat{P}_h\left(s'|s,\boldsymbol{a}\right)\left(u_{s'} - v_{s'}\right) + \rho_h^P\left(s,\boldsymbol{a}\right)\left(u_{s'} + v_{s'}\right) + w && (101)\\
&\text{such that } u_{s'} - v_{s'} + w \geq \overline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}\left(s'\right), \forall\, s' \in \mathcal{S}\right. && (102)\\
&u_{s'}, v_{s'} \geq 0, \forall\, s' \in \mathcal{S}, && (103)
\end{aligned}$$

where $u \in \mathbb{R}^{|\mathcal{S}|}, v \in \mathbb{R}^{|\mathcal{S}|}, w \in \mathbb{R}$.
The same problem needs to be solved for $\underline{Q}_{i,h+1}$, and for each h in $[H], s \in \mathcal{S}, i \in [n]$, and $\boldsymbol{a}$ in $\mathcal{A}$.

B.3 Section 6: Greedy Backward-Induction Attack

The optimization in the greedy algorithm (24)− (30) can be converted into the following linear problem, one for each learner $i \in [n]$, state $s \in \mathcal{S}$, and period $h \in [H]$,

For subproblems for learner i state s in period $h = H$,

$$\min_{r^{\dagger},t} \sum_{k=1}^{K} t_{i,H}^{(k)} \tag{104}$$

$$\text{such that } r_{i,H}^{\dagger,(k)} - r_{i,H}^{0,(k)} \leq t_{i,H}^{(k)}, \forall\, k \text{ with } s_H^{(k)} = s \tag{105}$$

$$r_{i,H}^{0,(k)} - r_{i,H}^{\dagger,(k)} \leq -t_{i,H}^{(k)}, \forall\, k \text{ with } s_H^{(k)} = s \tag{106}$$

$$R_{i,H}^{\dagger}(s, \boldsymbol{a}) = \frac{1}{N_H(s, \boldsymbol{a})} \sum_{k=1}^{K} r_{i,H}^{\dagger,(k)} \mathbb{1}_{\left\{s_H^{(k)} = s, \boldsymbol{a}_H^{(k)} = \boldsymbol{a}\right\}}, \forall\, \boldsymbol{a} \tag{107}$$

$$\underline{Q}_{i,H}(s, \boldsymbol{a}) = R_{i,H}^{\dagger}(s, \boldsymbol{a}) - \rho_h^R(s, \boldsymbol{a}), \forall\, \boldsymbol{a} \tag{108}$$

$$\overline{Q}_{i,H}(s, \boldsymbol{a}) = R_{i,H}^{\dagger}(s, \boldsymbol{a}) + \rho_h^R(s, \boldsymbol{a}), \forall\, \boldsymbol{a} \tag{109}$$

$$\overline{Q}_{i,H}(s, (a_i, a_{-i})) - \underline{Q}_{i,H}\left(s, \left(\pi_{i,h}^{\dagger}(s), a_{-i}\right)\right) \leq -\iota, \forall\, a_{-i}, a_i \neq \pi_{i,h}^{\dagger}(s) \tag{110}$$

$$r_{i,H}^{\dagger,(k)} \leq b, \forall\, k \tag{111}$$

$$-r_{i,H}^{\dagger,(k)} \leq -b, \forall\, k \tag{112}$$

Note that we store the variables $\underline{Q}_{i,H}(s, \boldsymbol{a})$ and $\overline{Q}_{i,H}(s, \boldsymbol{a})$ for each $s \in \mathcal{S}$ and $\boldsymbol{a} \in \mathcal{A}$ and use a subset of them in the following subproblems.

For subproblems for learner i state s in period $h < H$,

$$\min_{r^{\dagger},t,u,v,w} \sum_{k=1}^{K} t_{i,h}^{(k)} \tag{113}$$

$$\text{such that } r_{i,h}^{\dagger,(k)} - r_{i,h}^{0,(k)} \leq t_{i,h}^{(k)}, \forall\, k \text{ with } s_h^{(k)} = s \tag{114}$$

$$r_{i,h}^{0,(k)} - r_{i,h}^{\dagger,(k)} \leq -t_{i,h}^{(k)}, \forall\, k \text{ with } s_h^{(k)} = s \tag{115}$$

$$R_{i,h}^{\dagger}(s, \boldsymbol{a}) = \frac{1}{N_h(s, \boldsymbol{a})} \sum_{k=1}^{K} r_{i,h}^{\dagger,(k)} \mathbb{1}_{\left\{s_h^{(k)} = s, \boldsymbol{a}_h^{(k)} = \boldsymbol{a}\right\}}, \forall\, \boldsymbol{a} \tag{116}$$

$$\underline{Q}_{i,h}(s, \boldsymbol{a}) = \begin{array}{c} R_{i,h}^{\dagger}(s, \boldsymbol{a}) - \rho_h^R(s, \boldsymbol{a}) - \sum_{s' \in \mathcal{S}} \hat{P}_h(s'|s, \boldsymbol{a}) \left(\underline{u}_{s'}(\boldsymbol{a}) - \underline{v}_{s'}(\boldsymbol{a})\right) \\ -\left\{\sum_{s' \in \mathcal{S}} \rho_h^P(s, \boldsymbol{a}) \left(\underline{u}_{s'}(\boldsymbol{a}) + \underline{v}_{s'}(\boldsymbol{a})\right)\right\} - \underline{w}(\boldsymbol{a}) \end{array}, \forall\, \boldsymbol{a} \tag{117}$$

$$\overline{Q}_{i,h}(s, \boldsymbol{a}) = \begin{array}{c} R_{i,h}^{\dagger}(s, \boldsymbol{a}) + \rho_h^R(s, \boldsymbol{a}) + \sum_{s' \in \mathcal{S}} \hat{P}_h(s'|s, \boldsymbol{a}) \left(\overline{u}_{s'}(\boldsymbol{a}) - \overline{v}_{s'}(\boldsymbol{a})\right) \\ +\left\{\sum_{s' \in \mathcal{S}} \rho_h^P(s, \boldsymbol{a}) \left(\overline{u}_{s'}(\boldsymbol{a}) + \overline{v}_{s'}(\boldsymbol{a})\right)\right\} + \overline{w}(\boldsymbol{a}) \end{array}, \forall\, \boldsymbol{a} \tag{118}$$

$$-\underline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}(s')\right) \leq \underline{u}_{s'}(\boldsymbol{a}) - \underline{v}_{s'}(\boldsymbol{a}) + \underline{w}(\boldsymbol{a}), \forall\, s', \boldsymbol{a} \tag{119}$$

$$\overline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}(s')\right) \leq \overline{u}_{s'}(\boldsymbol{a}) - \overline{v}_{s'}(\boldsymbol{a}) + \overline{w}(\boldsymbol{a}), \forall\, s', \boldsymbol{a} \tag{120}$$

$$\overline{Q}_{i,h}(s, (a_i, a_{-i})) - \underline{Q}_{i,h}\left(s, \left(\pi_{i,h}^{\dagger}(s), a_{-i}\right)\right) \leq -\iota, \forall\, a_{-i}, a_i \neq \pi_{i,h}^{\dagger}(s) \tag{121}$$

$$r_{i,h}^{\dagger,(k)} \leq b, \forall\, k \tag{122}$$
$$-\, r_{i,h}^{\dagger,(k)} \leq -b, \forall\, k \tag{123}$$
$$\underline{u}_{s'}\left(\boldsymbol{a}\right), \underline{v}_{s'}\left(\boldsymbol{a}\right), \overline{u}_{s'}\left(\boldsymbol{a}\right), \overline{v}_{s'}\left(\boldsymbol{a}\right) \geq 0, \forall\, s', \boldsymbol{a} \tag{124}$$

The same linearization is done to the L^1 -norm objective and the inner optimizations. Note that we ignore the boundary clipping on the confidence bounds for this problem to simplify the notations.

