Attacks and Defense on Normal-Form Games and Markov Games

by

Young Wu

submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Science)

at the

UNIVERSITY OF WISCONSIN–MADISON

12/22/2023

Date of final oral examination: 12/08/2023

The dissertation is approved by the following members of the Final Oral Committee:

Xiaojin (Jerry) Zhu, Professor, Computer Sciences
Qiaomin Xie, Assistant Professor, Industrial and Systems Engineering
Yudong Chen, Associate Professor, Computer Sciences
Josiah Hanna, Assistant Professor, Computer Sciences

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Thesis Statement

There are vulnerabilities in multi-agent systems and attackers can influence the behavior of players of normal-form games or multi-agent reinforcement learners through data or environment poisoning.

Introduction

There is a history of adversarial learning, but the vast majority focuses on supervised learning. As outlined in Wang et al. (2019a); Barreno et al. (2010), attack can occur in the training phase (causative attacks (Dalvi et al., 2004; Wittel and Wu, 2004; Adler, 2005; Biggio et al., 2011)), in which the attacker modifies the training data to mislead the victims to learn an incorrect model, and in the test phase (exploratory attacks (Szegedy et al., 2013; Tramèr et al., 2017; Moosavi-Dezfooli et al., 2017; Carlini and Wagner, 2017)), in which the attacker creates adversarial examples that lead to incorrect predictions based on the victims' model. Such attacks can be performed for unsupervised learning (Biggio et al., 2013, 2014) and active learning (Zhao et al., 2012) as well.

There are relatively fewer research on adversarial reinforcement learning, in which an attacker disrupts sequential decision making in the training phase or the test phase. In training-time attacks, an attacker either modifies the training dataset or the training environment (which we will call these attacks planning attacks). In reinforcement learning, the environment is usually modeled as a Markov decision process, and learning can occur either online, where the agents obtain state and reward data through interaction with the environment, or offline, where the agents are directly given a dataset containing state and reward information based on some behavioral policy. In test-time attacks, an attacker manipulates

the state, action, and reward when the trained policies of the agents are deployed. We provide a more detailed literature review of adversarial reinforcement learning in the next chapter.

However, there are vulnerabilities in multi-agent systems that are not well-addressed in the literature. Understanding adversarial attacks in the multi-agent setting is critical since many real-life applications are susceptible to these attacks. Some of these real-life applications of multi-agent reinforcement learning include board games such as GO and Chess (Silver et al., 2017, 2016), competitive robotics (Gu et al., 2017; Riedmiller et al., 2009; Kober et al., 2013), finance applications, especially algorithmic or high-frequency stock or option trading (Lee et al., 2007; Lee and O, 2002), video games (Vinyals et al., 2019; Jaderberg et al., 2019; Berner et al., 2019), card games (Brown and Sandholm, 2019; Brown et al., 2017), autonomous driving (Shalev-Shwartz et al., 2016), automated warehouses (Yang et al., 2020), and economic policymaking. For all of these applications, the decision makers are vulnerable to adversarial attacks, including manipulation of the environment or poisoning of the training datasets.

The training-time adversarial attack methods on single-agent reinforcement learning do not apply directly in the multi-agent setting where the environment is usually modeled by a normal-form game or a Markov game. Instead of manipulating the optimal policies in single-agent reinforcement learning, in the multi-agent setting, there are no optimal policies, so the attacker has to manipulate equilibrium policies, for example, Nash equilibrium in normal-form games or Markov perfect equilibrium in Markov games. This requires new attack algorithms to compute the optimal attack, where optimality is based on measures such as the costs of manipulating the environment or the data. All of our work in this document focuses on training-time attacks on multi-agent reinforcement learning, and similar to single-agent attacks, we categorize the attack problem as planning, online, or offline. In all three settings, there is an attacker with a target policy

which the attacker would like the victims to learn. For normal-form games, the target can be a pure-strategy action profile or a mixed-strategy one, and for Markov games, the target can be a deterministic Markov policy or a stochastic one. In Chapter 3, we investigate the problem in the online setting, where the attacker adaptively and minimally modifies the rewards given to the victims during online learning. In Chapters 4 and 5, we study the data poisoning problem in the offline setting, where the attacker minimally modifies the offline training data to install the target policy as the unique dominant strategy equilibrium for general-sum games (Chapter 4) or the unique Nash equilibrium for zero-sum games (Chapter 5). In Chapter 6, we investigate the problem in the planning setting, the attacker tries to minimally manipulate the game environment, in particular, the reward structure, to install a possibly mixed or stochastic policy.

2 RELATED WORK

Single-Agent Planning Setting Adversarial Attacks

The adversarial attack problem in reinforcement learning has been studied in the planning setting as an application of inverse reinforcement learning (Choi and Kim, 2011; Lin et al., 2014; Ng et al., 2000) and policy teaching (Banihashem et al., 2022; Rakhsha et al., 2020, 2021b,a). In this setting, a single agent (the victim) is given a Markov decision process (S, A, R, P), where S is the set of states, A is the set of actions, R is the rewards and P is the transitions, and the victim computes the optimal policy based on the given the Markov decision process (there is no training data in this setting). An attacker has the power to change the original (R^o, P^o) to $\left(R^\dagger, P^\dagger\right)$ before giving it to the victim, and would like to do so with minimal modification and in a way that the victim will find some target policy $\pi^\dagger$ as the optimal policy,

$$\begin{aligned} &\min_{R^\dagger, P^\dagger} C\left(R^\dagger, P^\dagger, R^o, P^o\right) \\ &\text{s.t. } \left(R^\dagger, P^\dagger\right) \text{ has optimal policy } \pi^\dagger. \end{aligned}$$

Instead of installing a single target policy as the optimal policy (Liu and Lai, 2021; Rakhsha et al., 2020, 2021b,a; Sun et al., 2020b), there are other models where the attacker tries to mislead the victims to use one of multiple targets (Banihashem et al., 2022) or minimize victim reward (Huang and Zhu, 2019; Sun et al., 2020b). If only the rewards can be modified, and the transitions P^o cannot be changed, the problem is called a reward poisoning problem.

In the single-agent setting, there always exists an optimal policy that is deterministic. This is not true in the multi-agent setting. There are normal-form games where the unique Nash equilibrium is completely mixed and Markov games where the unique Markov perfect equilibrium is

stochastic. This makes the reward poisoning problem more difficult in the multi-agent setting, where the attacker has the following problem instead,

$$\begin{aligned}&\min_{R^\dagger} C\left(R^\dagger, R^o\right)\\&\text{s.t. } \left(R^\dagger, P^o\right) \text{ has the unique equilibrium policy } \pi^\dagger.\end{aligned}$$

Chapter 6 studies the problem of installing a stochastic target equilibrium policy as the unique equilibrium in the planning setting by providing a characterization of the uniqueness of Nash equilibrium or Markov perfect equilibrium that can be used as the constraints in the attacker's problem.

Single-Agent Offline Adversial Attacks

Data poisoning problem in reinforcement learning has also been studied in the offline setting (Huang and Zhu, 2019; Liu and Lai, 2021; Ma et al., 2019), where the victim estimates a Markov decision process and computes its optimal policy $\pi^\star$ based on an offline dataset containing states, actions, and rewards obtained by some behavior policy, for example, in the form,

$$D = \{(s^o, a^o, r^o, s'^o)\},$$

where r is the reward from choosing action a in state s, and the state transitions to s' after the action is performed.
An attacker then comes in and minimally modifies the dataset to,

$$D^\dagger = \left\{\left(s^\dagger, a^\dagger, r^\dagger, s'^\dagger\right)\right\},$$

in a way that the victim will learn the optimal policy $\pi^\dagger$ based on $D^\dagger$.

Reward poisoning is a special type of data poisoning where only rewards r^o can be manipulated (Banihashem et al., 2022; Huang and Zhu, 2019; Rakhsha et al., 2021b), and there are other settings where the attacker can also change s^o and a^o (Liu and Lai, 2021; Rakhsha et al., 2021a, 2020;

Zhang et al., 2020b; Rangi et al., 2022b). The reward poisoning problem can be written in a form similar to,

$$\min_{r^\dagger} C\left(r^\dagger, r^o\right)$$
$$\text{s.t.} \hat{R}\left(r^\dagger\right) \text{ has optimal policy } \pi^\dagger,$$

where C is a measure of the cost of modifying the rewards from r^o to $r^\dagger$ and $\hat{R}(r)$ is the expected reward estimated from dataset r.

In the case the $\hat{R}$ algorithm is known to the attacker, the attack is white box (Banihashem et al., 2022; Huang and Zhu, 2019; Liu and Lai, 2021; Rangi et al., 2022b; Zhang et al., 2020b), and if $\hat{R}$ is unknown, the attack is black box (Rakhsha et al., 2020, 2021a,b; Rangi et al., 2022b), and the attacker has to make assumptions about $\hat{R}$, for example, by constructing a set of plausible Markov decision processes based on upper and lower confidence bounds of the reward estimates and make sure all of them have the same optimal policy $\pi^\dagger$.

Chapters 4 and 5 study the black-box attack problem of installing a deterministic target equilibrium policy for general-sum and zero-sum games in this setting, by providing a characterization of the uniqueness of a dominant strategy or Nash equilibrium in the form of linear constraints.

Single-Agent Online Adversarial Attacks

Reward poisoning has also been studied in the online setting in which the attacker modifies the rewards during online learning (Rakhsha et al., 2020, 2021a,b; Rangi et al., 2022b; Sun et al., 2020b; Zhang and Parkes, 2008b; Zhang et al., 2009, 2020b). The attacker's reward poisoning problem can

usually be written in the form,

$$\min_{r^\dagger} \sum_t C\left(r_t^\dagger, r^o\right)$$
$$\text{s.t.} a_t\left(r^\dagger\right)(s) = \pi^\dagger(s) \text{ for all but sub-linear number of times ,}$$

where $a_t(r)(s)$ is the online learning algorithm used by the victims in state s given a history of realized rewards r.

If there are no states s, then the victim is learning a multi-armed bandit process and is usually assumed to use a no-regret bandit algorithm where the optimal action is used in all but a sub-linear number of times (Bogunovic et al., 2021; Garcelon et al., 2020; Guan et al., 2020; Jun et al., 2018; Liu and Shroff, 2019; Lu et al., 2021; Ma et al., 2018; Yang et al., 2021; Zuo, 2020). This assumption can be used to simplify the constraint, and a similar technique is applied to bandit games in Chapter 3.

Adversarial Attacks on Multi-Agent System

In the multi-agent setting, Gleave et al. (2019); Guo et al. (2021) study the poisoning attack on multi-agent reinforcement learners, assuming that the attacker controls one of the learners. We are not aware of prior work in adversarial attacks in the multi-agent setting where an external attacker directly manipulates the environment or the data, and a series of problems under this assumption is studied in Chapters 3 to 6.

In the end, our work is also related to the game theory literature, such as the characterization of Nash equilibrium uniqueness (Millham, 1972; Heuer, 1979; Quintas, 1988), and the mechanism design literature, such as k-implementation (Anderson et al., 2010; Monderer and Tennenholtz, 2004), and dynamic mechanism design (Bergemann and Välimäki, 2019; Pavan et al., 2014). The mechanism design problem has similar goals of installing a dominant strategy equilibrium or a Bayesian Nash equilibrium

by modifying the rewards from each action profile, but it differs from poisoning attacks mainly due to the existence of private types of victims that affect the victims' rewards but are unknown to the attacker. For example, in the multi-agent reinforcement learning setting, it could be that the attacker either cannot observe the state or is not allowed to modify the reward differently in different states and in this case, the state would be a private type of the victims. In the problems we study in Chapters 3 and 6, the attacker has full information about the states and transitions, so they could be considered simplified versions of the general mechanism design problem. In Chapter 7, we discuss potential future work that expands the data poisoning problem that better connects with the mechanism design literature.

3 ONLINE REWARD POISONING FOR BANDIT GAMES TO INSTALL A DOMINANT STRATEGY EQUILIBRIUM

Contribution Statement. This chapter is a joint work with Yuzhe Ma and Jerry Zhu. Yuzhe Ma is the main author. My contribution includes the design of the main algorithms 1 and 2 (but not the proofs) and the examples used in the experiments. The paper version of this chapter appears in IJCAI 2021.

3.1 Introduction

In this chapter, we study the online attack problem in a general-sum bandit game environment, where a single attacker tries to minimally modify the rewards so that no-regret learners would use a deterministic target joint action in the majority of the rounds. We formulate the attacker's game redesign problem as an optimization problem, and we propose a feasible solution to the problem.

Consider a finite normal-form game with loss function ℓ^o. This is the "original game." As an example, the Volunteer's Dilemma (see Table 3.1) has each player choose whether or not to volunteer for a cause that benefits all players. It is known that all pure Nash equilibria in this game involve a subset of the players free-riding the contribution from the remaining players. M players, who initially do not know ℓ^o, use no-regret algorithms to individually choose their action in each of the $t = 1 \dots T$ rounds. The players receive limited feedback: suppose the chosen action profile in round t is $a^t = (a_1^t, \dots, a_M^t)$, then the i-th player only receives her own loss $\ell_i^o(a^t)$ but not the other players' actions or losses.

Game redesign is the following task. A game designer – not a player – does not like the solution concept to ℓ^o. Instead, the designer wants to

incentivize a target action profile $a^\dagger$, for example "every player volunteers". The designer has the power to redesign the game: before each round t is played, the designer can change ℓ^o to some ℓ^t. The players will receive new losses $\ell_i^t(a^t)$, but the designer pays a design cost $C(\ell^o, \ell^t, a^t)$ in that round for deviating from ℓ^o. The designer's goal is to make the players play the target action profile $a^\dagger$ in the vast majority $(T - o(T))$ of rounds, while incurring $o(T)$ cumulative design cost. Game redesign naturally emerges in two opposing contexts:

- A benevolent designer (interested party) wants to redesign the game to improve social welfare, as in the Volunteer's Dilemma. This is the motivation behind k-implementation Monderer and Tennenholtz (2004);
- A malicious designer (attacker) wants to poison the payoffs to force a nefarious target action profile. This is an extension of reward-poisoning attacks (previously studied on bandits Jun et al. (2018); Liu and Shroff (2019); Ma et al. (2018); Yang et al. (2021); Guan et al. (2020); Garcelon et al. (2020); Bogunovic et al. (2021); Zuo (2020); Lu et al. (2021) and reinforcement learning Zhang et al. (2020b); Ma et al. (2019); Rakhsha et al. (2020); Sun et al. (2020b); Huang and Zhu (2019)) to game playing.

For both contexts the mathematical question is the same. Since the design costs are measured by deviations from the original game ℓ^o, the designer is not totally free in creating new games. Our idea for successful game redesign is:

1. Do not change the loss of the target action profile, i.e. let $\ell^t(a^\dagger) = \ell^o(a^\dagger), \forall t$. If game redesign is indeed successful, then $a^\dagger$ will be played for $T - o(T)$ rounds. As we will see, $\ell^t(a^\dagger) = \ell^o(a^\dagger)$ means there is no design cost in those rounds under our definition of C. The remaining rounds incur at most $o(T)$ cumulative design cost.

2. The target action profile $a^\dagger$ forms a strictly dominant strategy equilibrium. This ensures no-regret players will eventually learn to prefer $a^\dagger$ over any other action profiles.

Game redesign is closely related to the k-implementation problem Monderer and Tennenholtz (2004). Both aim to manipulate player behaviors by changing the payoff. However, there are major differences: k-implementation assumes players know the game, while in our case the players have to learn the game; k-implementation only allows increase to existing payoffs, while we allow both positive (subsidy) and negative (tax) changes. Our interior design (Algorithm 1) indeed produces a 0-implementation in their terminology because we keep the payoff of the desired strategy profile unchanged. Nonetheless, our players have to discover this strategy profile by exploration, meaning that the designer will still incur costs especially in earlier rounds.

More broadly, game redesign is related to, but distinct from, constrained mechanism design. The players in game redesign are no-regret learners, not rational (best-response) players of a repeated game.

3.2 Formal Definition

We first describe the original game without the designer. There are M players. Let $\mathcal{A}_i$ be the finite action space of player i, and let $A_i = |\mathcal{A}_i|$. The original game is defined by the loss function $\ell^o : \mathcal{A}_1 \times \ldots \mathcal{A}_M \mapsto \mathbb{R}^M$. The players do not know ℓ^o. Instead, we assume they play the game for T rounds using no-regret algorithms. This may be the case, for example, if the players are learning an approximate Nash equilibrium in zero-sum ℓ^o or coarse correlated equilibrium in general sum ℓ^o. In running the no-regret algorithm, the players maintain their own action selection policies $\pi_i^t \in \Delta^{\mathcal{A}_i}$ over time, where $\Delta^{\mathcal{A}_i}$ is the probability simplex over $\mathcal{A}_i$. In each round t, every player i samples an action a_i^t according to policy π_i^t. This

forms an action profile $a^t = (a_1^t, \ldots, a_M^t)$. The original game produces the loss vector $\ell^o(a^t) = (\ell_1^o(a^t), ..., \ell_M^o(a^t))$. However, player i only observes her own loss value $\ell_i^o(a^t)$, not the other players' losses or their actions. All players then update their policy according to their no-regret algorithms.

We now bring in the designer. The designer knows ℓ^o and wants players to frequently play an arbitrary but fixed target action profile $a^\dagger$. We stress that $a^\dagger$ does not need to coincide with any solution concept in ℓ^o. At the beginning of round t, the designer commits to a potentially different loss function ℓ^t. Note this involves preparing the loss vector $\ell^t(a)$ for all action profiles a (i.e. "cells" in the payoff matrix). The players then choose their action profile a^t. Importantly, the players receive losses $\ell^t(a^t)$, not $\ell^o(a^t)$. For example, in games involving money such as the volunteer game, the designer may achieve $\ell^t(a^t)$ via taxes or subsidies, and in zero-sum games such as the rock-paper-scissors game, the designer essentially "makes up" a new outcome and tell each player whether they win, tie, or lose via $\ell_i^t(a^t)$; The designer incurs a cost $C(\ell^o, \ell^t, a^t)$ for deviating from ℓ^o. The interaction among the designer and the players is summarized as below.

Protocol: Game Redesign

Designer knows ℓ^o, $a^\dagger$, M, $\mathcal{A}_{1:M}$, and player no-regret rate α

for $t = 1, \ldots, T$ **do**

Designer prepares new loss function ℓ^t.

Players form action profile $a^t = (a_1^t, ..., a_M^t)$, where $a_i^t \sim \pi_i^t, \forall i \in [M]$.

Player i observes its loss $\ell_i^t(a^t)$, updates policy π_i^t.

Designer incurs cost $C(\ell^o, \ell^t, a^t)$.

The designer has two goals simultaneously:

1. To incentivize the players to frequently choose the target action profile $a^\dagger$ (which may not coincide with any solution concept of ℓ^o). Let $N^T(a) = \sum_{t=1}^T \mathbb{1}[a^t = a]$ be the number of times an action profile a is chosen in T rounds, then this goal is to achieve $\mathbf{E}\left[N^T(a^\dagger)\right] = T - o(T)$.

2. To have a small cumulative design cost $C^T := \sum_{t=1}^{T} C(\ell^o, \ell^t, a^t)$, specifically $\mathbf{E}\left[C^T\right] = o(T)$.

The per-round design cost $C(\ell^o, \ell^t, a)$ is application dependent. One plausible is to account for the **overall cost** in all action profiles, not just what is actually chosen: an example is $C(\ell^o, \ell^t, a^t) = \sum_a \|\ell^o(a) - \ell^t(a)\|_1$. Note that it ignores the a^t argument. In many applications, though, only the chosen action profile costs the designer (the **implementation cost** in Monderer and Tennenholtz (2004)). An example is $C(\ell^o, \ell^t, a^t) = \|\ell^o(a^t) - \ell^t(a^t)\|_1$. We use a slight generalization of the latter cost:

Assumption 3.1. *The non-negative designer cost function* C *satisfies* $\forall t, \forall a^t$, $C(\ell^o, \ell^t, a^t) \leqslant \eta \|\ell^o(a^t) - \ell^t(a^t)\|_p$ *for some Lipschitz constant* η *and norm* $p \geqslant 1$.

This implies no design cost if the losses are not modified, i.e., when $\ell^o(a^t) = \ell^t(a^t)$, $C(\ell^o, \ell^t, a^t) = 0$.

3.3 Assumption: No-Regret Players

The designer assumes that the players are each running a no-regret learning algorithm like EXP3.P Bubeck and Cesa-Bianchi (2012). It is well-known that for two-player ($M = 2$) zero-sum games, no-regret learners can find an approximate Nash Equilibrium Blum and Mansour (2007). More general results suggest that for multi-player ($M \geqslant 2$) general-sum games, no-regret learners can find an approximate Coarse Correlated Equilibrium Hart and Mas-Colell (2000). We first define the player's regret. We use a^t_{-i} to denote the actions selected by all players except player i in round t.

Definition 3.1. (*Regret*). *For any player* i*, the best-in-hindsight regret with respect to a sequence of loss functions* $\ell_i^t(\cdot, a_{-i}^t), t \in [T]$*, is defined as*

$$R_i^T = \sum_{t=1}^{T} \ell_i^t(a_i^t, a_{-i}^t) - \min_{a_i \in \mathcal{A}_i} \sum_{t=1}^{T} \ell_i^t(a_i, a_{-i}^t). \tag{3.1}$$

The expected regret is defined as $\mathbf{E}\left[R_i^T\right]$*, where the expectation is taken with respect to the randomness in the selection of actions* $a^t, t \in [T]$ *over all players.*

Remark 3.2. *The loss functions* $\ell_i^t(\cdot, a_{-i}^t), t \in [T]$ *depend on the actions selected by the other players* a_{-i}^t*, while* a_{-i}^t *further depends on* $a^1, ..., a^{t-1}$ *of all players in the first* $t-1$ *rounds. Therefore,* $\ell_i^t(\cdot, a_{-i}^t)$ *depends on* $a_i^1, ..., a_i^{t-1}$*. That means, from player* i*'s perspective, the player is faced with a non-oblivious* (*adaptive*) *adversary Slivkins* (*2019*).