We solve the subproblems with $h = H-1, H-2, ..., 1$ in the backward order and we store the variables $\underline{Q}_{i,h}\left(s, \boldsymbol{a}\right)$ and $\overline{Q}_{i,h}\left(s, \boldsymbol{a}\right)$ for each $s \in \mathcal{S}$ and $\boldsymbol{a} \in \mathcal{A}$ and use a subset of them in the subproblems in period $h-1$.

C Proofs

C.1 Section 4.2

Proof of Lemma 1:

Proof. Given $G \in \mathrm{CI}^G$, we have, for every $i \in [n], h \in [H], s \in \mathcal{S}$, and $\boldsymbol{a} \in \mathcal{A}$,

$$R_{i,h}\left(s, \boldsymbol{a}\right) \in \mathrm{CI}_h^{R^\dagger}\left(s, \boldsymbol{a}\right), \tag{125}$$
$$P_h\left(s, \boldsymbol{a}\right) \in \mathrm{CI}_h^{P}\left(s, \boldsymbol{a}\right), \tag{126}$$

where we abuse the notation $R_{i,h}\left(s, \boldsymbol{a}\right)$ to represent the mean reward after the attack, and we compute the Q values based on $R_{i,h}\left(s, \boldsymbol{a}\right)$ and the target policy $\boldsymbol{\pi}^\dagger$.
In period H, we have, for every $i \in [n], h \in [H], s \in \mathcal{S}$, and $\boldsymbol{a} \in \mathcal{A}$,

$$\begin{aligned} Q_{i,H}\left(s, \boldsymbol{a}\right) &= R_{i,H}\left(s, \boldsymbol{a}\right) &(127)\\ &\leq \max_{R_{i,H} \in \mathrm{CI}_{i,H}^{R^\dagger}(s,\boldsymbol{a})} R_{i,H} &(128)\\ &= \overline{Q}_{i,H}\left(s, \boldsymbol{a}\right), &(129)\end{aligned}$$

and,

$$\begin{aligned} Q_{i,H}\left(s, \boldsymbol{a}\right) &= R_{i,H}\left(s, \boldsymbol{a}\right) &(130)\\ &\geq \min_{R_{i,H} \in \mathrm{CI}_{i,H}^{R^\dagger}(s,\boldsymbol{a})} R_{i,H} &(131)\\ &= \underline{Q}_{i,H}\left(s, \boldsymbol{a}\right). &(132)\end{aligned}$$

As a result, we have, for any $i \in [n], s \in \mathcal{S}, a_{-i}, a_i \neq \pi_{i,H}^\dagger\left(s\right)$,

$$\begin{aligned} Q_{i,H}\left(s, \left(\pi_{i,H}^\dagger\left(s\right), a_{-i}\right)\right) &\geq \underline{Q}_{i,H}\left(s, \left(\pi_{i,H}^\dagger\left(s\right), a_{-i}\right)\right. &(133)\\ &\geq \overline{Q}_{i,H}\left(s, \left(a_i, a_{-i}\right)\right) + \iota &(134)\\ &\geq Q_{i,H}\left(s, \left(a_i, a_{-i}\right)\right) + \iota. &(135)\end{aligned}$$

Therefore, we have, in period H, and every $s \in \mathcal{S}$, $\boldsymbol{\pi}_H^\dagger\left(s\right)$ is a ι-strict dominant strategy equilibrium.

We continue by induction, and assume in period $h+1$, for every $s \in \mathcal{S}$, we have

$$Q_{i,h+1}\left(s, \boldsymbol{a}\right) \in \left[\underline{Q}_{i,h+1}\left(s, \boldsymbol{a}\right), \overline{Q}_{i,h+1}\left(s, \boldsymbol{a}\right)\right], \tag{136}$$

and $\boldsymbol{\pi}_{h+1}^\dagger\left(s\right)$ is a ι-strict dominant strategy equilibrium. Then, we have in period h, for every $i \in [n], s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}$,

$$\begin{aligned} Q_{i,h}\left(s, \boldsymbol{a}\right) &= R_{i,h}\left(s, \boldsymbol{a}\right) + \sum_{s' \in \mathcal{S}} P_h\left(s' | s, \boldsymbol{a}\right) Q_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^\dagger\left(s'\right)\right) &(137)\\ &\leq R_{i,h}\left(s, \boldsymbol{a}\right) + \sum_{s' \in \mathcal{S}} P_h\left(s' | s, \boldsymbol{a}\right) \overline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^\dagger\left(s'\right)\right) &(138)\end{aligned}$$

$$\leq \max_{R_{i,h} \in \mathrm{CI}_h^{R^\dagger}(s,\boldsymbol{a})} R_{i,h} + \max_{P_h \in \mathrm{CI}_h^{P}(s,\boldsymbol{a})} \sum_{s' \in \mathcal{S}} P_h\left(s'\right) \overline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}\left(s'\right)\right) \tag{139}$$

$$= \overline{Q}_{i,h}\left(s, \boldsymbol{a}\right), \tag{140}$$

and,

$$Q_{i,h}\left(s, \boldsymbol{a}\right) = R_{i,h}\left(s, \boldsymbol{a}\right) + \sum_{s' \in \mathcal{S}} P_h\left(s' | s, \boldsymbol{a}\right) Q_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}\left(s'\right)\right) \tag{141}$$

$$\geq R_{i,h}\left(s, \boldsymbol{a}\right) + \sum_{s' \in \mathcal{S}} P_h\left(s' | s, \boldsymbol{a}\right) \underline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}\left(s'\right)\right) \tag{142}$$

$$\geq \min_{R_{i,h} \in \mathrm{CI}_h^{R^\dagger}(s,\boldsymbol{a})} R_{i,h} + \min_{P_h \in \mathrm{CI}_h^{P}(s,\boldsymbol{a})} \sum_{s' \in \mathcal{S}} P_h\left(s'\right) \underline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}\left(s'\right)\right) \tag{143}$$

$$= \underline{Q}_{i,h}\left(s, \boldsymbol{a}\right). \tag{144}$$

As a result, we have, for any $a_{-i}, a_i \neq \pi_{i,h}^{\dagger}\left(s\right)$,

$$Q_{i,h}\left(s, \left(\pi_{i,h}^{\dagger}\left(s\right), a_{-i}\right)\right) \geq \underline{Q}_{i,h}\left(s, \left(\pi_{i,h}^{\dagger}\left(s\right), a_{-i}\right)\right) \tag{145}$$

$$\geq \overline{Q}_{i,h}\left(s, \left(a_i, a_{-i}\right)\right) + \iota \tag{146}$$

$$\geq Q_{i,h}\left(s, \left(a_i, a_{-i}\right)\right) + \iota. \tag{147}$$

Therefore, $\boldsymbol{\pi}_h^{\dagger}\left(s\right)$ is the ι-strict dominant strategy equilibrium in period h, state s.
Since $\boldsymbol{\pi}^{\dagger}$ is a Markov policy, it is the ι-strict Markov perfect dominant strategy equilibrium. □