Remark 3.3. *Note that* $a_i^* := argmin_{a_i \in \mathcal{A}_i} \sum_{t=1}^{T} \ell_i^t(a_i, a_{-i}^t)$ *in* (3.1) *would have meant a baseline in which player* i *always plays the best-in-hindsight action* a_i^* *in all rounds* $t \in [T]$*. Such baseline action should have caused all other players to change their plays away from* $a_{-i}^1, ..., a_{-i}^T$*. However, we are disregarding this fact in* (3.1)*. For this reason,* (3.1) *is not fully counterfactual, and is called the best-in-hindsight regret Bubeck and Cesa-Bianchi* (*2012*)*. The same is true when we define the expected regret.*

Our key assumption is that the learners achieve sublinear expected regret. This assumption is satisfied by standard bandit algorithms such as EXP3.P Bubeck and Cesa-Bianchi (2012).

Assumption 3.2. (*No-regret Learner*) *We assume the players apply no-regret learning algorithm that achieves expected regret* $\mathbf{E}\left[R_i^T\right] = O(T^\alpha), \forall i$ *for some* $\alpha \in [0, 1)$.

3.4 Game Redesign Algorithms

There is an important consideration regarding the allowed values of ℓ^t. The original game ℓ^o has a set of "natural loss values" $\mathcal{L}$. For example, in the rock-paper-scissors game $\mathcal{L} = \{-1, 0, 1\}$ for the player wins (recall the value is the loss), ties, and loses, respectively; while for games involving money it is often reasonable to assume $\mathcal{L}$ as some interval $[L, U]$. Ideally, ℓ^t should take values in $\mathcal{L}$ to match the semantics of the game or to avoid suspicion (in the attack context). Our designer can work with discrete $\mathcal{L}$ (section 3.4); but for exposition we will first allow ℓ^t to take real values in $\tilde{\mathcal{L}} = [L, U]$, where $L = \min_{x \in \mathcal{L}} x$ and $U = \max_{x \in L} x$. We assume U and L are the same for all players and $U > L$, which is satisfied when $\mathcal{L}$ contains at least two distinct values.

Algorithm: Interior Design

The name refers to the narrow applicability of Algorithm 1: the original loss values for the target action profile $\ell^o(a^\dagger)$ must all be in the interior of $\tilde{\mathcal{L}}$. Formally, we require $\exists \rho \in (0, \frac{U-L}{2}]$, $\forall i, \ell_i^o(a^\dagger) \in [L + \rho, U - \rho]$. In Algorithm 1, we present the interior design. The key insight is to keep $\ell^o(a^\dagger)$ unchanged: If the designer is successful, $a^\dagger$ will be played in $T - o(T)$ rounds. In these rounds, the designer cost is zero. The other $o(T)$ rounds each incur bounded cost. Overall, this ensures sublinear design cost. For the design to be successful, the designer can make $a^\dagger$ the strictly dominant strategy. The designer can do this by judiciously increasing or decreasing the loss of other action profiles in ℓ^o: there is enough room because $\ell^o(a^\dagger)$ is in the interior. In fact, the designer can design a time-invariant game $\ell^t = \ell$ as Algorithm 1 shows.

Lemma 3.4. *The redesigned game* (3.2) *satisfies:*

1. $\forall i, a, \ell_i(a) \in \tilde{\mathcal{L}}$, *thus* ℓ *is valid.*

Algorithm 1 Interior Design

Input: the target action profile $a^\dagger$; the original game ℓ^o.
Output: a time-invariant game ℓ constructed as follows:

$$\forall i, a, \ell_i(a) = \begin{cases} \ell_i^o(a^\dagger) - (1 - \frac{d(a)}{M})\rho & \text{if } a_i = a_i^\dagger, \\ \ell_i^o(a^\dagger) + \frac{d(a)}{M}\rho & \text{if } a_i \neq a_i^\dagger, \end{cases} \tag{3.2}$$

where $d(a) = \sum_{j=1}^{M} \mathbb{1}\left[a_j = a_j^\dagger\right]$.

2. *For every player i, the target action $a_i^\dagger$ strictly dominates any other action by $(1 - \frac{1}{M})\rho$, i.e., $\ell_i(a_i, a_{-i}) = \ell_i(a_i^\dagger, a_{-i}) + (1 - \frac{1}{M})\rho, \forall i, a_i \neq a_i^\dagger, a_{-i}$.*

3. *$\ell(a^\dagger) = \ell^o(a^\dagger)$.*

4. *If the original loss for the target action profile $\ell^o(a^\dagger)$ is zero-sum, then the redesigned game ℓ is also zero-sum.*

Our main result is that Algorithm 1 achieves the design goal with sublinear cumulative design cost. It is worth noting that although many entries in the redesigned game ℓ can appear to be quite different than the original game ℓ^o, their contribution to the design cost is small because the design discourages them from being played often.

Theorem 3.5. *Using Algorithm 1, the designer can achieve $\mathbf{E}\left[N^T(a^\dagger)\right] = T - O(MT^\alpha)$ while incurring expected cumulative design cost $\mathbf{E}\left[C^T\right] = O(\eta M^{1+\frac{1}{p}} T^\alpha)$.*

Corollary 3.6. *If the players use EXP3.P, the designer can achieve $\mathbf{E}\left[N^T(a^\dagger)\right] = T - O(MT^{\frac{1}{2}})$ while incurring expected cumulative design cost $\mathbf{E}\left[C^T\right] = O(\eta M^{1+\frac{1}{p}} T^{\frac{1}{2}})$.*

If the original game ℓ^o is two-player zero-sum, then under redesign, players will think that $a^\dagger$ is a Nash equilibrium.

Corollary 3.7. *Assume $M = 2$ and ℓ^o is zero-sum. Then with the redesigned game (3.2), the expected averaged policy $\mathbf{E}\left[\bar{\pi}_i^T\right] = \mathbf{E}\left[\frac{1}{T}\sum_t \pi_i^t\right]$ converges to a point mass on $a_i^\dagger$.*

Boundary Design

When $\ell^o(a^\dagger)$ has some values hitting the boundary of $\tilde{\mathcal{L}}$, the designer cannot apply Algorithm 1 directly because the loss of other action profiles cannot be increased or decreased further to make $a^\dagger$ a dominant strategy. However, a time-varying design can still achieve the design goals with sublinear design cost. In Algorithm 2, we present the boundary design which is applicable to both boundary and interior $\ell^o(a^\dagger)$ values.

Algorithm 2 Boundary Design

Input: the target action profile $a^\dagger$; a loss vector $v \in \mathbb{R}^M$ whose elements are in the interior, i.e., $\forall i, v_i \in [L+\rho, U-\rho]$ for some $\rho > 0$; the regret rate α; $\epsilon \in (0, 1-\alpha)$;

Output: a time-varying game with loss $\ell^t, t \in [T]$.

1: Use v in place of $\ell^o(a^\dagger)$ in (3.2) and apply the interior design 1. Call the resulting game the "source game" $\underline{\ell}$.

2: Define a "destination game" $\bar{\ell}$ where $\bar{\ell}(a) = \ell^o(a^\dagger), \forall a$.

3: Interpolate the source and destination games:

$$\ell^t = w_t \underline{\ell} + (1 - w_t)\bar{\ell} \tag{3.3}$$

where $w_t = t^{\alpha+\epsilon-1}$.

The designer can choose any loss vector v as long as v lies in the interior of $\tilde{\mathcal{L}}$. We give two exemplary choices of v.

1. Let the average player cost of $a^\dagger$ be $\bar{\ell}(a^\dagger) = \sum_{i=1}^M \ell_i^o(a^\dagger)/M$, then if $\bar{\ell}(a^\dagger) \in (L, U)$, one could choose v to be a constant vector with value $\bar{\ell}(a^\dagger)$. The nice property about this choice is that if ℓ^o is zero-sum, then v is zero-sum, thus property 4 is satisfied and the redesigned game is zero-sum. However, note that when $\bar{\ell}(a^\dagger)$ does hit the boundary, the designer cannot choose this v.

2. Choose v to be a constant vector with value $(L+U)/2$. This choice is always valid, but may not preserve the zero-sum property of the

original game unless $L = -U$.

The designer applies the interior design on ν to obtain a "source game" $\underline{\ell}$. Note that the target action profile $a^\dagger$ strictly dominates in the source game. The designer also creates a "destination game" $\overline{\ell}(a)$ by repeating the $\ell^o(a^\dagger)$ entry everywhere. The boundary algorithm then interpolates between the source and destination games with a decaying weight w_t. Note after interpolation (3.3), the target $a^\dagger$ still dominates by roughly w_t. We design the weight $w_t = t^{\alpha+\epsilon-1}$ so that cumulatively, the sum of w_t grows with rate $\alpha+\epsilon$, which is faster than the regret rate α. This is a critical consideration to enforce frequent play of $a^\dagger$. Also note that asymptotically, ℓ^t converges toward the destination game. Therefore, in the long run, when $a^\dagger$ is played the designer incurs diminishing cost, resulting in $o(T)$ cumulative design cost.

Lemma 3.8. *The redesigned game* (3.3) *satisfies:*

1. $\forall i, a, \ell_i^t(a) \in \tilde{\mathcal{L}}$, *thus the loss function is valid.*
2. *For every player* i, *the target action* $a_i^\dagger$ *strictly dominates any other action by* $(1-\frac{1}{M})\rho w_t$, *i.e.,* $\ell_i^t(a_i, a_{-i}) = \ell_i^t(a_i^\dagger, a_{-i}) + (1-\frac{1}{M})\rho w_t, \forall i, t, a_i \neq a_i^\dagger, a_{-i}$.
3. $\forall t, C(\ell^o, \ell^t, a^\dagger) \leqslant \eta(U-L)M^{\frac{1}{p}} w_t$
4. *If the original loss for the target action profile* $\ell^o(a^\dagger)$ *and the vector* ν *are both zero-sum, then* $\forall t, \ell^t$ *is zero-sum.*

Given Lemma 3.8, we provide our second main result.

Theorem 3.9. *Using Algorithm 2, the designer can achieve* $\mathbf{E}\left[N^T(a^\dagger)\right] = T - O(MT^{1-\epsilon})$ *while incurring expected cumulative design cost* $\mathbf{E}\left[C^T\right] = O(M^{1+\frac{1}{p}}T^{1-\epsilon} + M^{\frac{1}{p}}T^{\alpha+\epsilon})$.

Remark 3.10. *By choosing a larger ϵ in Theorem 3.9, the designer increases* $\mathbf{E}\left[N^T(a^\dagger)\right]$. *However, the design cost can grow. When* $\epsilon = \frac{1-\alpha}{2}$, *the design cost attains the minimum order* $O\left(T^{\frac{1+\alpha}{2}}\right)$ *and* $\mathbf{E}\left[N^T(a^\dagger)\right] = T - O(T^{\frac{1+\alpha}{2}})$

Corollary 3.11. *Assume the no-regret learning algorithm is EXP3.P. The designer can achieve expected number of target plays* $\mathbf{E}\left[N^T(a^\dagger)\right] = T - O(MT^{\frac{3}{4}})$ *while incurring* $\mathbf{E}\left[C^T\right] = O\left(M^{\frac{1}{P}}(1+M)T^{\frac{3}{4}}\right)$ *design cost.*

Discrete Design

In previous sections, we assumed the games ℓ^t can take arbitrary continuous values in the relaxed loss range $\tilde{\mathcal{L}} = [L, U]$. However, there are many real-world situations where continuous loss does not have a natural interpretation. For example, in the rock-paper-scissors game, the loss is interpreted as win, lose or tie, thus ℓ^t should only take value in the original loss value set $\mathcal{L} = \{-1, 0, 1\}$. We now provide a discrete redesign to convert any game ℓ^t with values in $\tilde{\mathcal{L}}$ into a game $\hat{\ell}^t$ only involving loss values L and U, which are both in $\mathcal{L}$. Specifically, the discrete design is illustrated in Algorithm 3.

Algorithm 3 Discrete Design

Input: the target action profile $a^\dagger$; a loss vector $\nu \in \mathbb{R}^M$ whose elements are in the interior, i.e., $\forall i, \nu_i \in [L + \rho, U - \rho]$ for some $\rho > 0$; the regret rate α; $\epsilon \in (0, 1 - \alpha)$;

Output: a time-varying game with loss $\hat{\ell}^t \in \mathcal{L}$ as below:

$$\forall t, i, a, \hat{\ell}_i^t(a) = \begin{cases} U & \text{with probability } \frac{\ell_i^t(a) - L}{U - L} \\ L & \text{with probability } \frac{U - \ell_i^t(a)}{U - L}. \end{cases} \tag{3.4}$$

It is easy to verify $\mathbf{E}\left[\hat{\ell}^t\right] = \ell^t$. In experiments we show such discrete games also achieve the design goals.

Thresholding the Redesigned Game

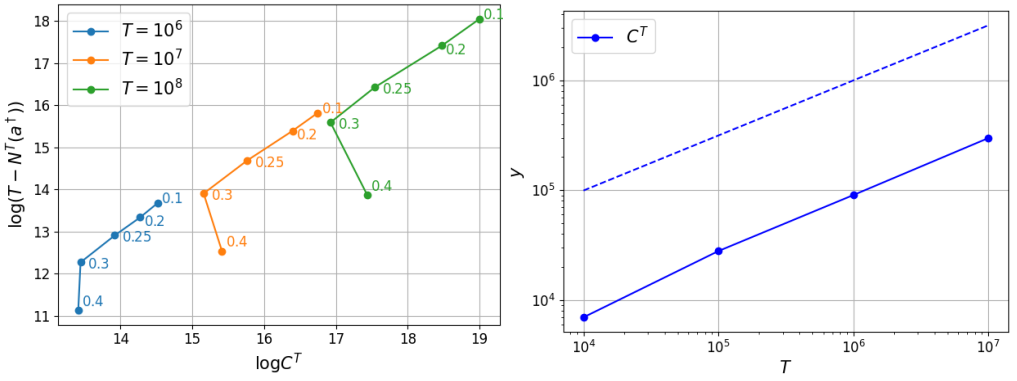
For all designs in previous sections, the designer could impose an additional min or max operator to threshold on the original game loss, e.g., for the interior design, the redesigned game loss after thresholding becomes $\forall i, a$,

$$\ell_i(a) = \begin{cases} \min\{\ell_i^o(a^\dagger) - (1 - \frac{d(a)}{M})\rho, \ell^o(a)\} & \text{if } a_i = a_i^\dagger, \\ \max\{\ell_i^o(a^\dagger) + \frac{d(a)}{M}\rho, \ell^o(a)\} & \text{if } a_i \neq a_i^\dagger. \end{cases} \tag{3.5}$$

We point out a few differences between (3.5) and (3.2). First, (3.5) guarantees a dominance gap of "at least" (instead of exactly) $(1 - \frac{1}{M})\rho$. As a result, the thresholded game can induce a larger $N^\intercal(a^\dagger)$ because the target action $a^\dagger$ is redesigned to stand out even more. Second, one can easily show that (3.5) incurs less design cost $C^\intercal$ compared to (3.2) due to thresholding. Therefore, Theorem 3.5 still holds. However, thresholding no longer preserves the zero-sum property.

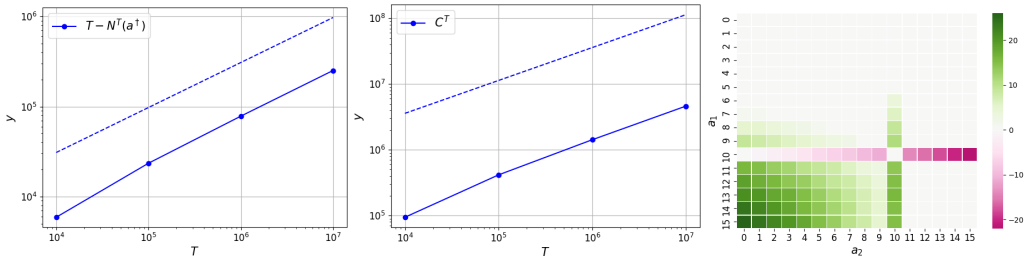
3.5 Experiments

We perform empirical evaluations of game redesign algorithms on four games — the volunteer's dilemma (VD), tragedy of the commons (TC), prisoner's dilemma (PD) and rock-paper-scissors (RPS). Throughout the experiments, we use EXP3.P Bubeck and Cesa-Bianchi (2012) as the no-regret learner. The concrete form of the regret bound for EXP3.P is illustrated in the appendix A.1. Based on that, we derive the exact form of our theoretical upper bounds for Theorem 3.5 and Theorem 3.9 (see (A.34)-(A.37)), and we show the theoretical value for comparison in our experiments. We let the designer cost function be $C(\ell^o, \ell^t, a^t) = \|\ell^o(a^t) - \ell^t(a^t)\|_p$ with $p = 1$. For VD, TC and PD, the original game is not zero-sum, and we apply the thresholding (3.5) to slightly improve the



(a) Total rounds $a^t \neq a^\dagger$. (b) Cumulative design cost.

Figure 3.1: Interior design on PD. The dashed line is the theoretical upper bound.



(a) Total rounds $a^t \neq a^\dagger$. (b) Cumulative design cost. (c) Loss change.

Figure 3.2: Interior design on TC. The dashed line is the theoretical upper bound. At $a^\dagger = (10, 10)$, the loss is unchanged.

redesign performance. For the RPS game, we apply the design without thresholding to preserve the zero-sum property. The results we show in all the plots are produced by taking the average of 5 trials.

Volunteer's Dilemma (VD)

In volunteer's dilemma (Table 3.1) there are M players. Each player has two actions: volunteer or not. When there exists at least one volunteer, those players who do not volunteer gain 1 (i.e. a -1 loss). The volunteers receive zero payoff. On the other hand, if no players volunteer, then every player loss 10.

		Other players: exists a volunteer	Other players: no volunteer exists
Player i	volunteer	0	0
	not volunteer	-1	10

Table 3.1: The loss function ℓ_i^o for individual player i in VD.

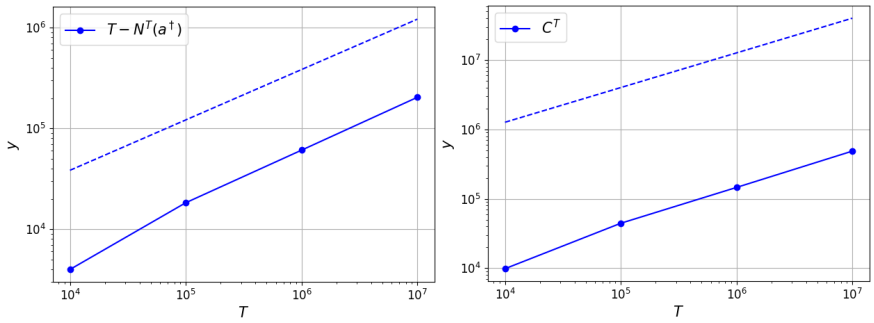
As mentioned earlier, all pure Nash equilibria involve free-riders. The designer aims at encouraging all players to volunteer, i.e., the target action profile $a_i^\dagger$ is "volunteer" for any player i. Note that $\forall i, \ell_i^o(a^\dagger) = 0$, which lies in the interior of $\mathcal{L} = [-1, 10]$. Therefore, the designer could apply the interior design Algorithm 1. The margin parameter is $\rho = 1$. We let $M = 3$. In table 3.2, we show the redesigned game ℓ. Note that when all three players volunteer (i.e., at $a^\dagger$), the loss is unchanged compared to ℓ^o. Furthermore, regardless of the other players, the action "volunteer" strictly dominates the action "not volunteer" by at least $(1 - \frac{1}{M})\rho = \frac{2}{3}$ for every player. When there is no other volunteers, the dominance gap is $\frac{32}{3} \geqslant (1 - \frac{1}{M})\rho$, which is due to the thresholding in (3.5). We simulated play for

		Number of other volunteers: 0	1	2
Player i	volunteer	$-2/3$	$-1/3$	0
	not volunteer	10	1/3	2/3

Table 3.2: The redesigned loss function ℓ_i for player i in VD.

$T = 10^4, 10^5, 10^6, 10^7$, respectively on this redesigned game ℓ. In Figure 3.3a, we show $T - N^T(a^\dagger)$ against T. The plot is in log scale. The standard deviation estimated from 5 trials is less than 3% of the corresponding value and is hard to see in log-scale plot, thus we do not show that. We also plot our theoretical upper bound in dashed lines for comparison. Note that the theoretical value indeed upper bounds our empirical results. In Figure 3.3b, we show C^T against T. Again, the theoretical upper bound holds. As our theory predicts, for the four T's the designer increasingly

enforces $a^\dagger$ in 60%, 82%, 94%, and 98% of the rounds, respectively; The per-round design costs C^T/T decreases at 0.98, 0.44, 0.15, and 0.05, respectively.



(a) Number of rounds with $a^t \neq a^\dagger$ grows sublinearly.

(b) The cumulative design cost grows sublinearly too.

Figure 3.3: Interior design on VD with $M = 3$. The dashed lines are theoretical upper bounds.

Tragedy of the Commons (TC)

Our second example is the tragedy of the commons (TC). There are $M = 2$ farmers who share the same pasture to graze sheep. Each farmer i is allowed to graze at most 15 sheep, i.e., the action space is $\mathcal{A}_i = \{0, 1, ..., 15\}$. The more sheep are grazed, the less well fed they are, and thus less price on market. We assume the price of each sheep is $p(a) = \sqrt{30 - \sum_{i=1}^{2} a_i}$, where a_i is the number of sheep that farmer i grazes. The loss function of farmer i is then $\ell_i^o(a) = -p(a)a_i$, i.e. negating the total price of the sheep that farmer i owns. The Nash equilibrium strategy of this game is that every farmer grazes $a_i^* = 12$ sheep.