Proof of Theorem 1:

Proof. We restate the constraints in the attacker's problem,

$$R_{i,h}^{\dagger}\left(s, \boldsymbol{a}\right) = \frac{1}{N_h\left(s, \boldsymbol{a}\right)} \sum_{k=1}^{K} r_{i,h}^{\dagger,(k)} \mathbb{1}_{\left\{s_h^{(k)}=s, \boldsymbol{a}_h^{(k)}=\boldsymbol{a}\right\}}, \forall\, h, s, i, \boldsymbol{a}, \tag{148}$$

$$\underline{Q}_{i,h}\left(s, \boldsymbol{a}\right) = \min_{\boldsymbol{R}_h \in \mathrm{CI}_h^{R^\dagger}(s,\boldsymbol{a})} \boldsymbol{R}_h + \mathbb{1}_{h<H} \min_{P_h \in \mathrm{CI}_h^{P}(s,\boldsymbol{a})} \sum_{s' \in \mathcal{S}} P_h\left(s'\right) \underline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}\left(s'\right)\right), \forall\, h, s, i, \boldsymbol{a}, \tag{149}$$

$$\overline{Q}_{i,h}\left(s, \boldsymbol{a}\right) = \max_{\boldsymbol{R}_h \in \mathrm{CI}_h^{R^\dagger}(s,\boldsymbol{a})} \boldsymbol{R}_h + \mathbb{1}_{h<H} \max_{P_h \in \mathrm{CI}_h^{P}(s,\boldsymbol{a})} \sum_{s' \in \mathcal{S}} P_h\left(s'\right) \overline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}\left(s'\right)\right), \forall\, h, s, i, \boldsymbol{a}, \tag{150}$$

$$\underline{Q}_{i,h}\left(s, \left(\pi_{i,h}^{\dagger}\left(s\right), a_{-i}\right)\right) \geq \overline{Q}_{i,h}\left(s, \left(a_i, a_{-i}\right)\right) + \iota, \forall\, h, s, i, a_{-i}, a_i \neq \pi_{i,h}^{\dagger}\left(s\right), \tag{151}$$

$$r_{i,h}^{\dagger,(k)} \in \left[-b, b\right], \forall\, h, k, i. \tag{152}$$

Consider the following attack,

$$r_{i,h}^{\dagger,(k)} = b \mathbb{1}_{\left\{a_{i,h}^{(k)}=\pi_{i,h}^{\dagger}\left(s^{(k)}\right)\right\}} - b \mathbb{1}_{\left\{a_{i,h}^{(k)}\neq\pi_{i,h}^{\dagger}\left(s^{(k)}\right)\right\}}, \forall\, h, k, i. \tag{153}$$

Given the attack, (148) implies,

$$\begin{aligned} R_{i,h}^{\dagger}\left(s, \boldsymbol{a}\right) &= \frac{1}{N_h\left(s, \boldsymbol{a}\right)} \sum_{k=1}^{K} r_{i,h}^{\dagger,(k)} \mathbb{1}_{\left\{s_h^{(k)}=s, \boldsymbol{a}_h^{(k)}=\boldsymbol{a}\right\}} \\ &= b \mathbb{1}_{\left\{a_i=\pi_{i,h}^{\dagger}(s)\right\}} - b \mathbb{1}_{\left\{a_i\neq\pi_{i,h}^{\dagger}(s)\right\}}, \forall\, h, s, i, \boldsymbol{a}. \end{aligned} \tag{154}$$

Then (149) implies, in period H,

$$
\begin{aligned}
\underline{Q}_{i,H}\left(s,\boldsymbol{\pi}_h^{\dagger}(s)\right) &= \min_{R_{i,h}\in \mathrm{CI}_{i,H}^{R^{\dagger}}\left(s,\boldsymbol{\pi}_h^{\dagger}(s)\right)} R_{i,h} \\
&= R_{i,H}^{\dagger}\left(s,\boldsymbol{\pi}_H^{\dagger}(s)\right) - \rho_H^R\left(s,\boldsymbol{\pi}_H^{\dagger}(s)\right) \\
&\geq b - \frac{b-\frac{\iota}{2}}{(H+1)/2}, \forall\, s,i,
\end{aligned}
\tag{155}
$$

and for $h<H$, assume $\underline{Q}_{i,h+1}\left(s,\boldsymbol{\pi}_h^{\dagger}(s)\right) \geq (H-h)\left(b-\frac{b-\frac{\iota}{2}}{(H+1)/2}\right), \forall\, s,i,$

$$
\begin{aligned}
\underline{Q}_{i,h}\left(s,\boldsymbol{\pi}_h^{\dagger}(s)\right) &= \min_{R_{i,h}\in \mathrm{CI}_{i,H}^{R^{\dagger}}\left(s,\boldsymbol{\pi}_h^{\dagger}(s)\right)} R_{i,h} + \min_{P_h\in \mathrm{CI}_h^{P}\left(s,\boldsymbol{\pi}_h^{\dagger}(s)\right)} \sum_{s'\in\mathcal{S}} P_h(s')\,\underline{Q}_{i,h+1}\left(s',\boldsymbol{\pi}_{h+1}^{\dagger}(s')\right) \\
&= R_{i,h}^{\dagger}\left(s,\boldsymbol{\pi}_h^{\dagger}(s)\right) - \rho_h^R\left(s,\boldsymbol{\pi}_h^{\dagger}(s)\right) \\
&\quad + \min_{P_h\in \mathrm{CI}_h^{P}\left(s,\boldsymbol{\pi}_h^{\dagger}(s)\right)} \sum_{s'\in\mathcal{S}} P_h(s')\,\underline{Q}_{i,h+1}\left(s',\boldsymbol{\pi}_{h+1}^{\dagger}(s')\right) \\
&\geq b - \frac{b-\frac{\iota}{2}}{(H+1)/2} + \min_{P_h\in \mathrm{CI}_h^{P}\left(s,\boldsymbol{\pi}_h^{\dagger}(s)\right)} \sum_{s'\in\mathcal{S}} P_h(s')(H-h)\left(b-\frac{b-\frac{\iota}{2}}{(H+1)/2}\right) \\
&= (H-h+1)\left(b-\frac{b-\frac{\iota}{2}}{(H+1)/2}\right), \forall\, h,s,i.
\end{aligned}
\tag{156}
$$