It is well-known that this Nash equilibrium is suboptimal. Instead, the designer hopes to maximize social welfare: $p(a)(a_1 + a_2)$, which is achieved when $a_1 + a_2 = 20$. Moreover, to promote equity the designer desires that the two farmers graze the same number of sheep. Thus the target action profile is $a_i^\dagger = 10, \forall i$. Note that the original loss function takes value in $[-15\sqrt{15}, 0]$ while $\ell_i^o(a^\dagger) = -10\sqrt{10}$, thus this is the interior

	R	P	S
R	$-0.5, 0.5$	$0, 0$	$-0.5, 0.5$
P	$0, 0$	$0.5, -0.5$	$0, 0$
S	$0, 0$	$0.5, -0.5$	$0, 0$

(a) $\ell^t(t = 1)$.

	R	P	S
R	$0.62, -0.62$	$0.75, -0.75$	$0.62, -0.62$
P	$0.75, -0.75$	$0.87, -0.87$	$0.75, -0.75$
S	$0.75, -0.75$	$0.87, -0.87$	$0.75, -0.75$

(b) $\ell^t(t = 10^3)$.

	R	P	S
R	$0.94, -0.94$	$0.96, -0.96$	$0.94, -0.94$
P	$0.96, -0.96$	$0.98, -0.98$	$0.96, -0.96$
S	$0.96, -0.96$	$0.98, -0.98$	$0.96, -0.96$

(c) $\ell^t(t = 10^7)$.

Table 3.3: The redesigned RPS games ℓ^t for selected t (with $\epsilon = 0.3$). Note the target entry $a^\dagger = (R, P)$ converges toward $(1, -1)$.

design scenario. Due to the large number of entries, we only visualize the difference $\ell_1(a) - \ell_1^o(a)$ for player 1 in Figure 3.2c; the other player is the same. We observe three patterns of loss change. For most a's, e.g., $a_1 \leqslant 6$ or $a_2 \geqslant 11$, the original loss $\ell_1^o(a)$ is already sufficiently large and satisfies the dominance gap in Lemma 3.4, thus the loss remains unchanged. For those a's where $a_1^\dagger = 10$, the designer reduces the loss to make the target action more profitable. For those a's close to the bottom left ($a_1 > a_1^\dagger$ and $a_2 \leqslant 10$), the designer increases the loss to enforce the gap $(1 - \frac{1}{M})\rho$.

We simulated play for $T = 10^4, 10^5, 10^6$ and 10^7 and show the results in Figure 3.2. Again the game redesign is successful: the figures confirm $T - O(T)$ target action play and $o(T)$ cumulative design cost. Numerically, for the four T's the designer enforces $a^\dagger$ in 41%, 77%, 92%, and 98% of rounds, and the per-round design costs are 9.4, 4.2, 1.4, and 0.5.

Prisoner's Dilemma (PD)

Out third example is the prisoner's dilemma (PD). There are two prisoners, each can stay mum or fink. The original loss function ℓ^o is given

	R	P	S
R	$0,0$	$1,-1$	$-1,1$
P	$-1,1$	$0,0$	$1,-1$
S	$1,-1$	$-1,1$	$0,0$

(a) The original loss ℓ^o.

	R	P	S
R	$1,1$	$1,1$	$-1,1$
P	$-1,-1$	$1,-1$	$-1,-1$
S	$-1,1$	$-1,-1$	$-1,-1$

(b) $\hat{\ell}^t(t=1)$.

	R	P	S
R	$1,-1$	$1,1$	$-1,-1$
P	$1,-1$	$1,-1$	$1,-1$
S	$1,-1$	$1,-1$	$1,1$

(c) $\hat{\ell}^t(t=10^3)$.

	R	P	S
R	$1,-1$	$1,-1$	$1,-1$
P	$1,-1$	$1,-1$	$1,-1$
S	$1,-1$	$1,-1$	$1,-1$

(d) $\hat{\ell}^t(t=10^7)$.

Table 3.4: Instantiation of discrete design on the same games as in Table 3.3. The redesigned loss lies in $\mathcal{L}=\{-1,0,1\}$.

	mum	fink
mum	$2,2$	$5,1$
fink	$1,5$	$4,4$

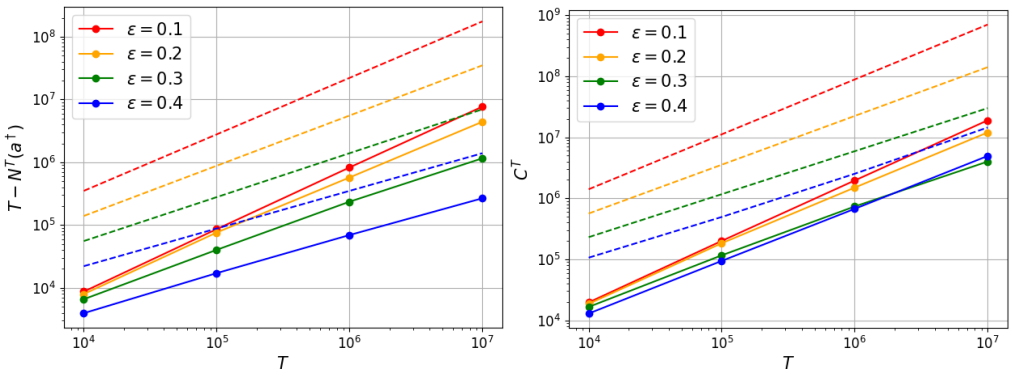
(a) The original loss ℓ^o of PD.

	mum	fink
mum	$2,2$	$1.5,2.5$
fink	$2.5,1.5$	$4,4$

(b) The redesigned loss ℓ of PD.

Table 3.5: Interior redesign on Prisoner's Dilemma.

in Table 3.5a. The Nash equilibrium strategy of this game is that both prisoners fink. Suppose a mafia designer hopes to force $a^\dagger$ =(mum, mum) by sabotaging the losses. Note that $\forall i, \ell_i^o(a^\dagger)=2$, which lies in the interior of the loss range $\mathcal{L}=[1,5]$. Therefore, this is again an interior design scenario. In Table 3.5b we show the redesigned game ℓ. Note that when both prisoners stay mum or both fink, the designer does not change the loss. On the other hand, when one prisoner stays mum and the other finks, the designer reduces the loss for the mum prisoner and increases the loss for the betrayer. We simulated plays for $T=10^4, 10^5, 10^6$, and 10^7, respectively. In Figure 3.1 we plot the number of non-target action selections $T-N^T(a^\dagger)$ and the cumulative design cost C^T. Both grow sublinearly. The designer enforces $a^\dagger$ in 85%, 94%, 98%, and 99% of rounds, and the per-round design costs are 0.71, 0.28, 0.09, and 0.03, respectively.



(a) Number of rounds $a^t \neq a^\dagger$. (b) The cumulative design cost.

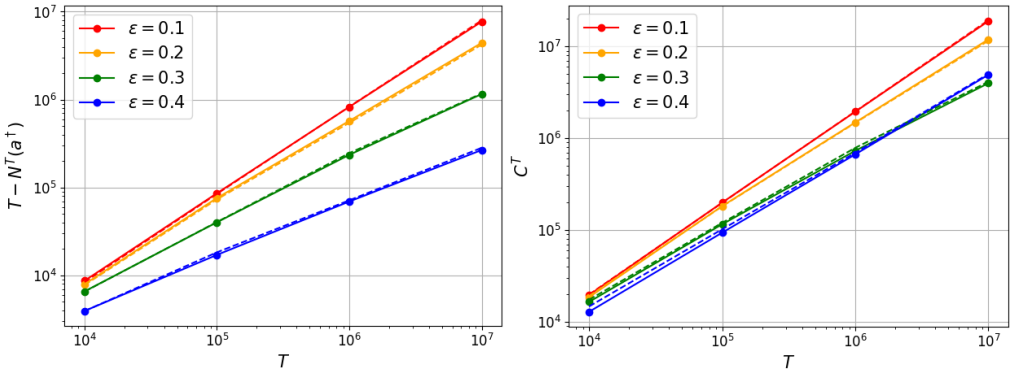
Figure 3.4: Boundary design on RPS. The dashed lines are the theoretical upper bound.

Rock-Paper-Scissors (RPS)

Our last example is the RPS game, as shown in Table 3.4a.

Boundary Design. Suppose the target profile is $a^\dagger = (R, P)$. Since $\ell^o(a^\dagger) = (1, -1)$ hits the boundary of loss range $\tilde{\mathcal{L}} = [-1, 1]$, the designer must use the boundary design. For simplicity we choose v with $v_i = \frac{U+L}{2}, \forall i$. This choice of v preserves the zero-sum property. Table 3.3 shows the redesigned games at $t = 1, 10^3$ and 10^7 under $\epsilon = 0.3$. Note that the designer maintains the zero-sum property of the games. Also note that the redesigned loss function guarantees strict dominance of $a^\dagger$ for all t, but the dominance gap decreases as t grows. Finally, the loss of the target action $a^\dagger = (R, P)$ converges to the original loss $\ell^o(a^\dagger) = (1, -1)$ asymptotically, thus the designer incurs diminishing cost.

We ran Algorithm 2 for $\epsilon = 0.1, 0.2, 0.3, 0.4$. For each ϵ we simulated game play for $T = 10^4, 10^5, 10^6$ and 10^7. In Figure 3.4a, we show $T - N^T(a^\dagger)$ under different ϵ (solid lines) and the theoretical upper bounds of Theorem 3.9 (dashed lines) for comparison. In Figure 3.4b, we show the cumulative design cost C^T and the upper bounds. Note that both $T - N^T(a^\dagger)$ and C^T grow sublinearly. For $\epsilon = 0.3$, for the four T's the designer forces $a^\dagger$ in 34%, 60%, 76%, and 88% rounds. The per-round



(a) Number of rounds $\alpha^t \neq \alpha^\dagger$. (b) The cumulative design cost.

Figure 3.5: Discrete redesign for $\alpha^\dagger = (R, P)$ with natural loss values in $\mathcal{L}$. The dashed lines are the corresponding boundary design.

design costs are 1.7, 1.2, 0.73 and 0.40. The results are similar for the other ϵ's. We note that empirically the cumulative design cost achieves the minimum at some $\epsilon \in (0.3, 0.4)$ while Theorem 3.9 suggests the minimum at $\epsilon^* = 0.25$. We investigate this inconsistency in appendix A.2.

Discrete Design. We now compare the performance of discrete design (Algorithm 3) with the boundary design. The target profile is still $\alpha^\dagger = (R, P)$. Recall the purpose of discrete design is to only use natural game loss values, in the RPS case $\mathcal{L} = \{-1, 0, 1\}$. Figure 3.5 shows that the performance of the discrete design nearly matches the boundary design. When $\epsilon = 0.3$, for the four T's discrete design enforces $\alpha^\dagger$ 35%, 59%,75% and 88% of the time. The per-round design costs are 1.7, 1.2, 0.79, and 0.41. Overall, discrete design does not lose much performance. Table 3.4 shows the redesigned "random" games at $t = 1, 10^3$ and 10^7 under $\epsilon = 0.3$. As t increases, the redesigned loss function converges to a constant function that takes the target loss value $\ell^o(\alpha^\dagger)$.

3.6 Conclusion

We studied the game redesign problem where players apply no-regret algorithms to play the game. We show that a designer can enforce a target action profile in $T - o(T)$ rounds while incurring $o(T)$ cumulative design cost. Experiments demonstrate the performance of our redesign algorithms.

In the next chapter, we investigate the extension of the problem on Markov games, and in the offline data poisoning setting. The approach to installing a dominant strategy equilibrium is similar to the one in this chapter, but we have to deal with the additional complication that the learners estimate the Markov game using confidence-based algorithms based on an offline dataset.

4 OFFLINE REWARD POISONING FOR GENERAL-SUM GAMES TO INSTALL A DOMINANT STRATEGY EQUILIBRIUM

Contribution Statement. This chapter is a joint work with Jeremy McMahan, Jerry Zhu, and Qiaomin Xie. I am the main author. My contribution includes the statements and proofs of the main propositions and theorems 1 to 3 and the writing of the paper. The paper version of this chapter appears in AAAI 2023.

4.1 Introduction

In this chapter, we study the offline data poisoning problem in a general-sum Markov game environment, where a single attacker tries to minimally modify the offline rewards so that learners that compute the Markov perfect dominant strategy equilibrium of a Markov game within some confidence region around the maximum likelihood estimate of the Markov game based on the offline rewards would find a deterministic target joint policy as the unique equilibrium. We formulate the attacker's reward poisoning problem as a linear program, which can be solved efficiently, and we provide sufficient conditions for the feasibility of such an attack.

Multi-agent reinforcement learning (MARL) has achieved tremendous empirical success across a variety of tasks such as autonomous driving, cooperative robotics, economic policy-making, and video games. In MARL, several agents interact with each other and the underlying environment, and each of them aims to optimize their individual long-term reward Zhang et al. (2021a). Such problems are often formulated under the framework of Markov Games Shapley (1953), which generalizes the Markov Decision Process model from single-agent RL. In offline MARL, the agents aim to learn a good policy by exploiting a pre-collected dataset without

further interactions with the environment or other agents Pan et al. (2022); Jiang and Lu (2021); Cui and Du (2022b); Zhong et al. (2022). The optimal solution in MARL typically involves equilibria concepts.

While the above empirical success is encouraging, MARL algorithms are susceptible to data poisoning attacks: the agents can reach the wrong equilibria if an exogenous attacker manipulates the feedback to agents. For example, a third-party attacker may want to interfere with traffic to cause autonomous vehicles to behave abnormally; teach robots an incorrect procedure so that they fail at certain tasks; misinform economic agents about the state of the economy and guide them to make irrational investments or saving decisions; or cause the non-player characters in a video game to behave improperly to benefit certain human players. In this paper, we study the security threat posed by reward-poisoning attacks on offline MARL. Here, the attacker wants the agents to learn a target policy $\pi^\dagger$ of the attacker's choosing ($\pi^\dagger$ does not need to be an equilibrium in the original Markov Game). Meanwhile, the attacker wants to minimize the amount of dataset manipulation to avoid detection and accruing high cost. This paper studies optimal offline MARL reward-poisoning attacks. Our work serves as a first step toward eventual defense against reward-poisoning attacks.

Our Contributions

We introduce reward-poisoning attacks in offline MARL. We show that any attack that reduces to attacking single-agent RL separately must be suboptimal. Consequently, new innovations are necessary to attack effectively. We present a reward-poisoning framework that guarantees the target policy $\pi^\dagger$ becomes a Markov Perfect Dominant Strategy Equilibrium (MPDSE) for the underlying Markov Game. Since any rational agent will follow an MPDSE if it exists, this ensures the agents adopt the target policy $\pi^\dagger$. We also show the attack can be efficiently constructed using a linear

program.

The attack framework has several important features. First, it is effective against a large class of offline MARL learners rather than a specific learning algorithm. Second, the framework allows partially decentralized agents who can only access their own individual rewards rather than the joint reward vectors of all agents. Lastly, the framework only makes the minimal assumption on the rationality of the learners that they will not take dominated actions.

We also give interpretable bounds on the minimal cost to poison an arbitrary dataset. These bounds relate the minimal attack cost to the structure of the underlying Markov Game. Using these bounds, we derive classes of games that are especially cheap or expensive for the attacker to poison. These results show which games may be more susceptible to an attacker, while also giving insight to the structure of multi-agent attacks.

In the right hands, our framework could be used by a benevolent entity to coordinate agents in a way that improves social welfare. However, a malicious attacker could exploit the framework to harm learners and only benefit themselves. Consequently, our work paves the way for future study of MARL defense algorithms.

Related Work

Online Reward-Poisoning: Reward poisoning problem has been studied in various settings, including online single-agent reinforcement learners Banihashem et al. (2022); Huang and Zhu (2019); Liu and Lai (2021); Rakhsha et al. (2021a,b, 2020); Sun et al. (2020b); Zhang et al. (2020b), as well as online bandits Bogunovic et al. (2021); Garcelon et al. (2020); Guan et al. (2020); Jun et al. (2018); Liu and Shroff (2019); Lu et al. (2021); Ma et al. (2018); Yang et al. (2021); Zuo (2020). Online reward poisoning for multiple learners is recently studied as a game redesign problem in Ma et al. (2021).

Offline Reward Poisoning: Ma et al. (2019); Rakhsha et al. (2020, 2021a); Rangi et al. (2022b); Zhang and Parkes (2008b); Zhang et al. (2009) focus on adversarial attack on offline single-agent reinforcement learners. Gleave et al. (2019); Guo et al. (2021) study the poisoning attack on multi-agent reinforcement learners, assuming that the attacker controls one of the learners. Our model instead assumes that the attacker is not one of the learners, and the attacker wants to and is able to poison the rewards of all learners at the same time. Our model pertains to many applications such as autonomous driving, robotics, traffic control, and economic analysis, in which there is a central controller whose interests are not aligned with any of the agents and can modify the rewards and therefore manipulate all agents at the same time.

Constrained Mechanism Design: Our paper is also related to the mechanism design literature, in particular, the K-implementation problem in Monderer and Tennenholtz (2004); Anderson et al. (2010). Our model differs mainly in that the attacker, unlike a mechanism designer, does not alter the game/environment directly, but instead modifies the training data, from which the learners infer the underlying game and compute their policy accordingly. In practical applications, rewards are often stochastic due to imprecise measurement and state observation, hence the mechanism design approach is not directly applicable to MARL reward poisoning. Conversely, constrained mechanism design can be viewed as a special case when the rewards are deterministic and the training data has uniform coverage of all period-state-action tuples.

Defense against Attacks on Reinforcement Learning: There is also recent work on defending against reward poisoning or adversarial attacks on reinforcement learning; examples include Banihashem et al. (2021); Lykouris et al. (2021); Rangi et al. (2022a); Wei et al. (2022); Wu et al. (2022); Zhang et al. (2021b,c). These work focus on the single-agent setting

where attackers have limited ability to modify the training data. We are not aware of defenses against reward poisoning in our offline multi-agent setting. Given the numerous real-world applications of offline MARL, we believe it is important to study the multi-agent version of the problem.

4.2 Preliminaries

Markov Games. A finite-horizon general-sum n-player Markov Game is given by a tuple $G = (\mathcal{S}, \mathcal{A}, P, R, H, \mu)$ Littman (1994). 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 that learner i takes action a_i and the other $n-1$ learners take joint action a_{-i}. For each period $h \in [H]$, $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition function, 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 and period h when the joint action $\boldsymbol{a}$ is taken. The initial state distribution is μ.

Policies and value functions. We use π to denote a deterministic Markovian *policy* for the n players, 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 and 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 represents the expected cumulative rewards of the game assuming learners take actions according to π. Formally, the Q value of learner i in state s in period h under a joint action $\boldsymbol{a}$ is given

recursively by

$$Q^{\pi}_{i,H}(s,\boldsymbol{a}) = R_{i,H}(s,\boldsymbol{a}),$$
$$Q^{\pi}_{i,h}(s,\boldsymbol{a}) = R_{i,h}(s,\boldsymbol{a}) + \sum_{s'\in\mathcal{S}} P_h(s'|s,\boldsymbol{a})\, V^{\pi}_{i,h+1}(s').$$

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

Offline MARL. In offline MARL, the learners are given a *fixed* batch dataset $\mathcal{D}$ that records historical plays of n agents under some behavior policies, and no further sampling is allowed. We assume that $\mathcal{D} = \left\{\left(s^{(k)}_h, \boldsymbol{a}^{(k)}_h, \boldsymbol{r}^{0,(k)}_h\right)_{h=1}^{H}\right\}_{k=1}^{K}$ contains K episodes of length H. The data tuple in period h of episode k consists of the state $s^{(k)}_h \in \mathcal{S}$, the joint action profile $\boldsymbol{a}^{(k)}_h \in \mathcal{A}$, and reward vector $\boldsymbol{r}^{0,(k)}_h \in \mathbb{R}^n$, where the superscript 0 denotes the original rewards before any attack. The next state $s^{(k)}_{h+1}$ can be found in the next tuple. Given the shared data $\mathcal{D}$, each learner independently constructs a policy π_i to maximize their own cumulative reward. They then behave according to the resulting joint policy $\pi = (\pi_1, \ldots, \pi_n)$ in future deployment. Note that in a multi-agent setting, the learners' optimal solution concept is typically an approximate Nash equilibrium or Dominant Strategy Equilibrium Cui and Du (2022b); Zhong et al. (2022).

An agent's access to $\mathcal{D}$ may be limited, for example, due to privacy reasons. There are multiple levels of accessibility. In the first level, the agents can only access data that directly involves itself: instead of the tuple $(s_h, \boldsymbol{a}_h, \boldsymbol{r}_h)$, agent i would only be able to see $(s_h, a_{i,h}, r_{i,h})$. In the second level, agent i can see the joint action but only its own reward: $(s_h, \boldsymbol{a}_h, r_{i,h})$. In the third level, agent i can see the whole $(s_h, \boldsymbol{a}_h, \boldsymbol{r}_h)$. We focus on the second level in this paper.

Let $N_h(s,\boldsymbol{a}) = \sum_{k=1}^{K} \mathbf{1}_{\{s^{(k)}_h = s, \boldsymbol{a}^{(k)}_h = \boldsymbol{a}\}}$ be the total number of episodes

containing $(s, \mathbf{a}, \cdot)$ in period h. We consider a dataset $\mathcal{D}$ that satisfies the following coverage assumption.

Assumption 4.1. (*Full Coverage*) *For each* $(s, \mathbf{a})$ *and* h, $N_h(s, \mathbf{a}) > 0$.

While this assumption might appear strong, we later show that it is necessary to effectively poison the dataset.

Attack Model

We assume that the attacker has access to the original dataset $\mathcal{D}$. The attacker has a pre-specified target policy $\boldsymbol{\pi}^\dagger$ and attempts to poison the rewards in $\mathcal{D}$ with the goal of forcing the learners to learn $\boldsymbol{\pi}^\dagger$ from the poisoned dataset. The attacker also desires that the attack has a minimal cost. We let $C(r^0, r^\dagger)$ denote the cost of a specific poisoning, where $r^0 = \left\{\left(\mathbf{r}_h^{0,(k)}\right)_{h=1}^H\right\}_{k=1}^K$ are the original rewards and $r^\dagger = \left\{\left(\mathbf{r}_h^{\dagger,(k)}\right)_{h=1}^H\right\}_{k=1}^K$ are the poisoned rewards. We focus on the L^1-norm cost $C(r^0, r^\dagger) = \|r^0 - r^\dagger\|_1$.

Rationality. For generality, the attacker makes minimal assumptions about the learners' rationality. Namely, the attacker only assumes that the learners never take dominated actions Monderer and Tennenholtz (2004). For technical reasons, we strengthen this assumption slightly by introducing an arbitrarily small margin $\iota > 0$ (e.g. representing the learners' numerical resolution).

Definition 4.1. *A* ι*-strict Markov perfect dominant strategy equilibrium* (ι*-MPDSE*) *of a Markov Game* G *is a policy* π *satisfying that for all learners* $i \in [n]$*, periods* $h \in [H]$*, and states* $s \in \mathcal{S}$*,*

$$\begin{aligned}&\forall\, a_i \in \mathcal{A}_i, a_i \neq \pi_{i,h}(s), \mathbf{a}_{-i} \in \mathcal{A}_{-i}:\\ &\qquad Q_{i,h}^\pi(s, (\pi_{i,h}(s), \mathbf{a}_{-i})) \geqslant Q_{i,h}^\pi(s, (a_i, \mathbf{a}_{-i})) + \iota.\end{aligned}$$

Note that a strict MPDSE, if exists, must be unique.

Assumption 4.2. (*Rationality*) *The learners will play an* ι*-MPDSE should one exist.*

Uncertainty-aware attack. State-of-the-art MARL algorithms are typically uncertainty-aware Cui and Du (2022b); Zhong et al. (2022), meaning that learners are cognizant of the model uncertainty due to finite, random data and will calibrate their learning procedure accordingly. The attacker accounts for such uncertainty-aware learners but does not know the learners' specific algorithm or internal parameters. It only assumes that the policies computed by the learners are solutions to some game that is plausible given the dataset. Accordingly, the attacker aims to poison the dataset in such a way that the target policy is an ι-MPDSE for every game that is plausible for the poisoned dataset.

To formally define the set of plausible Markov Games for a given dataset $\mathcal{D}$, we first need a few definitions.

Definition 4.2. (*Confidence Game Set*) *The confidence set on the transition function* $P_h(s, \mathbf{a})$ *has the form:*

$$\begin{aligned} \mathrm{CI}^{P}_{h}(s, \mathbf{a}) := \big\{ & P_h(s, \mathbf{a}) \in \Delta(\mathcal{A}) : \\ & \|P_h(s, \mathbf{a}) - \hat{P}_h(s, \mathbf{a})\|_1 \leqslant \rho^{P}_{h}(s, \mathbf{a}) \big\} \end{aligned}$$

where

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

$$\begin{aligned} \mathrm{CI}^{R}_{i,h}(s, \mathbf{a}) := \big\{ & R_{i,h}(s, \mathbf{a}) \in [-b, b] : \\ & |R_{i,h}(s, \mathbf{a}) - \hat{R}_{i,h}(s, \mathbf{a})| \leqslant \rho^{R}_{h}(s, \mathbf{a}) \big\}, \end{aligned}$$

where

$\hat{R}_{i,h}(s,\mathbf{a}) := \frac{1}{N_h(s,\mathbf{a})}\sum_{k=1}^{K} r_{i,h}^{0,(k)}\mathbf{1}_{\{s_h^{(k)}=s,\mathbf{a}_h^{(k)}=\mathbf{a}\}}$ *is the MLE of the reward. Then, the set of all plausible Markov Games consistent with* $\mathcal{D}$*, denoted by* CI^G*, is defined to be:*

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

Note that both the attacker and the learners know that all of the rewards are bounded within $[-b,b]$ (we allow $b=\infty$). The values of $\rho_h^P(s,\mathbf{a})$ and $\rho_h^R(s,\mathbf{a})$ are typically given by concentration inequalities. One standard choice takes the Hoeffding-type form $\rho_h^P(s,\mathbf{a}) \propto 1/\sqrt{\max\{N_h(s,a),1\}}$, and $\rho_h^R(s,\mathbf{a}) \propto 1/\sqrt{\max\{N_h(s,a),1\}}$, where we recall that $N_h(s,a)$ is the visitation count of the state-action pair (s,a) Xie et al. (2020); Cui and Du (2022b); Zhong et al. (2022). We remark that with proper choice of ρ_h^P and ρ_h^R, CI^G contains the game constructed by optimistic MARL algorithms with upper confidence bounds Xie et al. (2020), as well as that by pessimistic algorithms with lower confidence bounds Cui and Du (2022b); Zhong et al. (2022). See the appendix for details.

With the above definition, we consider an attacker that attempts to modify the original dataset $\mathcal{D}$ into $\mathcal{D}^\dagger$ so that $\boldsymbol{\pi}^\dagger$ is an ι-MPDSE for every plausible game in CI^G induced by the poisoned $\mathcal{D}^\dagger$. This would guarantee the learners adopt $\boldsymbol{\pi}^\dagger$.

The full coverage Assumption 4.1 is necessary for the above attack goal, as shown in the following proposition. We defer the proof to the appendix.

Proposition 4.1. *If* $N_h(s,\mathbf{a}) = 0$ *for some* $(h,s,\mathbf{a})$*, then there exist MARL learners for which the attacker's problem is infeasible.*

$\mathcal{A}_1 \setminus \mathcal{A}_2$	1	2
1	$(3,3)$	$(1,2)$
2	$(2,1)$	$(0,0)$

Table 4.1: Single-agent attack reduction example

$\mathcal{A}_i$	r
1	$\{3,1\}$
2	$\{2,0\}$

Table 4.2: Single-agent attack reduction

4.3 Poisoning Framework

In this section, we first argue that naively applying single-agent poisoning attacks separately to each agent results in suboptimal attack cost. We then present a new optimal poisoning framework that accounts for multiple agents and thereby allows for efficiently solving the attack problem.

Suboptimality of single-agent attack reduction. As a first attempt, the attacker could try to use existing single-agent RL reward poisoning methods. However, this approach is doomed to be suboptimal. Consider the game in Table 4.1 with $n = 2$ learners, one period, and one state.

Suppose that the original dataset $\mathcal{D}$ has full coverage. For simplicity, we assume that each $(s, \mathbf{a})$ pair appears sufficiently many times so that ρ^R is small. In this case, the target policy $\boldsymbol{\pi}^\dagger = (1,1)$ is already an MPDSE, so no reward modification is needed. However, if we use a single-agent approach, each learner i will observe the dataset in Table 4.2. In this case, to learner i it is not immediately clear which of the two actions is strictly better, for example, when $1,2$ appears relatively more often than $3,0$. To ensure that both players take action 1, the attacker needs to modify at least one of the rewards for each player, thus incurring a nonzero (and thus suboptimal) attack cost.

The example above shows that a new approach is needed to construct

an optimal poisoning framework tailored to the multi-agent setting. Below we develop such a framework, first for the simple Bandit Game setting, which is then generalized to Markov Games.

Bandit Game Setting

As a stepping stone, we start with a subclass of Markov Games with $|\mathcal{S}| = 1$ and $H = 1$, which are sometimes called bandit games. A bandit game consists of a single-stage normal-form game. For now, we also pretend that the learners simply use the data to compute an MLE point estimate $\hat{G}$ of the game and then solve the estimated game $\hat{G}$. This is unrealistic, but it highlights the attacker's strategy to enforce that $\boldsymbol{\pi}^\dagger$ is an ι-strict DSE in $\hat{G}$.

Suppose the original dataset is $\mathcal{D} = \left\{(\mathbf{a}^{(k)}, \mathbf{r}^{0,(k)})\right\}_{k=1}^{K}$ (recall we no longer have state or period). Also, let $N(\mathbf{a}) := \sum_{k=1}^{K} \mathbf{1}_{\{\mathbf{a}^{(k)}=\mathbf{a}\}}$ be the action counts. The attacker's problem can be formulated as a convex optimization problem given in (4.1).

$$
\begin{aligned}
&\min_{r^\dagger} \; C\left(r^0, r^\dagger\right) \\
&\text{s.t. } R^\dagger(\mathbf{a}) := \frac{1}{N(\mathbf{a})} \sum_{k=1}^{K} \mathbf{r}^{\dagger,(k)} \mathbf{1}_{\{\mathbf{a}^{(k)}=\mathbf{a}\}}, \forall\, \mathbf{a}; \\
&\qquad R_i^\dagger\left(\pi_i^\dagger, a_{-i}\right) \geqslant R_i^\dagger\left(a_i, a_{-i}\right) + \iota, \forall\, i, a_{-i}, a_i \neq \pi_i^\dagger; \\
&\qquad \mathbf{r}^{\dagger,(k)} \in [-b, b]^n, \forall\, k.
\end{aligned}
\tag{4.1}
$$

The first constraint in (4.1) models the learners' MLE $\hat{G}$ after poisoning. The second constraint enforces that $\boldsymbol{\pi}^\dagger$ is an ι-strict DSE of $\hat{G}$ by definition. We observe that:

1. The problem is feasible if $\iota \leqslant 2b$, since the attacker can always set, for each agent, the reward to be b for the target action and $-b$ for all other actions;

2. If the cost function $C(\cdot,\cdot)$ is the L^1-norm, the problem is a linear program (LP) with nK variables and $(A-1)A^{n-1}+2nK$ inequality constraints (assuming each learner has $|\mathcal{A}_i| = A$ actions);

3. After the attack, learner i only needs to see its own rewards to be convinced that $\pi_i^\dagger$ is a dominant strategy; learner i does not need to observe other learners' rewards.

This simple formulation serves as an asymptotic approximation to the attack problem for confidence-bound-based learners. In particular, when $N(\mathbf{a})$ is large for all $\mathbf{a}$, the confidence intervals on P and R are usually small.

With the above idea in place, we can consider more realistic learners that are uncertainty-aware. For these learners, the attacker attempts to enforce an ι separation between the lower bound of the target action's reward and the upper bounds of all other actions' rewards (similar to arm elimination in bandits). With such separation, all plausible games in CI^G would have the target action profile as the dominant strategy equilibrium. This approach can be formulated as a slightly more complex optimization problem (4.2), where the second and third constraints enforce the desired ι separation. The formulation (4.2) can be solved using standard optimization solvers, hence the optimal attack can be computed efficiently.

$$
\begin{aligned}
&\min_{r^\dagger} \; C(r^0, r^\dagger) \\
&\text{s.t. } R^\dagger(a) := \frac{1}{N(a)} \sum_{k=1}^{K} \mathbf{r}^{\dagger,(k)} \mathbf{1}_{\{\mathbf{a}^{(k)}=a\}}, \forall\, \mathbf{a}; \\
&\qquad CI_i^{R^\dagger}(\mathbf{a}) := \left\{ R_i(\mathbf{a}) \in [-b, b] : \left| R_i(\mathbf{a}) - R_i^\dagger(\mathbf{a}) \right| \right. \\
&\qquad\qquad \left. \leqslant \rho^R(\mathbf{a}) \right\}, \quad \forall\, i, \mathbf{a}; \\
&\qquad \min_{R_i \in CI_i^{R^\dagger}(\pi_i^\dagger, a_{-i})} R_i \geqslant \max_{R_i \in CI_i^{R^\dagger}(a_i, a_{-i})} R_i + \iota, \\
&\qquad\qquad \forall\, i, a_{-i}, a_i \neq \pi_i^\dagger; \\
&\qquad \mathbf{r}^{\dagger,(k)} \in [-b, b]^n, \forall\, k.
\end{aligned}
\tag{4.2}
$$

We next consider whether this formulation has a feasible solution. Below we characterize the feasibility of the attack in terms of the margin parameter ι and the confidence bounds.

Proposition 4.2. *The attacker's problem* (4.2) *is feasible if* $\iota \leqslant 2b - 2\rho^R(\mathbf{a}), \forall\, \mathbf{a} \in \mathcal{A}$.

Proposition 4.2 is a special case of the general Theorem 4.1 with $H = |\mathcal{S}| = 1$. We note that the condition in Proposition 4.2 has an equivalent form that relates to the structure of the dataset. We later present this form for a more general case.

When an L^1-norm cost function is used, we show in the appendix that the formulation (4.2) can also be efficiently solved.

Proposition 4.3. *With* L^1*-norm cost function* $C(\cdot, \cdot)$*, the problem* (4.2) *can be formulated as a linear program.*

Markov Game Setting

We now generalize the ideas from the bandit setting to derive a poisoning framework for arbitrary Markov Games. With multiple states and periods, there are two main complications:

1. In each period h, the learners' decision depends on Q_h, which involves both the immediate reward R_h and the future return Q_{h+1};

2. The uncertainty in Q_h amplifies as it propagates backward in h.

Accordingly, the attacker needs to design the poisoning attack recursively.

Our main technical innovation is an attack formulation based on Q *confidence-bound backward induction*. The attacker maintains confidence upper and lower bounds on the learners' Q function, $\overline{Q}$, and $\underline{Q}$, with backward induction. To ensure $\boldsymbol{\pi}^\dagger$ becomes an ι-MPDSE, the attacker again attempts to ι-separate the lower bound of the target action and the upper bound of all other actions, at all states and periods.

Recall Definition 4.2: given the training dataset $\mathcal{D}$, one can compute the MLEs $\mathbf{R}_h$ and corresponding confidence sets $\mathrm{CI}^R_{i,h}$ for the reward. The attacker aims to poison $\mathcal{D}$ into $\mathcal{D}^\dagger$ so that the MLEs and confidence sets become $\mathbf{R}^\dagger_h$ and $\mathrm{CI}^{R^\dagger}_{i,h}$, under which $\boldsymbol{\pi}^\dagger$ is the unique ι-MPDSE for all plausible games in the corresponding confidence game set. The attacker finds the minimum cost way of doing so by solving a Q confidence-bound backward induction optimization problem, given in (4.3)–(4.7).

$$\min_{r^\dagger} \; C\left(r^0, r^\dagger\right) \tag{4.3}$$

$$\text{s.t. } R^\dagger_{i,h}(s,\mathbf{a}) := \frac{1}{N_h(s,\mathbf{a})} \sum_{k=1}^{K} r^{\dagger,(k)}_{i,h} \mathbf{1}_{\left\{s_h^{(k)}=s, \mathbf{a}_h^{(k)}=\mathbf{a}\right\}},$$

$$\forall\, h, s, i, \mathbf{a}$$

$$\mathrm{CI}^{R^\dagger}_{i,h}(s,\mathbf{a}) := \Big\{ R_{i,h}(s,\mathbf{a}) \in [-b, b]$$

$$: \left| R_{i,h}(s,\mathbf{a}) - R^\dagger_{i,h}(s,\mathbf{a}) \right| \leqslant \rho^R_h(s,\mathbf{a}) \Big\},$$

$$\forall\, h, s, i, \mathbf{a}$$

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

$$\underline{Q}_{i,h}(s,\boldsymbol{a}) := \min_{R_{i,h}\in CI^{R^\dagger}_{i,h}(s,\boldsymbol{a})} R_{i,h} + \min_{P_h\in CI^{P}_{h}(s,\boldsymbol{a})} \sum_{s'\in\mathcal{S}} P_h(s')\,\underline{Q}_{i,h+1}\left(s',\boldsymbol{\pi}^\dagger_{h+1}(s')\right),$$
$$\forall\, h<H,s,i,\boldsymbol{a} \tag{4.4}$$

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

$$\overline{Q}_{i,h}(s,\boldsymbol{a}) := \max_{R_{i,h}\in CI^{R^\dagger}_{i,h}(s,\boldsymbol{a})} R_{i,h} + \max_{P_h\in CI^{P}_{h}(s,\boldsymbol{a})} \sum_{s'\in\mathcal{S}} P_h(s')\,\overline{Q}_{i,h+1}\left(s',\boldsymbol{\pi}^\dagger_{h+1}(s')\right),$$
$$\forall\, h<H,s,i,\boldsymbol{a} \tag{4.5}$$

$$\underline{Q}_{i,h}\left(s,\left(\pi^\dagger_{i,h}(s),a_{-i}\right)\right) \geqslant \overline{Q}_{i,h}(s,(a_i,a_{-i}))+\iota,$$
$$\forall\, h,s,i,a_{-i},a_i\neq\pi^\dagger_{i,h}(s) \tag{4.6}$$

$$\mathbf{r}^{\dagger,(k)}_h\in[-b,b]^n,\forall\, h,k. \tag{4.7}$$

The backward induction steps (4.4) and (4.5) 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 (4.6) enforce an ι-separation between the target action and other actions at all states and periods. We emphasize that the agents need not consider Q at all in their learning algorithm; Q only appears in the optimization due to its presence in the definition of MPDSE.

Again, pairing an efficient optimization solver with the above formulation gives an efficient algorithm for constructing the poisoning. We now answer the important questions of whether this formulation admits a feasible solution and whether these solutions yield successful attacks. The lemma below provides a positive answer to the second question.

Lemma 4.1. *If the attack formulation* (4.3)–(4.7) *is feasible,* $\boldsymbol{\pi}^{\dagger}$ *is the unique* ι*-MPDSE of every Markov Game* $G \in CI^{G}$.

Moreover, the attack formulation admits feasible solutions under mild conditions on the dataset.

Theorem 4.1. *The attacker formulation* (4.3)–(4.7) *is feasible if the following condition holds:*

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

We remark that the learners know the upper bound b and may use it to exclude implausible games. The accumulation of confidence intervals over the H periods results in the extra factor $(H+1)$ on ρ_h^R. Theorem 4.1 implies that the problem is feasible so long as the dataset is sufficiently populated; that is, each (s,a) pair should appear frequently enough to have a small confidence interval half-width ρ_h^R. The following corollary provides a precise condition on the visit accounts that guarantees feasibility.

Corollary 4.1. *Given a confidence probability* δ *and the confidence interval half-width* $\rho_h^R(s,\boldsymbol{a}) = f(\frac{1}{N_h(s,a)})$ *for some strictly increasing function* f*, the condition in Theorem 4.1 holds if*

$$N_h(s,\boldsymbol{a}) \geqslant \left(f^{-1}\left(\frac{2b-\iota}{H+1}\right)\right)^{-1}.$$

In particular, for the natural choice of Hoeffding-type,

$\rho_h^R(s,\boldsymbol{a}) = 2b\sqrt{\dfrac{\log\left((H|\mathcal{S}||\mathcal{A}|)/\delta\right)}{\max\{N_h(s,\boldsymbol{a}),1\}}}$*, it suffices that,*

$$N_h(s,\boldsymbol{a}) \geqslant \frac{4b^2(H+1)^2\log\left((H|\mathcal{S}||\mathcal{A}|)/\delta\right)}{(2b-\iota)^2}.$$

Despite the inner min and max in the problem (4.3)–(4.7), the problem can be formulated as an LP, thanks to LP duality.

Theorem 4.2. *With* L^1*-norm cost function* $C(\cdot,\cdot)$*, problem* (4.3)–(4.7) *can be formulated as an LP.*

The proofs of the above results can be found in the appendix.

4.4 Cost Analysis

Now that we know how the attacker can poison the dataset in the multi-agent setting, we can study the structure of attacks. The structure is most easily seen by analyzing the minimal attack cost. To this end, we give general bounds that relate the minimal attack cost to the structure of the underlying Markov Game. The attack cost upper bounds show which games are particularly susceptible to poison, and the attack cost lower bounds demonstrate that some games are expensive to poison.

Overview of results: Specifically, we shall present two types of upper/lower bounds on the attack cost: (i) *universal bounds* that hold for all attack problem instances simultaneously; (ii) *instance-dependent bounds* that are stated in terms of certain properties of the instance. We also discuss problem instances under which these two types of bounds are tight and coincide with each other.

We note that all bounds presented here are with respect to the L^1-cost, but many of them generalize to other cost functions, especially the L^∞-cost. The proofs of the results presented in this section are provided in the appendix.

Setup: Let $I = (\mathcal{D}, \boldsymbol{\pi}^\dagger, \rho^R, \rho^P, \iota)$ denote an instance of the attack problem, and $\hat{G}$ denote the corresponding MLE of the Markov Game derived from $\mathcal{D}$. We denote by $I_h = (\mathcal{D}_h, \boldsymbol{\pi}^\dagger_h, \rho^R_h, \rho^P_h, \iota)$ the restriction of the instance to period h. In particular, $\hat{R}_h(s)$ derived from $\mathcal{D}_h$ is exactly the

normal-form game at state s and period h of $\hat{G}$. We define $C^*(I)$ to be the optimal L^1-poisoning cost for the instance I; that is, $C^*(I)$ is the optimal value of the optimization problem (4.3)–(4.7) evaluated on I. We say the attack instance I is *feasible* if this optimization problem is feasible. If I is infeasible, we define $C^*(I) = \infty$. WLOG, we assume that $|\mathcal{A}_1| = \cdots = |\mathcal{A}_n| = A$. In addition, we define the minimum visit count for each 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 minimum over all periods as $\underline{N} := \min_{h\in H} \underline{N}_h$. We similarly define the maximum visit counts as $\overline{N}_h = \max_{s\in\mathcal{S}} \max_{\boldsymbol{a}\in\mathcal{A}} N_h(s, \boldsymbol{a})$ and $\overline{N} = \max_h \overline{N}_h$. Lastly, we define $\underline{\rho} = \min_{h,s,\boldsymbol{a}} \rho_h^R(s, \boldsymbol{a})$ and $\overline{\rho} = \max_{h,s,\boldsymbol{a}} \rho_h^R(s, \boldsymbol{a})$, the minimum and maximum confidence half-width.