Similarly, in period H, due to reward bound (152),

$$
\begin{aligned}
\overline{Q}_{i,H}\left(s,\boldsymbol{\pi}_H^{\dagger}(s)\right) &= \max_{R_{i,H}\in \mathrm{CI}_{i,H}^{R^{\dagger}}\left(s,\boldsymbol{\pi}_H^{\dagger}(s)\right)} R_{i,H} \\
&= \min\left\{b, b+\rho_H^R\left(s,\boldsymbol{\pi}_H^{\dagger}(s)\right)\right\} \\
&= b, \forall\, s,i,
\end{aligned}
\tag{157}
$$

and for $h<H$, assume $\overline{Q}_{i,h+1}\left(s,\boldsymbol{\pi}_h^{\dagger}(s)\right) = (H-h)\,b, \forall\; s,i$, using the reward bound (152) again, we have,

$$
\begin{aligned}
\overline{Q}_{i,h}\left(s,\boldsymbol{\pi}_h^{\dagger}(s)\right) &= \max_{R_{i,h}\in \mathrm{CI}_{i,h}^{R^{\dagger}}\left(s,\boldsymbol{\pi}_H^{\dagger}(s)\right)} R_{i,h} + \max_{P_h\in \mathrm{CI}_h^{P}\left(s,\boldsymbol{\pi}_h^{\dagger}(s)\right)} \sum_{s'\in\mathcal{S}} P_h(s')\,\overline{Q}_{i,h+1}\left(s',\boldsymbol{\pi}_{h+1}^{\dagger}(s')\right) \\
&= \min\left\{b, b+\rho_h^R\left(s,\boldsymbol{\pi}_h^{\dagger}(s)\right)\right\} + \max_{P_h\in \mathrm{CI}_h^{P}\left(s,\boldsymbol{\pi}_h^{\dagger}(s)\right)} \sum_{s'\in\mathcal{S}} P_h(s')\,\overline{Q}_{i,h+1}\left(s',\boldsymbol{\pi}_{h+1}^{\dagger}(s')\right) \\
&\geq b + \max_{P_h\in \mathrm{CI}_h^{P}\left(s,\boldsymbol{\pi}_h^{\dagger}(s)\right)} \sum_{s'\in\mathcal{S}} P_h(s')(H-h)\,b \\
&= (H-h+1)\,b,
\end{aligned}
\tag{158}
$$

On the other hand, (150) implies, in period H, $a_i \neq \pi_{i,H}^{\dagger}(s)$,

$$
\begin{aligned}
\overline{Q}_{i,H}(s,\boldsymbol{a}) &= \max_{R_{i,H}\in \mathrm{CI}_{i,H}^{R^{\dagger}}(s,\boldsymbol{a})} R_{i,H} \\
&= R_{i,H}^{\dagger}(s,\boldsymbol{a}) + \rho_H^R(s,\boldsymbol{a}) \\
&\leq -b + \frac{b-\frac{\iota}{2}}{(H+1)/2}, \forall\, s,i,
\end{aligned}
\tag{159}
$$

and for $h < H$,

$$\begin{aligned}\overline{Q}_{i,h}(s,\boldsymbol{a}) &= \max_{R_{i,h}\in \mathrm{CI}^{R^\dagger}_{i,h}(s,\boldsymbol{a})} R_{i,h} + \max_{P_h\in \mathrm{CI}^P_h(s,\boldsymbol{a})} \sum_{s'\in\mathcal{S}} P_h(s')\,\overline{Q}_{i,h+1}\left(s',\boldsymbol{\pi}^\dagger_{h+1}(s')\right)\\ &= R^\dagger_{i,h}(s,\boldsymbol{a}) + \rho^R_h(s,\boldsymbol{a}) + \max_{P_h\in \mathrm{CI}^P_h(s,\boldsymbol{a})} \sum_{s'\in\mathcal{S}} P_h(s')\,\overline{Q}_{i,h+1}\left(s',\boldsymbol{\pi}^\dagger_{h+1}(s')\right)\\ &\leq -b + \frac{b-\frac{\iota}{2}}{(H+1)/2} + \max_{P_h\in \mathrm{CI}^P_h(s,\boldsymbol{a})} \sum_{s'\in\mathcal{S}} P_h(s')(H-h)\,b\\ &= (H-h-1)\,b + \frac{b-\frac{\iota}{2}}{(H+1)/2}, \forall\, h,s,i.\end{aligned} \tag{160}$$

Then (151) is satisfied since,

$$\begin{aligned}\overline{Q}_{i,h}(s,(a_i,a_{-i})) + \iota &\leq (H-h-1)\,b + \frac{b-\frac{\iota}{2}}{(H+1)/2} + \iota\\ &= (H-h+1)\left(b - \frac{b-\frac{\iota}{2}}{(H+1)/2}\right) + (H-h+2)\frac{b-\frac{\iota}{2}}{(H+1)/2} - (2b-\iota)\\ &= (H-h+1)\left(b - \frac{b-\frac{\iota}{2}}{(H+1)/2}\right) + \left(\frac{H-h+2}{H+1} - 1\right)\frac{b-\frac{\iota}{2}}{1/2}, h\geq 1\\ &\leq (H-h+1)\left(b - \frac{b-\frac{\iota}{2}}{(H+1)/2}\right) + \left(\frac{H+1}{H+1} - 1\right)\frac{b-\frac{\iota}{2}}{1/2}\\ &= (H-h+1)\left(b - \frac{b-\frac{\iota}{2}}{(H+1)/2}\right)\\ &\leq \underline{Q}_{i,h}\left(s,\left(\pi^\dagger_{i,h}(s),a_{-i}\right)\right), \forall\, h,s,i,a_{-i}\neq\pi^\dagger_{-i,h}(s), a_i\neq \pi^\dagger_{i,h}(s).\end{aligned} \tag{161}$$

Finally, (152) holds by definition,

$$\begin{aligned} r^{\dagger,(k)}_{i,h} &= b\mathbb{1}_{\left\{a^{(k)}_{i,h}=\pi^\dagger_{i,h}\left(s^{(k)}\right)\right\}} - b\mathbb{1}_{\left\{a^{(k)}_{i,h}\neq\pi^\dagger_{i,h}\left(s^{(k)}\right)\right\}}\\ &\in[-b,b], \forall\, h,k,i.\end{aligned} \tag{162}$$

□

C.2 Section 5

Proof of lemma 2

Proof. Consider the stage game in period H, and a fixed state $s\in\mathcal{S}$, the dominant strategy constraints on $\boldsymbol{\pi}^\dagger_H$ imply,

$$R^\dagger_{i,H}\left(s,\left(\pi^\dagger_{i,H}(s),a_{-i}\right)\right) - \rho^R_H\left(s,\left(\pi^\dagger_{i,H}(s),a_{-i}\right)\right) \tag{163}$$

$$\geq R^\dagger_{i,H}(s,(a_i,a_{-i})) + \rho^R_H(s,(a_i,a_{-i})) + \iota, \forall\, a_{-i}, a_i\neq\pi^\dagger_{i,H}(s). \tag{164}$$

Given the original maximum likelihood estimate of the stage game before the attack $\hat{R}_{i,H}$, define $a^\star_i$ to be the best response in the original game (before attack) to a_{-i} if $\pi^\dagger_{i,H}(s)$ is not available,

$$a^\star_i(a_{-i}) = \operatorname*{argmax}_{a_i\neq\pi^\dagger_{i,H}(s)}\left\{\hat{R}_{i,H}(s,(a_i,a_{-i})) + \rho^R_H(s,(a_i,a_{-i})) + \iota\right\}. \tag{165}$$

We use the following short-hand notations,

$$a^{\star} := \left(a_i^{\star}\left(a_{-i}\right), a_{-i}\right), \tag{166}$$

$$a^{\dagger} := \left(\pi_{i,H}^{\dagger}\left(s\right), a_{-i}\right). \tag{167}$$

If,

$$\hat{R}_{i,H}\left(s, a^{\star}\right) + \rho_H^R\left(s, a^{\star}\right) - \hat{R}_{i,H}\left(s, a^{\dagger}\right) + \rho_H^R\left(s, a^{\dagger}\right) + \iota \leq 0, \tag{168}$$

we have,

$$d_{i,H}^{\iota}\left(s, a_{-i}\right) = 0, \tag{169}$$

then the following inequality holds,

$$\left|R_{i,H}^{\dagger}\left(s, a^{\dagger}\right) - \hat{R}_{i,H}\left(s, a^{\dagger}\right)\right| + \left|R_{i,H}^{\dagger}\left(s, a^{\star}\right) - \hat{R}_{i,H}\left(s, a^{\star}\right)\right| \geq 0 \tag{170}$$

$$= d_{i,H}^{\iota}\left(s, a_{-i}\right). \tag{171}$$

Otherwise, we have, due to triangle inequality,

$$\left|R_{i,H}^{\dagger}\left(s, a^{\dagger}\right) - \hat{R}_{i,H}\left(s, a^{\dagger}\right)\right| + \left|R_{i,H}^{\dagger}\left(s, a^{\star}\right) - \hat{R}_{i,H}\left(s, a^{\star}\right)\right| \tag{172}$$

$$\geq \left|R_{i,H}^{\dagger}\left(s, a^{\dagger}\right) - R_{i,H}^{\dagger}\left(s, a^{\star}\right) - \hat{R}_{i,H}\left(s, a^{\dagger}\right) + \hat{R}_{i,H}\left(s, a^{\star}\right)\right| \tag{173}$$

$$\geq R_{i,H}^{\dagger}\left(s, a^{\dagger}\right) - R_{i,H}^{\dagger}\left(s, a^{\star}\right) - \hat{R}_{i,H}\left(s, a^{\dagger}\right) + \hat{R}_{i,H}\left(s, a^{\star}\right) \tag{174}$$

$$\geq \hat{R}_{i,H}\left(s, a^{\star}\right) + \rho_H^R\left(s, a^{\star}\right) - \hat{R}_{i,H}\left(s, a^{\dagger}\right) + \rho_H^R\left(s, a^{\dagger}\right) + \iota \tag{175}$$

$$= d_{i,H}^{\iota}\left(s, a_{-i}\right). \tag{176}$$

Now consider any action profile $\boldsymbol{a}$, we have, by the triangle inequality and the definition of the maximum likelihood estimates,

$$\left|R_{i,H}^{\dagger}\left(s, \boldsymbol{a}\right) - \hat{R}_{i,H}\left(s, \boldsymbol{a}\right)\right| \tag{177}$$

$$= \left|\frac{1}{N_H\left(s, \boldsymbol{a}\right)} \sum_{k=1}^{K} \mathbb{1}_{\left\{s_H^{(k)}=s, \boldsymbol{a}_H^{(k)}=\boldsymbol{a}\right\}} r_{i,H}^{\dagger,(k)} - \frac{1}{N_H\left(s, \boldsymbol{a}\right)} \sum_{k=1}^{K} \mathbb{1}_{\left\{s_H^{(k)}=s, \boldsymbol{a}_H^{(k)}=\boldsymbol{a}\right\}} r_{i,H}^{0,(k)}\right| \tag{178}$$

$$= \frac{1}{N_H\left(s, \boldsymbol{a}\right)} \left|\sum_{k=1}^{K} \mathbb{1}_{\left\{s_H^{(k)}=s, \boldsymbol{a}_H^{(k)}=\boldsymbol{a}\right\}} \left(r_{i,H}^{\dagger,(k)} - r_{i,H}^{0,(k)}\right)\right| \tag{179}$$

$$\leq \frac{1}{N_H\left(s, \boldsymbol{a}\right)} \sum_{k=1}^{K} \mathbb{1}_{\left\{s_H^{(k)}=s, \boldsymbol{a}_H^{(k)}=\boldsymbol{a}\right\}} \left|r_{i,H}^{\dagger,(k)} - r_{i,H}^{0,(k)}\right|. \tag{180}$$

Therefore, the total cost,

$$\sum_{k=1}^{K} \sum_{h=1}^{H} \sum_{s \in \mathcal{S}} \sum_{i=1}^{n} \sum_{\boldsymbol{a} \in \mathcal{A}} \mathbb{1}_{\left\{s_h^{(k)}=s, \boldsymbol{a}_h^{(k)}=\boldsymbol{a}\right\}} \left|r_{i,h}^{\dagger,(k)} - r_{i,h}^{0,(k)}\right| \tag{181}$$

$$\geq \sum_{s \in \mathcal{S}} \sum_{i=1}^{n} \sum_{\boldsymbol{a} \in \mathcal{A}} \sum_{k=1}^{K} \mathbb{1}_{\left\{s_H^{(k)}=s, \boldsymbol{a}_H^{(k)}=\boldsymbol{a}\right\}} \left|r_{i,H}^{\dagger,(k)} - r_{i,H}^{0,(k)}\right| \tag{182}$$

$$\geq \sum_{s \in \mathcal{S}} \sum_{i=1}^{n} \sum_{\boldsymbol{a} \in \mathcal{A}} N_H\left(s, \boldsymbol{a}\right) \left|R_{i,H}^{\dagger}\left(s, \boldsymbol{a}\right) - \hat{R}_{i,H}\left(s, \boldsymbol{a}\right)\right| \tag{183}$$

$$\geq \underline{N}_H \sum_{s \in \mathcal{S}} \sum_{i=1}^{n} \sum_{\boldsymbol{a} \in \mathcal{A}} \left|R_{i,H}^{\dagger}\left(s, \boldsymbol{a}\right) - \hat{R}_{i,H}\left(s, \boldsymbol{a}\right)\right| \tag{184}$$

$$\geq \underline{N}_H \sum_{s \in \mathcal{S}} \sum_{i=1}^{n} \sum_{a_{-i} \in \mathcal{A}_{-i}} \left|R_{i,H}^{\dagger}\left(s, \left(a_i^{\star}\left(a_{-i}\right), a_{-i}\right)\right) - \hat{R}_{i,H}\left(s, \left(a_i^{\star}\left(a_{-i}\right), a_{-i}\right)\right)\right| \tag{185}$$

$$\geq \underline{N}_H \sum_{s\in\mathcal{S}} \sum_{i=1}^{n} \sum_{a_{-i}\in\mathcal{A}_{-i}} \left\{ \left| R^{\dagger}_{i,H}\left(s,a^{\dagger}\right) - \hat{R}_{i,H}\left(s,a^{\dagger}\right) \right| + \left| R^{\dagger}_{i,H}\left(s,a^{\star}\right) - \hat{R}_{i,H}\left(s,a^{\star}\right) \right| \right\} \tag{186}$$

$$\geq \underline{N}_H \sum_{s\in\mathcal{S}} \sum_{i=1}^{n} \sum_{a_{-i}\in\mathcal{A}_{-i}} d^{\iota}_{i,H}\left(s,a_{-i}\right). \tag{187}$$