Universal Cost Bounds

With the above definitions, we present universal attack cost bounds that hold simultaneously for all attack instances.

Theorem 4.3. *For any feasible attack instance* I*, we have that,*

$$0 \leqslant C^*(I) \leqslant \overline{N} H |\mathcal{S}| n A^n 2b.$$

As these upper and lower bounds hold for all instances, they are typically loose. However, they are nearly tight. If $\boldsymbol{\pi}^\dagger$ is already an ι-MPDSE for all plausible games, then no change to the rewards is needed and the attack cost is 0, hence the lower bound is tight for such instances. We can also construct a high-cost instance to show the near-tightness of the upper bound.

Specifically, consider the dataset for a bandit game, $\mathcal{D} = \left\{(\boldsymbol{a}^{(k)}, \boldsymbol{r}^{0,(k)})\right\}_{k=1}^{K}$, where $\mathcal{A} = A^n$ and each action appears exactly N times, i.e., $\overline{N} = \underline{N} = N$ and $K = NA^n$. The target policy is $\boldsymbol{\pi}^\dagger = (1, \ldots, 1)$. The dataset is constructed so that $\boldsymbol{r}_i^{0,(k)} = -b$ if $\boldsymbol{a}_i^{(k)} = \pi_{i,h}^\dagger(s)$ and $\boldsymbol{r}_i^{0,(k)} = b$ otherwise. These rewards are essentially the extreme opposite of what the attacker

$\mathcal{A}_1/\mathcal{A}_2$	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 4.3: MLE $\hat{\mathbf{R}}_h(s, \cdot)$ before attack

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

Table 4.4: MLE $\hat{\mathbf{R}}_h(s, \cdot)$ after attack

needs to ensure $\boldsymbol{\pi}^\dagger$ is an ι-DSE. Note, the dataset induces the MLE of the game shown in Table 4.3 for the special case with $n = 2$ players.

For simplicity, suppose that the same confidence half-width $\rho^R(\mathbf{a}) = \rho < b$ is used for all $\mathbf{a}$. Let $\iota \in (0, b)$ be arbitrary. For this instance, to install $\boldsymbol{\pi}^\dagger$ as the ι-DSE, the attacker can flip all rewards in a way that is illustrated in Table 4.4, inducing a cost as the upper bound in Theorem 4.3. The situation is the same for $n \geqslant 2$ learners. Our instance-dependent lower bound, presented later in Theorem 4.5, implies that any attack on this instance must have cost at least $NnA^{n-1}(2b + 2\rho + \iota)$. This lower bound matches the refined upper bound in the proof of Theorem 4.4, implying the refined bounds are tight for this instance. Noticing that the universal bound in Theorem 4.3 only differs by an $O(A)$-factor implies it is nearly tight.

Instance-Dependent Cost Bounds

Next, we derive general bounds on the attack cost that depends on the structure of the underlying instance. Our strategy is to reduce the problem of bounding Markov Game costs to the easier problem of bounding Bandit

Game costs. We begin by showing that the cost of poisoning a Markov Game dataset can be bounded in terms of the cost of poisoning the datasets corresponding to its individual period games.

Theorem 4.4. *For any feasible attack instance* I*, we have that* $C^*(I_H) \leqslant C^*(I)$ *and,*

$$C^*(I) \leqslant \sum_{h=1}^{H} C^*(I_h) + 2bnH|\mathcal{S}|\overline{N} + H^2\overline{\rho}|\mathcal{S}|nA^n\overline{N}$$

Here we see the effect of the learner's uncertainty. If ρ^R is small, then poisoning costs slightly more than poisoning each bandit instance independently. This is desirable since it allows the attacker to solve the much easier bandit instances instead of the full problem.

The lower bound is valid for all Markov Games, but it is weak in that it only uses the last period cost. However, this is the most general lower bound one can obtain without additional assumptions on the structure of the game. If we assume additional structure on the dataset, then the above lower bound can be extended beyond the last period, forcing a higher attack cost.

Lemma 4.2. *Let* I *be any feasible attack instance containing at least one uniform transition in* CI_h^P *for each period* h*, i.e., there is some* $\hat{P}_h(s' \mid s, \mathbf{a}) \in CI_h^P$ *with* $\hat{P}_h(s' \mid s, \mathbf{a}) = 1/|\mathcal{S}|, \forall h, s', s, \mathbf{a}$*. Then, we have that*

$$C^*(I) \geqslant \sum_{h=1}^{H} C^*(I_h).$$

In words, for these instances the optimal cost for poisoning is not too far off from the optimal cost of poisoning each period game independently. We note this is where the effects of ρ^P show themselves. If the dataset is highly uncertain on the transitions, it becomes likely that a uniform transition exists in CI^P. Thus, a higher ρ^P leads to a higher cost and effectively devolves the set of plausible games into a series of independent games.

Now that we have the above relationships, we can focus on bounding the attack cost for bandit games. To be precise, we bound the cost of poisoning a period game instance I_h. To this end, we define ι-dominance gaps.

Definition 4.3. (*Dominance Gaps*) *For every* $h \in [H], s \in \mathcal{S}, i \in [n]$ *and* $a_{-i} \in \mathcal{A}_{-i}$, *the* ι*-dominance gap,* $d^{\iota}_{i,h}(s, a_{-i})$, *is defined as*

$$
\begin{aligned}
&d^{\iota}_{i,h}(s, a_{-i}) := \\
&\Big[\max_{a_i \neq \pi^{\dagger}_{i,h}(s)} \Big[\hat{R}_{i,h}\big(s, (a_i, a_{-i})\big) + \rho^{R}_{h}\big(s, (a_i, a_{-i})\big) \Big] \\
&- \hat{R}_{i,h}\Big(s, \big(\pi^{\dagger}_{i,h}(s), a_{-i}\big)\Big) + \rho^{R}_{h}\Big(s, \big(\pi^{\dagger}_{i,h}(s), a_{-i}\big)\Big) + \iota \Big]_{+}
\end{aligned}
$$

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

The dominance gaps measure the minimum amount by which the attacker would have to increase the reward for learner i while others are playing a_{-i}, so that the action $\pi^{\dagger}_{i,h}(s)$ becomes ι-dominant for learner i. We then consolidate all the dominance gaps for period h into the variable $\Delta_h(\iota)$,

$$
\Delta_h(\iota) := \sum_{s \in S} \sum_{i=1}^{n} \sum_{a_{-i}} \Big(d^{\iota}_{i,h}(s, a_{-i}) + \delta^{\iota}_{i,h}(s, a_{-i}) \Big)
$$

Where $\delta^{\iota}_{i,h}(s, a_{-i})$ is a minor overflow term defined in the appendix. With all this machinery set up, we can give precise bounds on the minimal cost needed to attack a single-period game.

Lemma 4.3. *The optimal attack cost for* I_h *satisfies*

$$
\underline{N}_h \Delta_h(\iota) \leqslant C^*(I_h) \leqslant \overline{N}_h \Delta_h(\iota).
$$

Combining these bounds with Theorem 4.4 gives complete attack cost bounds for general Markov game instances.

The lower bounds in both Lemma 4.2 and Lemma 4.3 expose an exponential dependency on n, the number of players, for some datasets $\mathcal{D}$. These instances essentially require the attacker to modify $\hat{R}_{i,h}(s, \mathbf{a})$ for every $\mathbf{a} \in \mathcal{A}$. A concrete instance can be constructed by taking the high-cost dataset derived as the tight example before and extending it into a general Markov Game. We simply do this by giving the game several identical states and uniform transitions. In terms of the dataset, each episode consists of independent plays of the same normal-form game, possibly with a different state observed. For this dataset the ι-dominance gap can be shown to be $d^{\iota}_{i,h}(s, a_{-i}) = 2b + 2\rho + \iota$. A direct application of Lemma 4.2 gives the following explicit lower bound.

Theorem 4.5. *There exists a feasible attack instance* I *for which it holds that*

$$C^*(I) \geqslant \underline{N} H |\mathcal{S}| n A^{n-1} (2b + 2\rho + \iota).$$

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

4.5 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 an analysis on the attack feasibility and cost. This paper thus helps to raise awareness of the trustworthiness of multi-agent learning. We encourage the commu-

nity to study defense against such attacks, e.g. via robust statistics and reinforcement learning.

In the next chapter, we investigate the problem of installing a Markov perfect Nash equilibrium, which would require fewer data points and incur a smaller data modification cost; however, characterization of the uniqueness of Nash equilibrium for general-sum games is difficult, so we focus on the problem for two-player zero-sum games. We provide a more general adversarial attack framework on Markov games in the next chapter and derive similar efficiency and feasibility results to the ones in this chapter.

5 OFFLINE REWARD POISONING FOR ZERO-SUM GAMES TO INSTALL A NASH EQUILIBRIUM

Contribution Statement. This chapter is a joint work with Jeremy McMahan, Jerry Zhu, and Qiaomin Xie. I am the main author. My contribution includes the statements and proofs of all the propositions and theorems and the writing of the paper.

5.1 Introduction

In this chapter, we study the offline data poisoning problem in a zero-sum Markov game environment, where a single attacker tries to minimally modify the offline rewards so that learners that compute the Markov perfect Nash equilibrium of a Markov game within some confidence region around the maximum likelihood estimate of the Markov game based on the offline rewards would find a deterministic target joint policy as the unique equilibrium. We formulate the attacker's reward poisoning problem as a linear program, which can be solved efficiently, and we provide sufficient conditions for the feasibility of such an attack.

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

can achieve this goal while minimizing its attack cost, which we define below. Clearly, such power poses a threat to the security of Multi-Agent Reinforcement Learning (MARL).

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

Problem Statement: Offline Data Poisoning.

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

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

Our framework can be summarized by the mnemonic "ToM moves to the UN". (i) UN stands for the Unique Nash set, which is the set of Q functions that make the target $\pi^\dagger$ the unique NE. Uniqueness is crucial for the attacker to ensure that MARL agents choose the target NE with certainty, without breaking ties arbitrarily among multiple NEs. (ii) ToM

stands for the attacker's Theory of Mind of the MARL agents, namely the plausible set of Q functions that the attacker believes the agents will entertain upon receiving the poisoned dataset $D^\dagger$. (iii) The attack is successful if, by controlling $D^\dagger$, the ToM set is moved inside the UN set. A successful attack with the smallest cost $C(D, D^\dagger)$ is optimal.

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

Understanding adversarial attacks in the multi-agent setting is critical since many real-life applications of MARL problems are susceptible to adversarial attacks. Examples of two-player zero-sum games include board games such as GO and Chess Silver et al. (2017, 2016), where the learners use historical game plays as training data and an attacker can potentially alter the data to change the behavior of the trained agents. In the case of competitive robotics, for example, robot soccer Gu et al. (2017); Riedmiller et al. (2009); Kober et al. (2013), they are trained on offline datasets and the attacker can mislead the trained policies by modifying the training sets. For finance applications, especially algorithmic or high-frequency stock or option trading Lee et al. (2007); Lee and O (2002) that are usually trained on historical prices, if the database is corrupted by an attacker, the learned trading strategies can be sub-optimal as well. There are also examples of multi-player games that have two-player games as special cases, for example, video games Vinyals et al. (2019); Jaderberg et al. (2019); Berner et al. (2019), card games Brown and Sandholm (2019);

Brown et al. (2017), autonomous driving Shalev-Shwartz et al. (2016), automated warehouses Yang et al. (2020), and economic policymaking, which can all be trained on offline datasets and become vulnerable to adversarial attacks. In all of the above MARL applications, the threat of adversarial attacks has not been investigated.

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

5.2 Offline Attack on a Normal-form Game

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

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

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

the expected reward $R(\pi) \coloneqq \sum_{a_1 \in \mathcal{A}_1, a_2 \in \mathcal{A}_2} \pi_1(a_1)\, \pi_2(a_2)\, R((a_1, a_2))$.

Definition 5.1 (Nash Equilibrium)**.** *A Nash equilibrium (NE) of a normal-form game $(\mathcal{A}, R)$ is a mixed strategy profile π that satisfies,*

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

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

$$\begin{aligned} R((\pi_1, a_2)) \leqslant R(\pi) \leqslant R((a_1, \pi_2)), \\ \forall\, a_1 \neq \pi_1, a_2 \neq \pi_2. \end{aligned} \tag{5.1}$$

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

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

Definition 5.2 (Unique Nash)**.** *The unique Nash set of a pure strategy profile π is the set of reward functions R such that $(\mathcal{A}, R)$ has a unique Nash equilibrium π,*

$$\mathcal{U}(\pi) \coloneqq \mathcal{N}^{-1}(\{\pi\}) = \{R \in \mathcal{R} : \mathcal{N}(R) = \{\pi\}\}. \tag{5.2}$$

To characterize $\mathcal{U}(\pi)$, we note that for normal-form games, a pure strategy profile π is the unique Nash equilibrium of a game if and only if it is a strict Nash equilibrium, which is defined as a policy π that satisfies (5.1) with strict inequalities.

Proposition 5.1 (Unique Nash Polytope)**.** *For any pure strategy profile* π*,*

$$\begin{aligned}\mathcal{U}(\pi) &= \{R \in \mathcal{R} : \pi \textit{ is a strict NE of } (\mathcal{A}, R)\} \\ &= \{R \in \mathcal{R} : R((\pi_1, a_2)) < R(\pi) < R((a_1, \pi_2)), \\ &\qquad \forall\, a_1 \neq \pi_1, a_2 \neq \pi_2\}. \end{aligned} \tag{5.3}$$

Here, the uniqueness is among all Nash equilibria including mixed-strategy Nash equilibria. The proof of the equivalence between (5.2) and (5.3) is in the appendix. We restrict our attention to pure-strategy equilibria and defer the discussion of mixed strategy profiles to the last section.

To avoid working with strict inequalities, we define a closed subset of $\mathcal{U}(\pi)$ of reward functions that lead to strict Nash equilibria with an ι reward gap, which means all strict inequalities in (5.3) are satisfied with a gap of at least ι, for some $\iota > 0$.

Definition 5.3 (Iota Strict Unique Nash)**.** *For* $\iota > 0$*, the* ι *strict unique Nash set of a pure strategy profile* π *is,* $\underline{\mathcal{U}}(\pi; \iota) \coloneqq$

$$\begin{aligned}&\{R \in \mathcal{R} : R((\pi_1, a_2)) + \iota \leqslant R(\pi) \leqslant R((a_1, \pi_2)) - \iota, \\ &\qquad \forall\, a_1 \neq \pi_1, a_2 \neq \pi_2\}. \end{aligned} \tag{5.4}$$

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

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

We provide a model of the attacker's theory of mind of the victim, which is the attacker's belief about the learning algorithm the victim uses. Formally, we define the theory-of-mind set as the set of plausible rewards that the

victim uses based on the given training dataset, and we assume that the victims compute the Nash equilibria based on the reward functions estimated from a dataset $D \in \mathcal{D}$, where $\mathcal{D}$ is the set of possible datasets with K episodes in the form $\left\{\left(\mathbf{a}^{(k)}, r^{(k)}\right)\right\}_{k=1}^{K}$, with $\mathbf{a}^{(k)} \in \mathcal{A}$ and $r^{(k)} \in [-b, b]$ for every $k \in [K]$.

Definition 5.4 (Theory of Mind)**.** *Given a dataset $D \in \mathcal{D}$, the theory-of-mind set $\mathcal{T}(D) \subseteq \mathcal{R}$ is the set of plausible reward functions that the victims estimate based on D to compute their equilibria. In particular, if the victims learn an action profile π, then $\pi \in \bigcup_{R \in \mathcal{T}(D)} \mathcal{N}(R)$.*

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

Definition 5.5 (Outer Approximation of Theory of Mind)**.** *An outer approximation of $\mathcal{T}(D)$ is a set denoted by $\overline{\mathcal{T}}(D)$ that satisfies $\mathcal{T}(D) \subseteq \overline{\mathcal{T}}(D)$ for every $D \in \mathcal{D}$, and can be written in the form, $\overline{\mathcal{T}}(D) :=$*

$$\left\{R \in \mathcal{R} : \left|R(\boldsymbol{a}) - \hat{R}(\boldsymbol{a})\right| \leqslant \rho^{(R)}(\boldsymbol{a}), \forall\, \boldsymbol{a} \in \mathcal{A}\right\}, \tag{5.5}$$

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

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

Example 5.1 (Theory of Mind for Maximum Likelihood Victims)**.** *Given a dataset $D \in \mathcal{D}$, if the attacker believes the victims are maximum likelihood*

learners, then $\mathcal{T}(\mathrm{D})$ *is a singleton* R^{MLE}*, where, for every* $\boldsymbol{a} \in \mathcal{A}$*,*

$$\mathrm{R}^{MLE}(\boldsymbol{a}|\mathrm{r}) \coloneqq \begin{cases} \frac{1}{\mathrm{N}(\boldsymbol{a})} \sum_{k=1}^{K} \mathrm{r}^{(k)} \mathbb{I}_{\left\{\boldsymbol{a}^{(k)}=\boldsymbol{a}\right\}} & \text{if } \mathrm{N}(\boldsymbol{a}) > 0 \\ 0 & \text{if } \mathrm{N}(\boldsymbol{a}) = 0 \end{cases}$$
$$\mathrm{N}(\boldsymbol{a}) \coloneqq \sum_{k=1}^{K} \mathbb{I}_{\left\{\boldsymbol{a}^{(k)}=\boldsymbol{a}\right\}}. \tag{5.6}$$

The smallest outer approximation $\overline{\mathcal{T}}(\mathrm{D})$ *can be specified using* $\hat{\mathrm{R}} = \mathrm{R}^{MLE}$ *and* $\rho^{(\mathrm{R})} = 0$*, and* $\overline{\mathcal{T}}$ *is linear since* (5.6) *is linear in* $\left\{\mathrm{r}^{(k)}\right\}_{k=1}^{K}$*.*

Example 5.2 (Theory of Mind for Pessimistic Optimistic Victims)**.** *Given a dataset* $\mathrm{D} \in \mathcal{D}$*, if the attacker believes the victims are learners that use pessimism and optimism by adding and subtracting bonus terms and estimating one or two games, as in Cui and Du* (2022a)*, then* $\mathcal{T}(\mathrm{D})$ *may contain two reward functions* $\underline{\mathrm{R}}$ *and* $\overline{\mathrm{R}}$*, where for every* $\boldsymbol{a} \in \mathcal{A}$*,*

$$\begin{aligned} \underline{\mathrm{R}}(\boldsymbol{a}|\mathrm{r}) &\coloneqq \mathrm{R}^{MLE}(\boldsymbol{a}|\mathrm{r}) - \beta(\boldsymbol{a}) \\ \overline{\mathrm{R}}(\boldsymbol{a}|\mathrm{r}) &\coloneqq \mathrm{R}^{MLE}(\boldsymbol{a}|\mathrm{r}) + \beta(\boldsymbol{a}), \end{aligned} \tag{5.7}$$

with $\beta(\boldsymbol{a}) = \frac{c}{\sqrt{\mathrm{N}(\boldsymbol{a})}}$ *being the bonus term, for some constant* c*.*
The smallest outer approximation $\overline{\mathcal{T}}(\mathrm{D})$ *can be specified using* $\hat{\mathrm{R}} = \mathrm{R}^{MLE}$ *and* $\rho^{(\mathrm{R})}(\boldsymbol{a}) = \beta(\boldsymbol{a})$ *for every* $\boldsymbol{a} \in \mathcal{A}$*, and* $\overline{\mathcal{T}}$ *is linear since* (5.6) *and* (5.7) *are both linear in* $\left\{\mathrm{r}^{(k)}\right\}_{k=1}^{K}$*.*

Example 5.3 (Theory of Mind for Data Splitting Victims)**.** *Given a dataset* $\mathrm{D} \in \mathcal{D}$*, if the attacker believes the victims use maximum likelihood estimates on a subsample of the* D*, similar to the data-splitting procedure in Cui and Du* (2022a)*, then* $\overline{\mathcal{T}}(\mathrm{D})$ *could be viewed as a high-probability set of rewards that the victims are estimating and* $\rho^{(\mathrm{R})}$ *would be half of the confidence interval width for the mean of the subsample around the mean of the complete dataset* R^{MLE}*.*

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

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

Example 5.4 (L_1 Cost Function)**.** *For reward poisoning problems, we define the* L_1 *cost of modifying the dataset from* $D = \left\{\left(\boldsymbol{a}^{(k)}, r^{(k)}\right)\right\}_{k=1}^{K}$ *to* $D^\dagger = \left\{\left(\boldsymbol{a}^{(k)}, r^{\dagger,(k)}\right)\right\}_{k=1}^{K}$ *by* $C^{(1)}\left(D, D^\dagger\right) \coloneqq \sum_{k=1}^{K} \left|r^{(k)} - r^{\dagger,(k)}\right|$.

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

Definition 5.6 (Attacker's Problem)**.** *The attacker's problem with the target action profile* $\pi^\dagger$ *is,*

$$\begin{aligned} &\inf_{D^\dagger \in \mathcal{D}(D)} C\left(D, D^\dagger\right) \\ &\quad \text{s.t. } \mathcal{T}\left(D^\dagger\right) \subseteq \mathcal{U}\left(\pi^\dagger\right). \end{aligned} \tag{5.8}$$

In general, (5.8) cannot be solved efficiently, but for reward poisoning problems with L_1 cost objective, we can relax the attacker's problem using ι strict unique Nash sets, which is a polytope described by (5.4), and a linear outer approximation of the theory-of-mind set, a hypercube

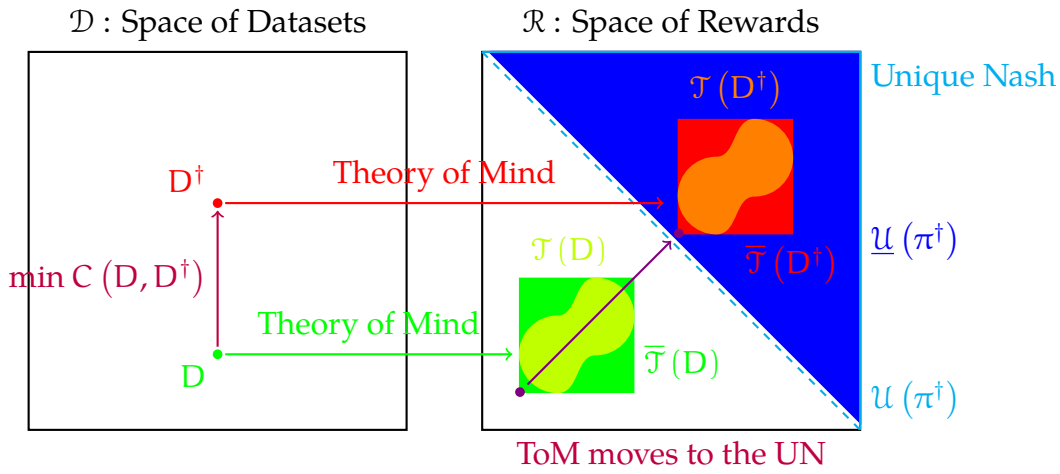


Figure 5.1: Attacker's Problem

described by (5.5), which can be converted into a linear program and solved efficiently. We state this observation as the following proposition and depict the relationship between the sets in Figure 5.1.

Proposition 5.2 (Reward Poisoning Linear Program)**.** *Given* $\iota > 0$ *and a linear* $\overline{\mathcal{T}}$*, the following problem is a relaxation of the attacker's reward poisoning problem and can be converted into a linear program,*

$$\min_{D^\dagger \in \mathcal{D}^{(R)}(D)} C^{(1)}(D, D^\dagger) \tag{5.9}$$
$$\text{s.t. } \overline{\mathcal{T}}(D^\dagger) \subseteq \underline{\mathcal{U}}(\pi^\dagger; \iota).$$

In Figure 5.1, given a dataset D, the general attacker's problem (5.8) of moving $\mathcal{T}(D)$ (light green) to $\mathcal{T}(D^\dagger)$ (light red) such that it is inside $\mathcal{U}(\pi^\dagger)$ (light blue) while minimizing the distance from D to $D^\dagger$ is often intractable. We construct a relaxed problem (5.9) of moving $\overline{\mathcal{T}}(D)$ (green) to $\overline{\mathcal{T}}(D^\dagger)$ (red) such that it is inside $\underline{\mathcal{U}}(\pi^\dagger)$ (blue), in which all sets are polytopes and thus can be converted to a linear program for linear costs and linear theory-of-mind mappings.

In the appendix, we provide the complete linear program and show that the solution of (5.9) is feasible for (5.8). The optimality of the linear program solution depends on how close the outer approximation of the theory-of-mind set is, and in the case when the theory-of-mind set is already a hypercube, the infimum in (5.8) can be achieved by taking the limit as $\iota \to 0$.

Example 5.5 (Maximum Likelihood Centered Linear Program)**.** *In the case* $\hat{R} = R^{MLE}$ *in the theory-of-mind set,* (5.9) *is given by,*

$$\begin{aligned} \min_{r^\dagger \in [-b,b]^K} & \sum_{k=1}^{K} \left| r^{(k)} - r^{\dagger,(k)} \right| \\ \text{s.t. } & R^{MLE}\left(r^\dagger\right) \text{ is linear in } r^\dagger \text{ satisfying } (5.6) \\ & \overline{R}\left(r^\dagger\right) \text{ and } \underline{R}\left(r^\dagger\right) \text{ satisfying } (5.5) \\ & \quad \text{are upper and lower bounds of } \overline{\mathcal{T}}\left(r^\dagger\right) \\ & \left[\overline{R}\left(r^\dagger\right), \underline{R}\left(r^\dagger\right)\right] \text{ is in } \underline{\mathcal{U}}\left(\pi^\dagger\right) \text{ satisfying } (5.4) \end{aligned} \tag{5.10}$$

Since $\overline{\mathcal{T}}\left(r^\dagger\right)$ *is a hypercube and* $\underline{\mathcal{U}}\left(\pi^\dagger\right)$ *is a polytope, the fact that the corners of the hypercube are inside the unique Nash set if and only if every element in the hypercube is in the unique Nash set implies that the constraint in* (5.9) *is satisfied. Technically, we only require one corner of the hypercube to be inside the unique Nash polytope, as shown in Figure 5.1, and we leave the details to the proof of Proposition 5.2 in the appendix. Then, because the objective and all of the constraints in* (5.10) *are linear in* $r^\dagger, \overline{R}, \underline{R}$ *and* R^{MLE}*, this problem is a linear program.*

5.3 Offline Attack on a Markov Game

The Unique Nash Set (UN) of a Markov Game

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

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

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

$$\begin{aligned} Q_h(s, \mathbf{a}) &\coloneqq R_h(s, \mathbf{a}) \\ &+ \sum_{s' \in \mathcal{S}} P_h(s' | s, \mathbf{a}) \max_{\pi_1 \in \Delta\mathcal{A}_1} \min_{\pi_2 \in \Delta\mathcal{A}_2} Q_{h+1}(s', \pi), \end{aligned} \tag{5.11}$$

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

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

Given $\mathcal{S}, \mathcal{A}, H$, we denote the set of Q functions by $\mathcal{Q} = \left\{ \{Q_h : \mathcal{S} \times \mathcal{A} \to \mathbb{R}\}_{h=1}^{H} \right\}$. Technically, $\mathcal{Q}$ is not the set of proper Q functions of Markov games since both the reward functions and the transition functions do not have to be proper, and given $Q \in \mathcal{Q}$, we may not be able to construct a Markov game

that induces Q. This choice is made to accommodate both model-based and model-free victims who may or may not estimate the rewards and transitions explicitly from the dataset.

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

Definition 5.7 (Markov Perfect Equilibrium)**.** *A Markov perfect equilibrium (MPE) policy π is a policy such that $\pi_h(s)$ is a Nash equilibrium in the stage game $(\mathcal{A}, Q_h(s))$.*
We define the set of all Markov perfect equilibria policies of a Markov game that induces $Q \in \mathcal{Q}$ by

$$\mathcal{M}(Q) = \{\pi : \pi \text{ is an MPE of a Markov game with } Q \text{ function } Q\}.$$

We note that Nash equilibria for Markov games can also be defined by converting the Markov game into a single normal-form game, but we only consider Markov perfect equilibria since Nash equilibria that are not Markov perfect require coordination and commitment to policies in stage games that are not visited along equilibrium paths, which is not realistic in the MARL setting.

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

Definition 5.8 (Unique Nash)**.** *The unique Nash set of a deterministic Markovian policy π for a Markov game G is the set of Q functions such that π is the unique Markov perfect equilibrium under policy π,*

$$\mathcal{U}(\pi) \coloneqq \mathcal{M}^{-1}(\{\pi\}) = \{Q \in \mathcal{Q} : \mathcal{M}(Q) = \{\pi\}\}. \tag{5.12}$$

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

Theorem 5.1 (Unique Nash Polytope)**.** *For any deterministic policy π,*

$$\begin{aligned}\mathcal{U}(\pi) = \{&Q \in \mathcal{Q} : \pi_h(s) \text{ is a strict NE of } (\mathcal{A}, Q_h(s)), \\ &\forall\, h \in [H], s \in \mathcal{S}\} \\ = \{&Q \in \mathcal{Q} : Q_h(s, (\pi_{1,h}(s), a_2)) < Q_h(s, \pi(s)) \\ &< Q_h(s, (a_1, \pi_{2,h}(s))), \forall\, a_1 \neq \pi_{1,h}(s), \\ &, a_2 \neq \pi_{2,h}(s), h \in [H], s \in \mathcal{S}\}, \end{aligned} \tag{5.13}$$

We show the equivalence between (5.12) and (5.13) in the proof of Theorem 5.1 in the appendix. To avoid working with strict inequalities in (5.13), we again define the ι strict version of the unique Nash polytope.

Definition 5.9 (Iota Strict Unique Nash)**.** *For $\iota > 0$, the ι strict unique Nash set of a deterministic policy π is, $\underline{\mathcal{U}}(\pi; \iota) :=$*

$$\begin{aligned}:= \{&Q \in \mathcal{Q} : Q_h(s, (\pi_{1,h}(s), a_2)) + \iota \leqslant Q_h(s, \pi(s)) \\ &\leqslant Q_h(s, (a_1, \pi_{2,h}(s))) - \iota, \forall\, a_1 \neq \pi_{1,h}(s), \\ &a_2 \neq \pi_{2,h}(s), h \in [H], s \in \mathcal{S}\}. \end{aligned} \tag{5.14}$$

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

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

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

compute the Markov perfect equilibria based on the Q functions estimated from such datasets.

Definition 5.10 (Theory of Mind)**.** *Given a dataset* $D \in \mathcal{D}$*, the theory-of-mind set* $\mathcal{T}(D) \subseteq \mathcal{Q}$ *is the set of Q functions that the victims estimate based on* D *to compute their equilibria. In particular, if the victims learn a policy* π*, then* $\pi \in \bigcup_{Q \in \mathcal{T}(D)} \mathcal{M}(Q)$*.*

Example 5.6 (Theory of Mind for Maximum Likelihood Victims)**.** *To extend Example 5.1 in the Markov game setting, we define* R^{MLE} *the same way and* P^{MLE} *as follows, if* $N_h(s, \boldsymbol{a}) \coloneqq \sum_{k=1}^{K} \mathbb{I}_{\left\{s_h^{(k)} = s, \boldsymbol{a}_h^{(k)} = \boldsymbol{a}\right\}} > 0$*,*

$$R_h^{MLE}(s, \boldsymbol{a}|r) \coloneqq \frac{\sum_{k=1}^{K} r_h^{(k)} \mathbb{I}_{\left\{s_h^{(k)} = s, \boldsymbol{a}_h^{(k)} = \boldsymbol{a}\right\}}}{N_h(s, \boldsymbol{a})} \tag{5.15}$$

$$P_h^{MLE}(s'|s, \boldsymbol{a}) \coloneqq \frac{\sum_{k=1}^{K} \mathbb{I}_{\left\{s_{h+1}^{(k)} = s', s_h^{(k)} = s, \boldsymbol{a}_h^{(k)} = \boldsymbol{a}\right\}}}{N_h(s, \boldsymbol{a})} \tag{5.16}$$

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

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

We can construct Q^{MLE} *based on* R^{MLE} *and* P^{MLE} *according to* (5.11)*, and since all Nash equilibria have the same value for zero-sum games,* Q^{MLE} *is unique for every Markov perfect equilibrium of the Markov game with rewards* R^{MLE} *and transitions* P^{MLE}*. Then we have that* $\mathcal{T}(D)$ *is a singleton* Q^{MLE}*.*

Example 5.7 (Theory of Mind for Confidence Bound Victims)**.** *Given a dataset* $D \in \mathcal{D}$*, if the attacker believes the victims estimate the Markov game by estimating the rewards and transitions within some confidence region around some*

point estimates such as the maximum likelihood estimates, as described in Wu et al. (2023b), then $\mathcal{T}(D)$ *would be a polytope with Q functions induced by the Markov games* $(\mathcal{S}, \mathcal{A}, P, R, H)$ *with* P *and* R *satisfying, for every* $h \in [H], s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}$,

$$R_h(s, \boldsymbol{a}|r) \in \mathcal{C}_h^{(R)}(s, \boldsymbol{a}|r) \tag{5.17}$$

$$\mathcal{C}_h^{(R)}(s, \boldsymbol{a}|r) \coloneqq \left\{ R \in \mathbb{R} : \left| R - \hat{R}_h(s, \boldsymbol{a}|r) \right| \leqslant \rho_h^{(R)}(s, \boldsymbol{a}) \right\},$$

$$P_h(s, \boldsymbol{a}) \in \mathcal{C}_h^{(P)}(s, \boldsymbol{a}) \tag{5.18}$$

$$\mathcal{C}_h^{(P)}(s, \boldsymbol{a}) \coloneqq \left\{ P \in \Delta\mathcal{S} : \left\| P - \hat{P}_h(s, \boldsymbol{a}) \right\|_1 \leqslant \rho_h^{(P)}(s, \boldsymbol{a}) \right\},$$

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

$$Q_h(s, \boldsymbol{a}|r) \in \left[\underline{Q}_h(s, \boldsymbol{a}|r), \overline{Q}_h(s, \boldsymbol{a}|r) \right], \tag{5.19}$$

$$\underline{Q}_h(s, \boldsymbol{a}|r) \coloneqq \min_{R \in \mathcal{C}_h^{(R)}(s, \boldsymbol{a}|r)} R + \min_{P \in \mathcal{C}_h^{(P)}(s, \boldsymbol{a})} \sum_{s' \in \mathcal{S}} P(s') \max_{\pi_1 \in \Delta\mathcal{A}_1} \min_{\pi_2 \in \Delta\mathcal{A}_2} \underline{Q}_{h+1}(s', \pi),$$

$$\overline{Q}_h(s, \boldsymbol{a}|r) \coloneqq \max_{R \in \mathcal{C}_h^{(R)}(s, \boldsymbol{a}|r)} R + \max_{P \in \mathcal{C}_h^{(P)}(s, \boldsymbol{a})} \sum_{s' \in \mathcal{S}} P(s') \max_{\pi_1 \in \Delta\mathcal{A}_1} \min_{\pi_2 \in \Delta\mathcal{A}_2} \overline{Q}_{h+1}(s', \pi).$$

We omit Example 5.2 and Example 5.3 for Markov games since the constructions are identical, except it is done for every stage game. As described in Example 5.7, we define $\hat{Q}_h(s, \mathbf{a}|r) \coloneqq \frac{1}{2}\left(\overline{Q}_h(s, \mathbf{a}|r) + \underline{Q}_h(s, \mathbf{a}|r)\right)$ and $\rho_h^{(Q)}(s, \mathbf{a}|r) \coloneqq \frac{1}{2}\left(\overline{Q}_h(s, \mathbf{a}|r) - \underline{Q}_h(s, \mathbf{a}|r)\right)$, and we formally define the outer approximation of the theory-of-mind set for Markov games as fol-

lows.

Definition 5.11 (Outer Approximation of Theory of Mind)**.** *An outer approximation of* $\mathcal{T}(D)$ *is a set denoted by* $\overline{\mathcal{T}}(D)$ *that satisfies* $\mathcal{T}(D) \subseteq \overline{\mathcal{T}}(D)$ *for every* $D \in \mathcal{D}$*, and can be written in the form,*

$$\overline{\mathcal{T}}(D) = \Big\{ Q \in \mathcal{Q} : \Big| Q_h(s, \boldsymbol{a}) - \hat{Q}_h(s, \boldsymbol{a}|r) \Big| \leqslant \rho_h^{(Q)}(s, \boldsymbol{a}|r), \\ \forall\, \boldsymbol{a} \in \mathcal{A}, h \in [H], s \in \mathcal{S} \}, \tag{5.20}$$

for some point estimate $\hat{Q}$ *and radius* $\rho^{(Q)}$*.*

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

The Cheapest Way to Move ToM into UN for Markov Games

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

Definition 5.12 (Attacker's Problem)**.** *The attacker's problem with target policy* $\pi^\dagger$ *is,*

$$\inf_{D^\dagger \in \mathcal{D}(D)} C\left(D, D^\dagger\right) \\ \text{s.t. } \mathcal{T}\left(D^\dagger\right) \subseteq \mathcal{U}\left(\pi^\dagger\right). \tag{5.21}$$

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

Example 5.8 (L_1 Cost Function)**.** *For reward poisoning problem, where* $\mathcal{D}^{(R)}(D)$ *is all possible datasets in the form* $D^\dagger = \left\{ \left\{ \left(s_h^{(k)}, \boldsymbol{a}_h^{(k)}, r_h^{\dagger,(k)} \right) \right\}_{h=1}^{H} \right\}_{k=1}^{K}$ *that*

are modified from $\mathsf{D} = \left\{ \left\{ \left(s_h^{(k)}, \boldsymbol{a}_h^{(k)}, r_h^{(k)} \right) \right\}_{h=1}^{H} \right\}_{k=1}^{K}$*, we define the* L_1 *cost by* $C^{(1)}\left(\mathsf{D}, \mathsf{D}^\dagger\right) = \sum_{k=1}^{K} \sum_{h=1}^{H} \left| r_h^{(k)} - r_h^{\dagger,(k)} \right|$.

We use the same ι strictness relaxation of the unique Nash set and the linear outer approximation of the theory-of-mind set to convert (5.21) into a linear program, which can be solved efficiently. We state this observation as the following theorem.

Theorem 5.2 (Reward Poisoning Linear Program)**.** *Given* $\iota > 0$ *and a linear* $\overline{\mathcal{T}}$*, the following problem is a relaxation of the attacker's reward poisoning problem and can be converted into a linear program,*

$$\begin{aligned} \min_{\mathsf{D}^\dagger \in \mathcal{D}^{(\mathsf{R})}(\mathsf{D})} \; & C^{(1)}\left(\mathsf{D}, \mathsf{D}^\dagger\right) \\ \text{s.t. } & \overline{\mathcal{T}}\left(\mathsf{D}^\dagger\right) \subseteq \underline{\mathcal{U}}\left(\pi^\dagger; \iota\right). \end{aligned} \tag{5.22}$$

Example 5.9 (Maximum Likelihood Centered Linear Program)**.** *In the case* $\hat{\mathsf{R}} = \mathsf{R}^{MLE}$ *and* $\hat{\mathsf{P}} = \mathsf{P}^{MLE}$*, and we construct* $\overline{\mathcal{T}}(\mathsf{D})$ *as described in Example 5.7,* (5.22) *can be converted into a linear program even without explicitly constructing the* $\overline{\mathcal{T}}(\mathsf{D})$ *set. We provide an intuition here and the formal construc-*

tion in the proof of Theorem 5.2,

$$\min_{r^\dagger \in [-b,b]^K} \sum_{k=1}^{K} \sum_{h=1}^{H} \left| r_h^{(k)} - r_h^{\dagger,(k)} \right| \tag{5.23}$$

$$\text{s.t. } R^{MLE}\left(r^\dagger\right) \textit{ is linear in } r^\dagger \textit{ satisfying } (5.15)$$

$$P^{MLE} \textit{ is independent of } r^\dagger \textit{ satisfying } (5.16)$$

$$Q^{MLE}\left(r^\dagger\right) \textit{ satisfying } (5.11)$$

$$\textit{is linear in } R^{MLE}\left(r^\dagger\right) \textit{ thus } r^\dagger$$

$$\overline{Q}\left(r^\dagger\right) \textit{ and } \underline{Q}\left(r^\dagger\right) \textit{ satisfying } (5.19)$$

$$\textit{are upper and lower bounds of } \overline{\mathcal{T}}\left(r^\dagger\right)$$

$$\left[\overline{Q}\left(r^\dagger\right), \underline{Q}\left(r^\dagger\right)\right] \textit{ is in } \underline{\mathcal{U}}\left(\pi^\dagger\right) \textit{ satisfying } (5.14)$$

We move the hypercube $\overline{\mathcal{T}}\left(r^\dagger\right)$ *into the polytope* $\underline{\mathcal{U}}\left(\pi^\dagger\right)$ *by moving one of the corners into the polytope. Note that if* $\overline{Q}$ *and* $\underline{Q}$ *are not constructed directly as linear functions of* $r^\dagger$*, and are computed by* (5.19)*, then these constraints are not linear in* $r^\dagger$*. We avoid this problem by using the dual linear program of* (5.19)*. We present the details in the appendix in the proof of Theorem 5.2. All other constraints are linear in* $r^\dagger$*, and as a result,* (5.23) *is a linear program.*

In the end, we present a sufficient but not necessary condition for the feasibility of (5.22) and (5.21). This condition applies directly to normal-form games with $H = 1$.

Theorem 5.3 (Reward Poisoning Linear Program Feasibility)**.** *For* $\iota > 0$*,* $\mathcal{T}(D)$ *with* $\hat{Q} = Q^{MLE}$*, and* $N_h(s, \boldsymbol{a}) > 0$ *for every* $h \in [H], s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}$ *where either* $a_1 = \pi_{1,h}^\dagger(s)$ *or* $a_2 = \pi_{2,h}^\dagger(s)$*, the attacker's reward poisoning problem is feasible if for every* $h \in [H], s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}$*,*

$$\rho_h^{(R)}(s, \boldsymbol{a}) \leqslant \frac{b - \iota}{4H}. \tag{5.24}$$

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

Table 5.1: A Feasible Attack

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

Table 5.2: The original dataset generation distributions

To construct a feasible attack under (5.24), we use the poisoned rewards similar to the one shown in Table 5.1, which is an example where each agent has three actions and the target action profile being action $(1,1)$. With this $r^\dagger$, the maximum likelihood estimate of the game has a unique Nash equilibrium $\pi_h^\dagger(s)$ with a value of 0 in every stage (h,s). Furthermore, if either the radius of rewards or the radius of Q functions for the theory-of-mind set is less than $\frac{b-\iota}{4H}$, we can show inductively that $\pi_h^\dagger(s)$ remains the unique Nash equilibrium in every stage (h,s), thus showing that every Q function in the theory-of-mind set is also in the unique Nash set, which means the attack is feasible. The complete proof is in the appendix.

5.4 Experiments

Rock Paper Scissors

We start with a simple toy dataset for the Rock Paper Scissors (RPS) game, shown in Table 5.3 with partial coverage, where each entry appears once in the dataset, and the target action profile is $\pi^\dagger = (R, R)$, leading to a tie.