$\square$

Proof of Lemma 3

Proof. Fix s, h, i, a_i, and some a_{-i}. As in the last proof, we let

$$a^{\dagger} := \left(\pi^{\dagger}_{i,h}\left(s\right), a_{-i}\right).$$

Since these parameters are fixed, we use the shorthand $\underline{u}_{s'}(\boldsymbol{a})$ to refer to $[\underline{u}_{i,h}(s,\boldsymbol{a})]_{s'}$ and similarly for all the other slack variables.We focus on the constraints in the LP-formulation for the problem. Recall that,

$$\underline{Q}_{i,h}\left(s,\boldsymbol{a}\right) = \begin{array}{c} R^{\dagger}_{i,h}(s,\boldsymbol{a}) - \rho^R_h(s,\boldsymbol{a}) - \sum_{s'\in\mathcal{S}} \hat{P}_h\left(s'|s,\boldsymbol{a}\right)\left(\underline{u}_{s'}\left(\boldsymbol{a}\right) - \underline{v}_{s'}\left(\boldsymbol{a}\right)\right) \\ -\left\{ \sum_{s'\in\mathcal{S}} \rho^P_h\left(s,\boldsymbol{a}\right)\left(\underline{u}_{s'}\left(\boldsymbol{a}\right) + \underline{v}_{s'}\left(\boldsymbol{a}\right)\right) \right\} - \underline{w}(\boldsymbol{a}) \end{array}, \forall\ \boldsymbol{a}$$

$$\overline{Q}_{i,h}\left(s,\boldsymbol{a}\right) = \begin{array}{c} R^{\dagger}_{i,h}(s,\boldsymbol{a}) + \rho^R_h(s,\boldsymbol{a}) + \sum_{s'\in\mathcal{S}} \hat{P}_h\left(s'|s,\boldsymbol{a}\right)\left(\overline{u}_{s'}\left(\boldsymbol{a}\right) - \overline{v}_{s'}\left(\boldsymbol{a}\right)\right) \\ +\left\{ \sum_{s'\in\mathcal{S}} \rho^P_h\left(s,\boldsymbol{a}\right)\left(\overline{u}_{s'}\left(\boldsymbol{a}\right) + \overline{v}_{s'}\left(\boldsymbol{a}\right)\right) \right\} + \overline{w}(\boldsymbol{a}) \end{array}, \forall\ \boldsymbol{a}$$

We first bound $\underline{Q}_{i,h}(s,\boldsymbol{a})$. Notice that we can write $\underline{w}(\boldsymbol{a})$ as $\sum_{s'\in\mathcal{S}} \hat{P}_h(s' \mid s,\boldsymbol{a})\underline{w}(\boldsymbol{a})$ since $\hat{P}_h(\cdot \mid s,\boldsymbol{a})$ is a probability distribution. Thus, we can combine the first sum with the rewritten $\underline{w}(\boldsymbol{a})$ sum to get

$$-\sum_{s'\in\mathcal{S}} \hat{P}_h(s' \mid s,\boldsymbol{a})(\underline{u}_{s'}\left(\boldsymbol{a}\right) - \underline{v}_{s'}\left(\boldsymbol{a}\right) + \underline{w}(\boldsymbol{a}))$$

appearing in the definition of $\underline{Q}_{i,h}(s,\boldsymbol{a})$. Recall the following inequalities on the slack variables,

$$-\underline{Q}_{i,h+1}\left(s',\boldsymbol{\pi}^{\dagger}_{h+1}\left(s'\right)\right) \leq \underline{u}_{s'}\left(\boldsymbol{a}\right) - \underline{v}_{s'}\left(\boldsymbol{a}\right) + \underline{w}\left(\boldsymbol{a}\right), \forall\ s',\boldsymbol{a}$$
$$\overline{Q}_{i,h+1}\left(s',\boldsymbol{\pi}^{\dagger}_{h+1}\left(s'\right)\right) \leq \overline{u}_{s'}\left(\boldsymbol{a}\right) - \overline{v}_{s'}\left(\boldsymbol{a}\right) + \overline{w}\left(\boldsymbol{a}\right), \forall\ s',\boldsymbol{a}$$

Using the negative of the first inequality on the rewritten first sum, we have

$$-\sum_{s'\in\mathcal{S}} \hat{P}_h(s' \mid s,\boldsymbol{a})(\underline{u}_{s'}\left(\boldsymbol{a}\right) - \underline{v}_{s'}\left(\boldsymbol{a}\right) + \underline{w}(\boldsymbol{a})) \leq \sum_{s'\in S} \underline{Q}_{i,h+1}\left(s',\boldsymbol{\pi}^{\dagger}_{h+1}\left(s'\right)\right)$$

Applying this to the definition of $\underline{Q}_{i,h}(s,a^{\dagger})$ gives,

$$\underline{Q}_{i,h}(s,a^{\dagger}) \leq R^{\dagger}_{i,h}\left(s,a^{\dagger}\right) - \rho^R_h\left(s,a^{\dagger}\right) + \sum_{s'\in S} \hat{P}_h\left(s'|s,a^{\dagger}\right)\underline{Q}_{i,h+1}\left(s',\boldsymbol{\pi}^{\dagger}_{h+1}\left(s'\right)\right)$$
$$-\left\{ \sum_{s'\in\mathcal{S}} \rho^P_h\left(s,a^{\dagger}\right)\left(\underline{u}_{s'}\left(a^{\dagger}\right) + \underline{v}_{s'}\left(a^{\dagger}\right)\right) \right\}$$

$$\implies \underline{Q}_{i,h}(s,a^{\dagger}) \leq R^{\dagger}_{i,h}\left(s,a^{\dagger}\right) - \rho^R_h\left(s,a^{\dagger}\right) + \sum_{s'\in S} \frac{1}{|\mathcal{S}|}\underline{Q}_{i,h+1}\left(s',\boldsymbol{\pi}^{\dagger}_{h+1}\left(s'\right)\right)$$

as the slack variables other than $\underline{w}$ are non-negative as are the uncertainty parameters ρ_h^P. We also used the fact that the transitions are uniform and so $\hat{P}_h(s' \mid s, \boldsymbol{a}) = \frac{1}{|\mathcal{S}|}$ for all s', s, and $\boldsymbol{a}$.