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

Table 5.3: The RPS game.

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

Table 5.4: The original dataset.

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

Table 5.5: The poisoned dataset.

Given the original dataset with 5 entries described in Table 5.4, our algorithm with $\rho = 0$ and $\iota = 0.01$ leads to the poisoned dataset described

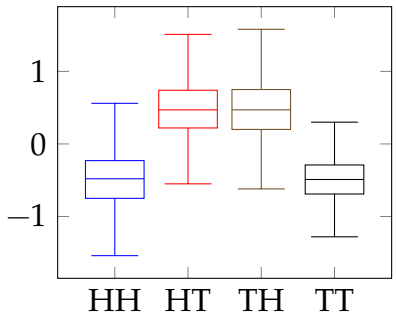


Figure 5.2: The original distribution of rewards

in Table 5.5. The attack cost is 2.02, whereas the attack cost from the feasible attack described in Table 5.1 with $b = 1$ is 4. In addition, note that given the partial coverage, the attack described in Wu et al. (2023b) is not feasible due to their full coverage requirement.

Stochastic Matching Penny

We follow up with the matching penny game, which is also the penalty kick game in soccer, and the rewards are usually estimated by random data points. We generate the datasets randomly with Uniform distributions summarized in Table 5.2. The attacker would like to install a target action profile of (H, H), and in the context of the penalty kick game, the attacker's motivation might be to increase or decrease the total number of goals.

We summarize the before-vs-after box plots in Figure 5.2 for the $n = 100$ case. The cost comparison of our attack, the feasible attack in Table 5.1 with $b = 1$, and the Dominant Strategy Equilibrium (DSE) attack in Wu et al. (2023b), is given in Table 5.6.

5.5 Conclusion

We discuss a few extensions. Faking a unique mixed strategy Nash equilibrium is in general impossible due to the sensitivity of mixing probabilities

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

Table 5.6: Cost comparison between different attacks

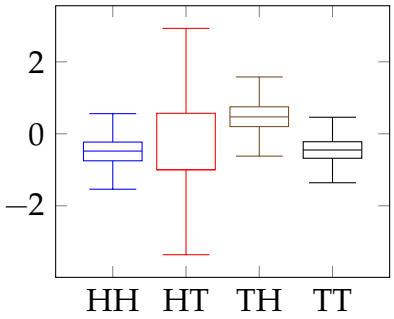


Figure 5.3: The distribution of poisoned rewards

from small perturbations of the reward function, and as long as the theory-of-mind set has non-zero volume, it is impossible to install a mixed strategy profile (or stochastic policy for Markov games) as the unique equilibrium. Faking a unique optimal policy for single-agent reinforcement learners can be easily adapted from our linear program (5.22). Faking a unique coarse correlated equilibrium in every stage game is equivalent to our problem as well since for a two-player zero-sum game, a policy is the unique Markov perfect coarse correlated equilibrium if and only if it is the unique Markov perfect Nash equilibrium.

In the next chapter, we study the problem of installing a stochastic Markov perfect Nash equilibrium, which is impossible for data poisoning with reward uncertainty, as discussed in this chapter, and as a result, we focus on the planning setting instead. We provide characterization of uniqueness of a mixed strategy Nash equilibrium in the a zero-sum game, and we formulate the attacker's reward poisoning problem as a convex optimization problem, and derive similar efficiency and feasibility results

like the ones in this chapter.

6 PLANNING SETTING, REWARD POISONING FOR ZERO-SUM GAMES TO INSTALL A MIXED-STRATEGY NASH EQUILIBRIUM

Contribution Statement. This chapter is a joint work with Jeremy McMahan, Yiding Chen, Yudong Chen, Jerry Zhu, and Qiaomin Xie. I am the main author. My contribution includes the statements and proofs of the main theorems 2 and 3, their corollaries, the main algorithms, and the writing of the paper.

6.1 Introduction

In this chapter, we study the offline data poisoning problem in a zero-sum Markov game environment, where a single attacker tries to minimally modify the rewards so that learners that compute the Markov perfect Nash equilibrium of a Markov game based on the reward matrices provided by the attacker would find a stochastic target joint policy as the unique equilibrium. We formulate the attacker's reward poisoning problem as a convex program, for which we provide an efficient relax-and-perturb algorithm to solve for a near-optimal feasible solution, and we provide sufficient conditions for the feasibility of such an attack.

Consider a two-player zero-sum Markov game $G^\circ = (R^\circ, P^\circ)$ with payoff matrices R° and transition probability matrices P°. Let $\mathcal{S}$ be the finite state space, $\mathcal{A}_i$ the finite set of actions for player $i \in \{1, 2\}$, and H is the horizon. It is well known that such a game has at least one Markov Perfect (Nash) Equilibrium (MPE)[1] $(\mathbf{p}^\circ, \mathbf{q}^\circ)$, where $\mathbf{p}^\circ$ is the Markov policy for player 1 and $\mathbf{q}^\circ$ for player 2 (Maskin and Tirole, 2001). Furthermore,

[1] In the special case where the Markov game has $H = 1$ stage, it reduces to a matrix normal form game; the Markov Perfect Equilibrium reduces to a Nash Equilibrium (NE).

all the MPEs of G° have the same game value v°, which is the expected payoff for player 1 and loss for player 2 at equilibrium.

There may be reasons for a third party to prefer an outcome with a different MPE and/or game value. For instance, a **benevolent** third party may want to achieve fairness. Many games are unfair in that $v^\circ \neq 0$ (an example, two-finger Morra, is given in the experiment section). The third party can modify the payoffs R° into R such that the new game given to the players is fair with value $v = 0$. Similarly, many games have non-intuitive MPEs, and players with bounded rationality (e.g., average people) may fail to find them. For the benefit of such players, the third party may seek a new game whose MPE $(\mathbf{p}, \mathbf{q})$ is an intuitive strategy profile, such as uniform randomization among actions.

In addition, one often desires an MPE consisting of stochastic policies (i.e., a *mixed* strategy equilibrium). If actions represent resources (roads, advertisement slots, etc), the game designer might want all resources to be utilized; if actions represent customers, requests or demand, the designer might want all of them to be served; if a board/video game is concerned, the designer might want the agents to take diverse actions so that the game is more entertaining. Conversely, a **malicious** third party may want to trick the players into playing an MPE $(\mathbf{p}, \mathbf{q})$ of its choice. As most games have mixed equilibria, the players may get suspicious if the modified game turns out to have a pure strategy MPE, whereas a mixed equilibrium is harder to detect. Furthermore, the adversary may want to control the game value v to favor one player over the other—this is the analogue of adversarial attacks in supervised learning. Regardless of intention, such modification typically incurs a cost to the third party, who seeks to minimize it. We assume that the cost is measured by an appropriate loss function $\ell(R, R^\circ)$ (e.g., $\ell(R, R^\circ) = \|R - R^\circ\|$ for some norm $\|\cdot\|$).

It is important to understand when efficient modification is possible,

and to understand malicious attacks so as to develop effective defense. This motivates us to study the following Game Modification problem.

Definition 6.1 (Game Modification)**.** *A game modification problem is specified by the tuple* $(R^\circ, P^\circ, b, (\mathbf{p}, \mathbf{q}), [\underline{v}, \overline{v}], \ell)$*. Here* R° *and* P° *are the payoff and transition matrices, respectively, of the original Markov game. A valid payoff value must be in* $[-b, b]$*. The third party has in mind an arbitrary (and potentially stochastic) target MPE* $(\mathbf{p}, \mathbf{q})$*, which is typically not the unique MPE of* R°*. The third party also has in mind a target game value range* $[\underline{v}, \overline{v}]$*. It is possible that* $b = \infty$*,* $\underline{v} = -\infty$ *or* $\overline{v} = \infty$*. Game modification is the following optimization problem:*

$$\begin{aligned} \inf_{R} \quad & \ell(R, R^\circ) \\ \textit{s.t.} \quad & (\mathbf{p}, \mathbf{q}) \textit{ is the unique MPE of } (R, P^\circ) \\ & \text{value}(R, P^\circ) \in [\underline{v}, \overline{v}], \quad R \textit{ has entries in } [-b, b]. \end{aligned} \tag{6.1}$$

It is important to require that the modified game (R, P°) has a **unique** MPE. In this case, no matter what solver the players use, they will inevitably find $(\mathbf{p}, \mathbf{q})$ and not some other MPEs of R. Henceforth, we refer to a Markov game simply by its payoff matrices R and suppress reference to the transition matrices P°, which cannot be changed by the third party.

To the best of our knowledge, the Game Modification problem in the generality of Definition 6.1 has not been studied in the literature. The main challenge is to ensure uniqueness of the MPE. We present a complete characterization of games with a unique MPE and give an efficient algorithm to find the solution. We will first study the special case of normal form games in Section 6.3, followed by Markov games in Section 6.4.

6.2 Related Work

Reward modification in single-agent reinforcement learning has been studied in Banihashem et al. (2022); Huang and Zhu (2019); Rakhsha et al. (2021a,b, 2020); Zhang et al. (2020b). In this setting, there always exists a deterministic optimal policy. Generalizing to the multi-agent setting, even in the zero-sum case, involves the additional complication of multiple equilibria and the non-existence of deterministic equilibrium policies.

Adversarial attacks on multi-agent reinforcement learners are studied in Wu et al. (2023c); Ma et al. (2021), who consider the setting where an attacker installs a target dominant strategy equilibrium by modifying the underlying bandit or Markov game. In general, mixed strategies that assign positive probabilities to multiple actions cannot be dominant (they are not dominated by at least one of the actions in the support). Therefore, the approach in Wu et al. (2023c); Ma et al. (2021) cannot be directly applied in our setting targeting at a mixed strategy equilibrium.

Our model is similar to Wu et al. (2023a), where an attacker installs a target Nash equilibrium by poisoning the training data set. Their work requires the target equilibrium to be a deterministic action profile, and they assume the victims estimate confidence regions of the game payoff matrices based on a noisy data set. Since it is in general impossible for all games in the confidence region to have the same mixed strategy Nash equilibrium, the modification goal in our setting is infeasible under their setting. Instead, we consider the problem in which the players are provided with the precise payoff matrix by the game designer so that it is possible to install a mixed strategy as the unique equilibrium of the modified game. For a similar reason, data poisoning techniques in Ma et al. (2019); Rangi et al. (2022b); Zhang and Parkes (2008b); Zhang et al. (2009) are not applicable to our setting.

Monderer and Tennenholtz (2003); Anderson et al. (2010) explore the problem of installing a pure strategy equilibrium while minimizing

the cost of modification, but their method does not directly extend to mixed-strategy equilibria. Previous work also studied games with a specific mixed strategy profile as the unique Nash equilibrium (Millham, 1972; Heuer, 1979; Quintas, 1988). However, these works do not provide conditions that can be easily converted into constraints in an optimization problem for minimizing the modification cost, nor do they provide algorithms for finding such games given a target equilibrium. In our work, we provide new conditions for NE uniqueness, which can be used as constraints in a cost-minimization optimization formulation that can be solved efficiently. Our conditions are related to results on the uniqueness of optimal solutions to linear programs (Mangasarian, 1978; Appa, 2002; Szilágyi, 2006); these results do not provide operational characterizations for our purpose. Our results and algorithms generalize to Markov games, whereas the aforementioned work focuses on normal-form games.

6.3 Modifying Normal Form Games

We begin with matrix normal form games, a special case Markov Game with horizon $H = 1$.

Preliminaries

Consider a finite two-player zero-sum game with action space $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$ and a b-bounded payoff matrix $R \in [-b, b]^{|\mathcal{A}_1| \times |\mathcal{A}_2|}$. When a joint action $(i, j) \in \mathcal{A}_1 \times \mathcal{A}_2$ is played, player 1 receives reward $[R]_{ij}$ and player 2 receives reward $-[R]_{ij}$. Let $(\mathbf{p}, \mathbf{q})$ denote a (possibly mixed) strategy profile, where $\mathbf{p} \in \Delta_{\mathcal{A}_1}$ and $\mathbf{q} \in \Delta_{\mathcal{A}_2}$, with $\Delta_{\mathcal{D}}$ denoting the probability simplex on $\mathcal{D}$. The expected reward for player 1 is given by $\mathbf{p}^\top R \mathbf{q}$.

A standard characterization of Nash Equilibrium is in terms of the lack of incentive for unilateral deviation:

Definition 6.2 (Nash Equilibrium)**.** $(\mathbf{p}, \mathbf{q})$ *is a Nash Equilibrium* (*NE*) *of a game* R *if and only if*

$$\begin{aligned} \mathbf{p}^\top \mathsf{R}\mathbf{q} &\geqslant \mathbf{p}'^\top \mathsf{R}\mathbf{q}, \quad \forall\, \mathbf{p}' \in \Delta_{\mathcal{A}_1}, \\ \mathbf{p}^\top \mathsf{R}\mathbf{q} &\leqslant \mathbf{p}^\top \mathsf{R}\mathbf{q}', \quad \forall\, \mathbf{q}' \in \Delta_{\mathcal{A}_2}. \end{aligned}$$

A finite two-player zero-sum game has at least one NE and possibly more (Nash Jr, 1950). We denote the set of NEs of a game R by

$$\begin{aligned} \mathsf{NE}(\mathsf{R}) := \{ &(\mathbf{p}, \mathbf{q}) \in \Delta_{\mathcal{A}_1} \times \Delta_{\mathcal{A}_2} : \\ &(\mathbf{p}, \mathbf{q}) \text{ is an NE of game } \mathsf{R} \}. \end{aligned}$$

Definition 6.3 (Inverse Nash Function)**.** *Given an arbitrary set of strategy profiles* $\Pi \subset \Delta_{\mathcal{A}_1} \times \Delta_{\mathcal{A}_2}$*, we define the inverse Nash function*

$$\mathsf{NE}^{-1}(\Pi) := \left\{ \mathsf{R} \in [-\mathsf{b}, \mathsf{b}]^{|\mathcal{A}_1| \times |\mathcal{A}_2|} : \mathsf{NE}(\mathsf{R}) = \Pi \right\},$$

which gives games with Π *as the exact set of NEs.*

Example 6.1. *Suppose* $\mathcal{A}_1 = \{0\}, \mathcal{A}_2 = \{1, 2\}$*, and the payoff matrix is* $\mathsf{R} = \begin{bmatrix} \mathsf{R}_{01} & \mathsf{R}_{02} \end{bmatrix}$*. Consider the pure strategy* $(\mathbf{p}, \mathbf{q}) = ((1), (1, 0))$*, where player 1 plays action* 0 *and player 2 plays action* 1*. For the singleton set* $\Pi = \{(\mathbf{p}, \mathbf{q})\}$*, we have* $\mathsf{NE}^{-1}(\Pi) = \{\mathsf{R} \in [-\mathsf{b}, \mathsf{b}]^2 : \mathsf{R}_{01} > \mathsf{R}_{02}\}$*, so there are infinitely many games with* $(\mathbf{p}, \mathbf{q})$ *as the unique NE. However, games like* $\mathsf{R} = [0, 0]$ *are not in* $\mathsf{NE}^{-1}(\Pi)$*; see next example.*

Example 6.2. *Under the same setting as above, let* $\Pi = \{((1), \mathbf{q}) : \mathbf{q} \in \Delta_{\mathcal{A}_2}\}$*, which is the set of all strategy profiles. Then* $\mathsf{NE}^{-1}(\Pi) = \{\mathsf{R} \in [-\mathsf{b}, \mathsf{b}]^2 : \mathsf{R}_{01} = \mathsf{R}_{02}\}$*.*

Example 6.3. *Continuing, let* $(\mathbf{p}, \mathbf{q}) = \left((1), \left(\frac{1}{2}, \frac{1}{2}\right)\right)$ *and* $\Pi = \{(\mathbf{p}, \mathbf{q})\}$*. Then* $\mathsf{NE}^{-1}(\Pi) = \emptyset$*. To see this, note tthat the second inequality in Definition 6.2 implies* $\frac{1}{2}\mathsf{R}_{01} + \frac{1}{2}\mathsf{R}_{02} \leqslant \mathsf{q}'\mathsf{R}_{01} + (1 - \mathsf{q}')\mathsf{R}_{02}$ *for all* $\mathsf{q}' \in [0, 1]$*. Setting* $\mathsf{q}' = 0$

and $\mathsf{q}' = 1$ *separately, we obtain* $\mathsf{R}_{01} = \mathsf{R}_{02}$. *However, the last example shows that* $(\mathbf{p}, \mathbf{q})$ *is not the unique NE of such games.*

As mentioned in the Introduction, the game designer seeks games with a given $(\mathbf{p}, \mathbf{q})$ as the unique NE, that is, games in the set $\mathsf{NE}^{-1}(\{(\mathbf{p}, \mathbf{q})\})$. However, the last example shows that for certain $(\mathbf{p}, \mathbf{q})$, the set $\mathsf{NE}^{-1}(\{(\mathbf{p}, \mathbf{q})\})$ may be empty. We will soon completely characterize when this set is nonempty: it turns out $\mathbf{p}$ and $\mathbf{q}$ must have equal support sizes.

To this end, we exploit the well-established connection between Nash equilibrium and linear program duality. In particular, any $(\mathbf{p}, \mathbf{q}) \in \mathsf{NE}(\mathsf{R})$ is an optimal solution pair to the following pair of primal-dual linear programs (LPs), and vice versa (Dantzig, 1963).

Definition 6.4 (Linear Programs for NE)**.**

$$\begin{aligned} \text{(Primal)} \quad & \max_{\mathbf{p}' \in \Delta\mathcal{A}_1, \nu} \ \nu \\ & \text{s.t. } \mathbf{p}'^{\top} \mathsf{R} \geqslant \nu 1_{|\mathcal{J}|}^{\top} \end{aligned} \tag{6.2}$$

$$\begin{aligned} \text{(Dual)} \quad & \min_{\mathbf{q}' \in \Delta\mathcal{A}_2, \nu} \ \nu \\ & \text{s.t. } \mathsf{R}\mathbf{q}' \leqslant \nu 1_{|\mathcal{I}|} \end{aligned} \tag{6.3}$$

The inequalities are elementwise.

The optimal values of the two linear programs both equal ν^*, the value of the game.

We emphasize that these LPs are used only for characterizing the properties of $\mathsf{NE}(\mathsf{R})$ and its uniqueness. We do *not* assume that the players must use LP to find an NE: they can use any other solvers and may find any one of the NEs if there are multiple ones. This reflects how NE solvers typically work in practice.

Necessary and Sufficient Conditions for Unique NE

As mentioned, a key question in Game Modification is to characterize when the NE is unique. We provide a complete and operational answer to this question.

For a given strategy profile $(\mathbf{p}, \mathbf{q})$, let $\mathcal{I} = \mathrm{supp}(\mathbf{p})$, $\mathcal{J} = \mathrm{supp}(\mathbf{q})$ denote the supports. Denote by $\mathbf{e}_i$ the canonical (one-hot) vector in appropriate dimension corresponding to the ith action. Our characterization of NE uniqueness is based on two conditions.

Condition 6.1 (SIISOW: Switch-In Indifferent, Switch-Out Worse)**.** *A game* R *satisfies SIISOW with respect to* $(\mathbf{p}, \mathbf{q})$ *if*

$$\mathbf{e}_i^\top R\mathbf{q} = \mathbf{p}^\top R\mathbf{q}, \quad \forall i \in \mathcal{I}, \tag{6.4}$$

$$\mathbf{p}^\top R\mathbf{e}_j = \mathbf{p}^\top R\mathbf{q}, \quad \forall j \in \mathcal{J}, \tag{6.5}$$

$$\mathbf{e}_i^\top R\mathbf{q} < \mathbf{p}^\top R\mathbf{q}, \quad \forall i \notin \mathcal{I}, \tag{6.6}$$

$$\mathbf{p}^\top R\mathbf{e}_j > \mathbf{p}^\top R\mathbf{q}, \quad \forall j \notin \mathcal{J}. \tag{6.7}$$

Let us parse this condition. If the strict inequalities above were changed to weak inequalities, then the four equations would be equivalent to the Definition 6.2 of an NE (Osborne, 2004). Therefore, the SIISOW condition implies that $(\mathbf{p}, \mathbf{q})$ is an NE of R. Moreover, given that one player plays at this NE, if the other player switches to any pure strategy outside its NE support, its reward will be *strictly* worse by equations (6.6) and (6.7) ("switch-out worse"); if the other player uses any pure strategy within its support, it will achieve the same game value by equations (6.4) and (6.5) (known as the "switch-in indifference" principle).

To state the second condition, some notations are needed. We use $[R]_{\mathcal{I}\mathcal{J}}$ or $R_{\mathcal{I}\mathcal{J}}$ to denote the $|\mathcal{I}| \times |\mathcal{J}|$ submatrix of R with rows in $\mathcal{I}$ and columns in $\mathcal{J}$. We write $R_{\mathcal{I}\bullet}$ for the $|\mathcal{I}| \times |\mathcal{A}_2|$ submatrix with rows in $\mathcal{I}$, and $R_{\bullet\mathcal{J}}$ for the $|\mathcal{A}_1| \times |\mathcal{J}|$ submatrix with columns in $\mathcal{J}$. Denotes by $1_{|\mathcal{I}|}$ the $|\mathcal{I}|$-dimensional

all-one vector.

Condition 6.2 (INV: Invertability)**.** *A game* R *satisfies INV with respect to* $(\mathbf{p}, \mathbf{q})$ *if the matrix* $\begin{bmatrix} \mathsf{R}_{\mathcal{J}\mathcal{J}} & -1_{|\mathcal{J}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix}$ *is invertible.*

We now present the first main theorem of this paper: a sufficient and necessary condition for a game R to admit a given $(\mathbf{p}, \mathbf{q})$ as the unique NE.