Using the same approach on $\overline{Q}_{i,h}(s, \boldsymbol{a})$ gives the inequality,

$$\begin{aligned}\overline{Q}_{i,h}(s, \boldsymbol{a}) \geq R_{i,h}^{\dagger}(s, \boldsymbol{a}) + \rho_h^R(s, \boldsymbol{a}) + \sum_{s' \in \mathcal{S}} \hat{P}_h(s'|s, \boldsymbol{a}) \overline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}(s')\right) \\ + \left\{\sum_{s' \in \mathcal{S}} \rho_h^P(s, \boldsymbol{a}) \left(\overline{u}_{s'}(\boldsymbol{a}) + \overline{v}_{s'}(\boldsymbol{a})\right)\right\} \\ \implies \overline{Q}_{i,h}(s, \boldsymbol{a}) \geq R_{i,h}^{\dagger}(s, \boldsymbol{a}) + \rho_h^R(s, \boldsymbol{a}) + \sum_{s' \in \mathcal{S}} \frac{1}{|\mathcal{S}|} \overline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}(s')\right)\end{aligned}$$

Thus,

$$\begin{aligned}\overline{Q}_{i,h}(s, \boldsymbol{a}) - \underline{Q}_{i,h}(s, a^{\dagger}) &\geq R_{i,h}^{\dagger}(s, \boldsymbol{a}) - R_{i,h}^{\dagger}\left(s, a^{\dagger}\right) + \rho_h^R(s, \boldsymbol{a}) + \rho_h^R\left(s, a^{\dagger}\right) + \\ &\quad \sum_{s' \in \mathcal{S}} \frac{1}{|\mathcal{S}|}\left(\overline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}(s')\right) - \underline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}(s')\right)\right) \\ &\geq R_{i,h}^{\dagger}(s, \boldsymbol{a}) - R_{i,h}^{\dagger}\left(s, a^{\dagger}\right) + \rho_h^R(s, \boldsymbol{a}) + \rho_h^R\left(s, a^{\dagger}\right)\end{aligned}$$

The last inequality uses the fact that by definition, $\overline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}(s')\right) \geq \underline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}(s')\right)$ for all s'.

Recall, the ι-dominance constraint is

$$\overline{Q}_{i,h}(s, (a_i, a_{-i})) - \underline{Q}_{i,h}\left(s, \left(\pi_{i,h}^{\dagger}(s), a_{-i}\right)\right) \leq -\iota$$

Combining this with the derived bound on the differences of the Q's gives,

$$R_{i,h}^{\dagger}(s, \boldsymbol{a}) - R_{i,h}^{\dagger}\left(s, a^{\dagger}\right) + \rho_h^R(s, \boldsymbol{a}) + \rho_h^R\left(s, a^{\dagger}\right) \leq -\iota$$

which by rearranging implies that

$$R_{i,h}^{\dagger}\left(s, a^{\dagger}\right) - \rho_h^R\left(s, a^{\dagger}\right) \geq R_{i,h}^{\dagger}(s, \boldsymbol{a}) + \rho_h^R(s, \boldsymbol{a}) + \iota$$

As this is the only constraint used in the proof of 2, the proof of 2 carries over for each period h. Then, summing over the lower bounds for each period h gives the total lower-bound.

□

Proof of Theorem 3

Proof. Consider the Markov game with the same stage game in all periods $h \in [H]$ in all states $s \in \mathcal{S}$, given by the following reward matrix,

$\mathcal{A}_1 \setminus \mathcal{A}_2$	1	2	...	$\mathcal{A}_2$
1	$-b, -b$	$-b, b$	...	$-b, b$
2	$b, -b$	b, b	...	b, b
...	...	...	...	...
$\mathcal{A}_1$	$b, -b$	b, b	...	b, b

with uniform transitions, meaning,

$$\hat{P}_h(s'|s, \boldsymbol{a}) = \frac{1}{|\mathcal{S}|}, \forall\, h \in [H], s', s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}. \tag{188}$$

Given $\rho_h^R(s, \boldsymbol{a}) = \rho$ for each $h \in [H], s \in \mathcal{S}$ and $\boldsymbol{a} \in \mathcal{A}$,

$$d_{i,h}^{\iota}(s, a_{-i}) = \max\left\{0, \max_{a_i \neq \pi_{i,h}^{\dagger}(s)} \left\{\begin{array}{l}\hat{R}_{i,h}(s, (a_i, a_{-i})) + \rho_h^R(s, (a_i, a_{-i})) \\ -\hat{R}_{i,h}\left(s, \left(\pi_{i,H}^{\dagger}(s), a_{-i}\right)\right) + \rho_h^R\left(s, \left(\pi_{i,H}^{\dagger}(s), a_{-i}\right)\right) + \iota\end{array}\right\}\right\} \tag{189}$$

$$= \max\left\{0, \max_{a_i \neq \pi_{i,h}^{\dagger}(s)} \{b + \rho - (-b) + \rho + \iota\}\right\} \tag{190}$$

$$= 2b + 2\rho + \iota, \tag{191}$$

and thus, by Lemma 3, assuming $|\mathcal{A}_i| = |\mathcal{A}_1|$ for each $i \in [n]$, we have,

$$\sum_{k=1}^{K}\sum_{h=1}^{H}\sum_{s\in\mathcal{S}}\sum_{i=1}^{n}\sum_{\boldsymbol{a}\in\mathcal{A}} \mathbb{1}_{\left\{s_h^{(k)}=s, \boldsymbol{a}_h^{(k)}=\boldsymbol{a}\right\}} \left|r_{i,h}^{\dagger,(k)} - r_{i,h}^{0,(k)}\right| \tag{192}$$

$$\geq \sum_{h=1}^{H} \underline{N}_h \sum_{s\in\mathcal{S}}\sum_{i=1}^{n}\sum_{a_{-i}\in\mathcal{A}_{-i}} d_{i,h}^{\iota}(s, a_{-i}) \tag{193}$$

$$= \sum_{h=1}^{H} \underline{N}_h \sum_{s\in\mathcal{S}}\sum_{i=1}^{n}\sum_{a_{-i}\in\mathcal{A}_{-i}} (2b + 2\rho + \iota) \tag{194}$$

$$\geq \sum_{h=1}^{H} \min_{h\in[H]} \underline{N}_h \sum_{s\in\mathcal{S}}\sum_{i=1}^{n}\sum_{a_{-i}\in\mathcal{A}_{-i}} (2b + 2\rho + \iota) \tag{195}$$

$$= H|\mathcal{S}|\underline{N}n|\mathcal{A}_1|^{n-1}(2b + 2\rho + \iota). \tag{196}$$

□

C.3 Section 6

Proof of Lemma 4:

Proof. Given a solution to the greedy backward-induction attack $\left\{r_{i,h}^{\dagger,(k)}(s)\right\}_{h\in[H], s\in\mathcal{S}, i\in[n]}$, we have, from the subproblems,

$$R_{i,h}^{\dagger}(s, \boldsymbol{a}) = \frac{1}{N_h(s, \boldsymbol{a})}\sum_{k=1}^{K} r_{i,h}^{\dagger,(k)} \mathbb{1}_{\left\{s_h^{(k)}=s, \boldsymbol{a}_h^{(k)}=\boldsymbol{a}\right\}}, \forall\, h, s, i, \boldsymbol{a} \tag{197}$$

$$\underline{Q}_{i,h}(s, \boldsymbol{a}) = \min_{\boldsymbol{R}_h \in \mathrm{CI}_h^{R^{\dagger}}(s,\boldsymbol{a})} \boldsymbol{R}_h + \mathbb{1}_{h<H} \min_{P_h \in \mathrm{CI}_h^{P}(s,\boldsymbol{a})} \sum_{s'\in\mathcal{S}} P_h(s')\, \underline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}(s')\right), \forall\, h, s, i, \boldsymbol{a} \tag{198}$$

$$\overline{Q}_{i,h}(s, \boldsymbol{a}) = \max_{\boldsymbol{R}_h \in \mathrm{CI}_h^{R^{\dagger}}(s,\boldsymbol{a})} \boldsymbol{R}_h + \mathbb{1}_{h<H} \max_{P_h \in \mathrm{CI}_h^{P}(s,\boldsymbol{a})} \sum_{s'\in\mathcal{S}} P_h(s')\, \overline{Q}_{i,h+1}\left(s', \boldsymbol{\pi}_{h+1}^{\dagger}(s')\right), \forall\, h, s, i, \boldsymbol{a} \tag{199}$$

which are identical to the constraints in the original attacker's problem. Therefore, we have the dominant strategy equilibrium condition satisfied due to the corresponding constraints in the subproblems,

$$\underline{Q}_{i,h}\left(s, \left(\pi_{i,h}^{\dagger}(s), a_{-i}\right)\right) \geq \overline{Q}_{i,h}(s, (a_i, a_{-i})) + \iota, \forall\, h, s, i, a_{-i}, a_i \neq \pi_{i,h}^{\dagger}(s) \tag{200}$$

The bound constraints also come from the subproblems.

$$r_{i,h}^{\dagger,(k)} \in [-b, b], \forall\, h, k, i. \tag{201}$$

Therefore, $\left\{r_{i,h}^{\dagger,(k)}(s)\right\}_{h\in[H], s\in\mathcal{S}, i\in[n]}$ satisfies all constraints from the original attacker's problem, implying its feasibility. □

Proof of Lemma 5:

Proof. Consider a game with $H = 2, \mathcal{S} = \{1, 2\}, n = 2, \mathcal{A}_1 = \mathcal{A}_2 = \{1, 2, 3\}$, and in period 1 in both states, the rewards are given by,

$\mathcal{A}_1 \setminus \mathcal{A}_2$	1	2	3
1	b, b	$-b, b-\iota$	$-b, b-\iota$
2	$b-\iota, -b$	b, b	b, b
3	$b-\iota, -b$	b, b	b, b

and in period 2 in both states, the rewards are given by,

$\mathcal{A}_1 \setminus \mathcal{A}_2$	1	2	3
1	ι, ι	$\iota, 0$	$\iota, 0$
2	$0, \iota$	$0, 0$	$0, 0$
3	$0, \iota$	$0, 0$	$0, 0$

Consider the transitions,

$$\mu(s_1 = 1) = 1, \tag{202}$$

and,

$$\mathbb{P}\{s_2 = 1 | s_1, \boldsymbol{a}_1 = (1, 1)\} = 1, \mathbb{P}\{s_2 = 2 | s_1, \boldsymbol{a}_1 \neq (1, 1)\} = 1, \tag{203}$$

with a full coverage data set associated with it with $N_1(s, \boldsymbol{a}) = 9$ for every $s \in \mathcal{S}$ and $\boldsymbol{a} \in \mathcal{A}$, $N_2(s = 1, \boldsymbol{a}) = 1$ for every $\boldsymbol{a} \in \mathcal{A}$, $N_2(s = 2, \boldsymbol{a}) = 8$ for every $\boldsymbol{a} \in \mathcal{A}$. In particular, this means, each possible two-period joint policy appears exactly once in the data set.

For an attacker whose target is $\boldsymbol{\pi}_h^{\dagger}(s) = (1, 1)$ for every $h \in [H]$ and $s \in \mathcal{S}$, and learners with $\rho_h^R(s, \boldsymbol{a}) = \rho_h^P(s, \boldsymbol{a}) = 0$ for each $h \in [H], s \in \mathcal{S}$ and $\boldsymbol{a} \in \mathcal{A}$, the backward induction attack would not change rewards in period 2, and change the period 1 rewards so that the resulting stage game is given by,

$\mathcal{A}_1 \setminus \mathcal{A}_2$	1	2	3
1	b, b	$b, b-\iota$	$b, b-\iota$
2	$b-\iota, b$	$b-\iota, b-\iota$	$b-\iota, b-\iota$
3	$b-\iota, b$	$b-\iota, b-\iota$	$b-\iota, b-\iota$

The total attack cost would be,

$$2(9)(4b + 4\iota), \tag{204}$$

where the 2 is for the two learners, 9 is for the nine copies of period 1 game, and $(4b + 4\iota)$ is due to the modification of the reward entries.
However, another attack, not necessarily optimal, that results in following stage games in period 1,

$\mathcal{A}_1 \setminus \mathcal{A}_2$	1	2	3
1	b, b	$2\iota, b-\iota$	$2\iota, b-\iota$
2	$b-\iota, 2\iota$	b, b	b, b
3	$b-\iota, 2\iota$	b, b	b, b

and the following game in period 2, state 1,

$\mathcal{A}_1 \setminus \mathcal{A}_2$	1	2	3
1	b, b	$\iota, 0$	$\iota, 0$
2	$0, \iota$	$0, 0$	$0, 0$
3	$0, \iota$	$0, 0$	$0, 0$

and leave the stage game in period, state 2 unchanged, would have a strictly smaller total attack cost,

$$2\left((9)\left(2b+4\iota\right)+\left(b-\iota\right)\right). \tag{205}$$

where the 2 is for the two learners, 9 is for the nine copies of period 1 game, and $(2b+4\iota)$ is the modification of the reward entries in period 1, $(b-\iota)$ is the modification in period 2.
Note that even if we are allowed to directly modify the reward entries of the game (meaning each entry appears exactly once in the data set, $N_h\left(s,\boldsymbol{a}\right)=1$ for each $h, s \in \{1,2\}$, which is impossible in an actual full coverage data set due to the transition structure), the greedy attack incurs a cost of,

$$2\left(4b+4\iota\right), \tag{206}$$

whereas the alternative attack incurs a smaller cost of,

$$2\left(3b+3\iota\right). \tag{207}$$

Therefore, the backward induction attack can be suboptimal for the original attacker's problem. □

D Implementation Details and Code

The only package dependencies of note are numpy and scipy. The code assumes python3. The code can be run as is without any additional system arguments. Which experiment's output is returned can be changed by changing the value of the variable `example` in the main method.