Theorem 6.1 (Uniqueness of NE)**.** $\mathsf{R} \in \mathsf{NE}^{-1}(\{(\mathbf{p}, \mathbf{q})\})$ *if and only if* R *satisfies both SIISOW (Condition 6.1) and INV (Condition 6.2) with respect to* $(\mathbf{p}, \mathbf{q})$.

With Theorem 6.1, the Game Modification problem (6.1) for a normal form game can be instantiated as an optimization problem with linear and spectral constraints as follows.

Definition 6.5 (Game Modification for Two-Player Zero-Sum Normal Form Game)**.** *Given the cost function* ℓ*, the target policy* $(\mathbf{p}, \mathbf{q})$ *with supports* $\mathcal{I}, \mathcal{J}$*, and target game value range* $[\underline{v}, \bar{v}]$*, the game modification problem for normal form games can be written as the following optimization problem:*

$$
\begin{aligned}
&\inf_{\mathsf{R}, v} \ \ell(\mathsf{R}, \mathsf{R}^\circ) \\
&\text{s.t. } \mathsf{R}_{\mathcal{I}\bullet}\mathbf{q} = v1_{|\mathcal{I}|} && \text{[row SII]} \\
&\quad \mathbf{p}^\top \mathsf{R}_{\bullet\mathcal{J}} = v1_{|\mathcal{J}|}^\top && \text{[column SII]} \\
&\quad \mathsf{R}_{\mathcal{A}_1\setminus\mathcal{I}\bullet}\mathbf{q} < v1_{|\mathcal{A}_1\setminus\mathcal{I}|} && \text{[row SOW]} \\
&\quad \mathbf{p}^\top \mathsf{R}_{\bullet\mathcal{A}_2\setminus\mathcal{J}} > v1_{|\mathcal{A}_2\setminus\mathcal{J}|}^\top && \text{[column SOW]} \\
&\quad \sigma_{\min}\left(\begin{bmatrix} \mathsf{R}_{\mathcal{I}\mathcal{J}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix}\right) > 0 && \text{[INV]} \\
&\quad \underline{v} \leqslant v \leqslant \bar{v} && \text{[value range]} \\
&\quad -b \leqslant \mathsf{R}_{ij} \leqslant b, \forall (i,j) \in \mathcal{A} && \text{[reward bound]}
\end{aligned}
\tag{6.8}
$$

where $\sigma_{\min}(\cdot)$ *denotes the smallest singular value.*

Feasibility of Game Modification

We now show that Game Modification in normal form games, as formulated in Definition 6.5, is feasible as long as $|\mathcal{I}| = |\mathcal{J}|$ and the intervals $[\underline{v}, \overline{v}]$ and $(-b, b)$ has non-empty intersection. Our proof is constructive. We present a special matrix game, which we call the Extended Rock-Paper-Scissors (eRPS), with the desired $(\mathbf{p}, \mathbf{q})$ as the unique NE. This game can be defined for arbitrary strategy space sizes $|\mathcal{A}_1|$ and $|\mathcal{A}_2|$. The standard rock paper scissors game is a special case when the sizes are 3, hence the name.

Definition 6.6 (Extended Rock-Paper-Scissors Game)**.** *Given strategy spaces $\mathcal{A}_1, \mathcal{A}_2$, and target strategy profile $(\mathbf{p}, \mathbf{q}) \in \Delta_{\mathcal{A}_1} \times \Delta_{\mathcal{A}_2}$ with equal supports $\mathcal{I} = \mathcal{J} = \{0, \ldots, k-1\}$, where $1 \leqslant k \leqslant \min(|\mathcal{A}_1|, |\mathcal{A}_2|)$, the Extended Rock Paper Scissors Game $R^{\text{eRPS}(\mathbf{p},\mathbf{q})}$ is:*

$$R_{ij}^{\text{eRPS}(\mathbf{p},\mathbf{q})} = \begin{cases} -\dfrac{c}{\mathbf{p}_i \mathbf{q}_j} & \text{if } {}^{k>1, i,j<k}_{j=(i+1) \bmod k} \\ \dfrac{c}{\mathbf{p}_i \mathbf{q}_j} & \text{if } {}^{k>1, i,j<k}_{j=(i+2) \bmod k} \\ 1 & \text{if } i < k, j \geqslant k \\ -1 & \text{if } i \geqslant k, j < k \\ 0 & \text{otherwise ,} \end{cases} \tag{6.9}$$

where $c = \min_{i \in \mathcal{I}} \left(\mathbf{p}_i \mathbf{q}_{(i+1 \bmod k)}, \mathbf{p}_i \mathbf{q}_{(i+2 \bmod k)} \right)$ is a normalizing constant ensuring that all the entries of R^{eRPS} are between -1 and 1.

For support size $k = 1$, namely $(\mathbf{p}, \mathbf{q})$ is a pure strategy profile, the R^{eRPS} game is visualized in Table 6.1. It is easy to check that the upper left corner $(0, 0)$ is indeed the unique pure Nash equilibrium.

For support size $k \geqslant 2$, namely $(\mathbf{p}, \mathbf{q})$ is a mixed strategy profile, the R^{eRPS} game is visualized in Table 6.1. As a special case, for $\mathbf{p} = \mathbf{q} = (1/3, 1/3, 1/3)$, R^{eRPS} is the standard Rock-Paper-Scissors game.

0	1	...	1
−1	0	...	0
...	...	...	
−1	0	...	0

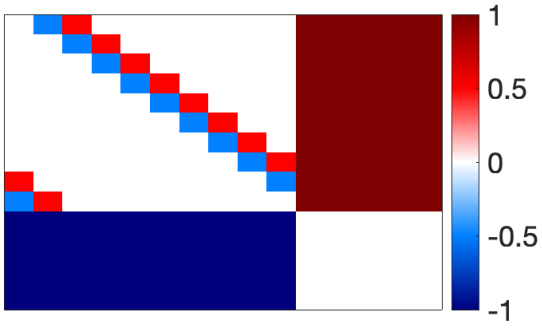


Table 6.1: R^{eRPS} when $k = 1$ (left) and $k \geqslant 2$ (right).

Lemma 6.1. *Given any* $(\mathbf{p}, \mathbf{q})$ *with equal support sizes, the Extended Rock-Paper-Scissors Game* $R^{\text{eRPS}(\mathbf{p},\mathbf{q})}$ *in Definition 6.6 has* $(\mathbf{p}, \mathbf{q})$ *as the unique Nash equilibrium, and its game value is* 0.

Note that applying any positive affine transformation to the reward matrix preserves the set of Nash equilibria of the game (Tewolde, 2023). Therefore, if we want the game R to be bounded between $[-b, b]$ for $b > 0$, we can simply scale R^{eRPS} by b. More generally, for each $\iota > 0$ and $\nu \in \mathbb{R}$, the game $\iota R^{\text{eRPS}} + \nu$ has entries in $[\nu - \iota, \nu + \iota]$ and $(\mathbf{p}, \mathbf{q})$ as the unique Nash equilibrium with value ν.

Combining the above observation and Theorem 6.1, we provide a complete characterization of when there exists a game admitting a given strategy profile $(\mathbf{p}, \mathbf{q})$ as the unique NE and satisfying the given value and reward bounds.

Theorem 6.2 (Feasibility of Game Modification)**.** *The Game Modification problem in Definition 6.5 for normal-form games is feasible if and only if* $(\mathbf{p}, \mathbf{q})$ *satisfies* $|\mathcal{I}| = |\mathcal{J}|$ *and* $(-b, b) \cap [\underline{v}, \bar{v}] \neq \emptyset$.

Here, the equal-support condition $|\mathcal{I}| = |\mathcal{J}|$ arises due to the INV condition, which requires $R_{\mathcal{I}\mathcal{J}}$ to be a square matrix. The game value cannot equal b or $-b$ due to the SIISOW condition, which requires a strictly posi-

tive gap between the value of the game and the value of the off-support actions. The complete proof is provided in the appendix.

An Efficient Algorithm for Game Modification in Normal Form Games

We now describe an efficient algorithm to approximately solve Game Modification in normal form games. Thanks to Theorem 6.1, the unique NE requirement $R \in \mathsf{NE}^{-1}(\{(\mathbf{p}, \mathbf{q})\})$ can be fulfilled by the equivalent SIISOW and the INV conditions, as done in the Game Modification formulation in Definition 6.5. If we ignore the INV condition therein for a moment and tighten the strict inequalities, we obtain an optimization problem with linear constraints:

$$\min_{R} \ \ell(R, R^{\circ}) \tag{6.10a}$$

$$\text{s.t. } R_{\mathcal{J}\bullet}\mathbf{q} = \nu 1_{|\mathcal{J}|} \tag{6.10b}$$

$$\mathbf{p}^{\top} R_{\bullet\mathcal{J}} = \nu 1_{|\mathcal{J}|}^{\top} \tag{6.10c}$$

$$R_{\mathcal{A}_1\setminus\mathcal{J}\bullet}\mathbf{q} \leqslant (\nu - \iota)\, 1_{|\mathcal{A}_1\setminus\mathcal{J}|} \tag{6.10d}$$

$$\mathbf{p}^{\top} R_{\bullet\mathcal{A}_2\setminus\mathcal{J}} \geqslant (\nu + \iota)\, 1_{|\mathcal{A}_2\setminus\mathcal{J}|}^{\top} \tag{6.10e}$$

$$\underline{\nu} \leqslant \nu \leqslant \bar{\nu} \tag{6.10f}$$

$$-b + \lambda \leqslant R_{ij} \leqslant b - \lambda, \forall (i,j) \in \mathcal{A}. \tag{6.10g}$$

In the above, the first four constraints (6.10b)–(6.10e) encode the SIISOW condition. Notice we introduced a small SIISOW margin parameter $\iota > 0$ in (6.10d), (6.10e). This is a tightening of the strict inequalities in Definition 6.5 and ensures that the feasible set is closed. A margin λ is also added to the reward bound (6.10g) for reasons that would become clear momentarily.

One can readily solve the program (6.10) for a solution R. To ensure R has a unique NE, it remains to satisfy the INV condition, i.e., the ma-

trix $\begin{bmatrix} R_{\mathcal{I}\mathcal{J}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix}$ must be invertible. However, enforcing INV directly by constraining the smallest singular value of the matrix leads to a nonlinear, nonconvex optimization problem that is difficult to solve.

We adopt an alternative approach: we take the solution R' to the program (6.10)—which may not satisfy the INV condition—and add a small special random matrix to R' in such a way that: (1) the resulting matrix R is invertible with probability 1; (2) R still has $(\mathbf{p}, \mathbf{q})$ as its unique NE and satisfies the value constraint $v \in [\underline{v}, \overline{v}]$ in (6.10f). Note that by introducing a small margin λ in the reward bound (6.10g) and using a sufficiently small perturbation, we ensure that the perturbed rewards remain in the original designated range $[-b, b]$. Specifically, the matrix we add is $\varepsilon R^{\text{eRPS}}$, where ε is a random number in $[-\lambda, \lambda]$ and R^{eRPS} the Extended Rock-Paper-Scissors game matrix, which has entries in $[-1, 1]$.

Putting together the above ingredients, we have approximately solved the Game Modification problem, provably satisfying the constraints with probability 1 and achieving a near minimal cost $\ell(R, R^\circ)$ as long as the random perturbation is small (Proposition 6.1).

We present the complete procedure, Relax And Perturb (RAP), in Algorithm 4.

Algorithm 4 Relax And Perturb (RAP)

Input: original game R°, cost function ℓ, target policy $(\mathbf{p}, \mathbf{q})$, target value range $[\underline{v}, \overline{v}]$, reward bound $b \in \mathbb{R}^+ \cup \{\infty\}$.
Parameters: margins $\iota \in \mathbb{R}^+$ and $\lambda \in \mathbb{R}^+$.
Output: modified game R.

1: Solve the problem (6.10). Call the solution R'.
2: Sample $\varepsilon \sim \text{uniform}[-\lambda, \lambda]$
3: Return $R = R' + \varepsilon R^{\text{eRPS}(\mathbf{p},\mathbf{q})}$.

When the cost function ℓ is convex, the problem (6.10) is a convex program with linear constraints, for which efficient solvers exist (Wright,

2006). When ℓ is piecewise linear, (6.10) is further reduced to a linear program, as shown in the following examples.

Example 6.4 (L^1 Cost)**.** *One may measure the cost of modifying the game from* R° *to* R *by the* L^1 *norm* $\|R - R^\circ\|_1$*; explicitly,*

$$\ell(R, R^\circ) = \sum_{i \in \mathcal{A}_1, j \in \mathcal{A}_2} \left| R_{ij} - R^\circ_{ij} \right|.$$

Example 6.5 (Occupancy Weighted Cost)**.** *If the cost of modifying an entry is proportional to how often it is visited by the players at Nash equilibrium, we can use the following cost function:*

$$\ell(R, R^\circ) = \sum_{i \in \mathcal{A}_1, j \in \mathcal{A}_2} \mathbf{p}_i \mathbf{q}_j \left| R_{ij} - R^\circ_{ij} \right|. \tag{6.11}$$

Note that it is costless to modify the entries outside the product of the supports of $\mathbf{p}, \mathbf{q}$*. Applications of this weighted cost include online reward poisoning in multi-agent reinforcement learning, where an attacker pays for the modified reward entry only when the corresponding action profile is used by the online learners.*

Below we show that the RAP Algorithm has the desired feasibility and near optimality properties.

Proposition 6.1 (Feasibility and Optimality of RAP Algorithm)**.** *Suppose that the parameters* ι, λ *of Algorithm 4 satisfy* $\lambda + \iota < \min\{b + \overline{v}, b - \underline{v}\}$ *and let* $R(\iota, \lambda) = R' + \varepsilon R^{\text{eRPS}}$ *be the output of the algorithm. The following hold.*

- (***Existence***) *The solution* R' *to* (6.10) *exists.*
- (***Feasibility***) *With probability 1,* $R(\iota, \lambda)$ *is feasible for the original Game Modification problem in Definition 6.5.*
- (***Optimality***) *Assume in addition that the cost function* ℓ *is Lipschitz with constant* L*, (i.e.* $|\ell(R, R^\circ) - \ell(R', R^\circ)| \leqslant L \|R - R'\|_1, \forall R, R'$.*) Then*

$R(\iota, \lambda)$ *is near-optimal with respect to the optimal objective value* $C^\star$ *in Definition 6.5, in the sense that* $\lim_{\max\{\iota,\lambda\}\to 0} \ell\left(R(\iota,\lambda), R^\circ\right) = C^\star$.

In the result above, existence follows from Theorem 6.2. Feasibility holds because the matrix sum

$$\begin{bmatrix} R'_{\mathcal{IJ}} & -1_{|\mathcal{I}|} \\ 1^\top_{|\mathcal{J}|} & 0 \end{bmatrix} + \varepsilon \begin{bmatrix} R^{\text{eRPS}}_{\mathcal{IJ}} & -1_{|\mathcal{I}|} \\ 1^\top_{|\mathcal{J}|} & 0 \end{bmatrix}$$

is invertible with probability 1, as ε is a continuous random variable and the second matrix above is invertible. To prove optimality, we take a feasible solution $R^{(\varepsilon)}$ to the original game modification problem (6.8) with a cost at most $C^\star + \varepsilon$, and then slightly modify its entries to get a new solution $R'^{(\varepsilon)}$ so that (i) the reward bound (6.10g) with λ margin is satisfied, (ii) the SIISOW properties (6.10b)–(6.10e) are preserved, and (iii) the game value is the same. The costs of $R'^{(\varepsilon)}$ and $R^{(\varepsilon)}$ are close thanks to the Lipschitz property of the cost. In particular, we show that $\ell\left(R'^{(\varepsilon)}, R^\circ\right) = \ell\left(R^{(\varepsilon)}, R^\circ\right) + O\left(\max\{\iota, \lambda\}\right)$, hence the cost suboptimality can be made small by using small margins ι, λ. We provide the details in the appendix.

6.4 Markov Games Modification

In this section, we generalize our algorithm and theoretical results to Markov games.

Preliminaries

A finite-horizon two-player zero-sum Markov game can be described by a pair (P, R), given the finite state space $\mathcal{S}$, the finite joint action space $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$, and horizon H. Here $P = \left\{P_h : \mathcal{S} \times \mathcal{S} \to [0,1]^{|\mathcal{A}_1|\times|\mathcal{A}_2|}\right\}_{h=1}^{H}$ is the transition probabilities, $P_0 : \mathcal{S} \to [0,1]$ is the initial state distribution,

and $R = \left\{R_h : S \to [-b, b]^{|\mathcal{A}_1| \times |\mathcal{A}_2|}\right\}_{h=1}^{H}$ is the mean reward function. In particular, for each $h \in [H]$, $s \in S$, we treat $R_h(s)$ as an $|\mathcal{A}_1| \times |\mathcal{A}_2|$ matrix, where $[R_h(s)]_{ij}$ is the reward when joint action profile $(i, j) \in \mathcal{A}_1 \times \mathcal{A}_2$ is applied. Similarly, the transition probabilities are given by a $|\mathcal{A}_1| \times |\mathcal{A}_2|$ matrix $P_h(s'|s)$, where $[P_h(s'|s)]_{ij}$ is the probability of transitioning from state $s \in S$ in period $h \in [H]$ to state $s' \in S$ when the joint action profile (i, j) is used. $P_0(s)$ is the probability that the game starts in state $s \in S$. The above matrix representations are chosen to follow the convention used in the last section for normal form matrix games.

A Markovian policy $(\mathbf{p}, \mathbf{q})$ consists of a pair of policies for the two players: $\mathbf{p} = \{\mathbf{p}_h : S \to \Delta_{\mathcal{A}_1}\}_{h=1}^{H}$ and $\mathbf{q} = \{\mathbf{q}_h : S \to \Delta_{\mathcal{A}_2}\}_{h=1}^{H}$. Here $\mathbf{p}_h(s)$ and $\mathbf{q}_h(s)$ are probability vectors; in period $h \in [H]$, state $s \in S$, $[\mathbf{p}_h(s)]_i$ specifies the probability that player 1 takes action $i \in \mathcal{A}_1$, and $[\mathbf{q}_h(s)]_j$ specifies the probability that player 2 takes action $j \in \mathcal{A}_2$.

Remark 6.1. *A Markovian policy above is also called a behavioral policy, where the two players use independent randomization in each stage game. This is in contrast to so-called mixed strategies, where the players randomize upfront (before period 0) among multiple deterministic policies. Due to Kuhn's Theorem for Markov games (Heinrich, Lanctot, and Silver, 2015; Lu and Yan, 2020), these two class of policies are payoff-equivalent: given a mixed strategy, we can find a behavioral strategy that leads to the same expected total rewards for both players. Consequently, we focus on the setting where the target policy $(\mathbf{p}, \mathbf{q})$ is given as a Markovian policy, omiting the generalization to mixed policies.*

Each zero-sum Markov game has at least one Markov perfect equilibrium and a unique Nash value. The action-value or Q function of the MPE, denoted by $Q^\star$, satisfies the following Bellman equations: for each

$h \in [H], s \in \mathcal{S}, (i,j) \in \mathcal{A}$,

$$Q_h^\star(s,(i,j)) \coloneqq R_h(s,(i,j)) + \sum_{s' \in \mathcal{S}} P_h(s'|s,(i,j)) \max_{p' \in \Delta_{\mathcal{A}_1}} \min_{q' \in \Delta_{\mathcal{A}_2}} Q_{h+1}^\star(s',(p',q')), \tag{6.12}$$

where for a possibly stochastic strategy profile $(p',q') \in \Delta_{\mathcal{A}_1} \times \Delta_{\mathcal{A}_2}$, we define

$$Q_h^\star(s,(p',q')) \coloneqq \sum_{i \in \mathcal{A}_1, j \in \mathcal{A}_2} p_i' q_j' Q_h^\star(s,(i,j)). \tag{6.13}$$

We use the convention $Q_{H+1}^\star(s,(i,j)) = 0, \forall s,i,j$.

Under an MPE policy, the stage game of the Markov game in each period $h \in [H]$ and state $s \in \mathcal{S}$ is a normal form game with payoff matrix $\mathbb{Q}_h(s)$, where

$$[\mathbb{Q}_h(s)]_{ij} \coloneqq Q_h^\star(s,(i,j)) \tag{6.14}$$

gives the payoff under the action profile $(i,j) \in \mathcal{A}$. Consequently, an MPE can be defined recursively as the Nash equilibrium for every stage game.

Definition 6.7 (Markov Perfect Equilibrium)**.** *A Markov perfect equilibrium policy* $(\mathbf{p},\mathbf{q})$ *is a policy that satisfies, for every* $h \in [H], s \in \mathcal{S}$,

$$(\mathbf{p}_h(s),\mathbf{q}_h(s)) \in \mathsf{NE}(\mathbb{Q}_h(s)),$$

where $\mathbb{Q}_h(s)$ *is defined by equations* (6.12)–(6.14).

An alternative approach to study the equilibria of a Markov games is by converting it to a single big normal-form game and considering the NEs of the latter. A Nash equilibrium defined in this way is in general not Markov perfect—it requires coordination and commitment to policies in stage games that are not visited along equilibrium paths. Such policies are often not realistic. Moreover, it is computationally intractable to manipulate such a big normal-form game. Therefore, we focus on MPE and make use

of its recursive characterization through Bellman equations.

An Efficient Algorithm for Game Modification in Markov Games

A two-player zero-sum Markov game has a unique MPE if and only if every stage game $\mathbb{Q}_h(s)$ has a unique NE. Our results on the uniqueness of NE for normal form games (Theorem 6.1) apply to each stage game of the Markov game. Combining these two observations and the Bellman equations for $\mathbb{Q}_h(s)$'s, the Game Modification problem (Definition 6.1) can be instantiated to a Markov game as an optimization problem similar to (6.8), where SIISOW (Condition 6.1), INV (Condition 6.2) and the Bellman equations are imposed as constraints for every stage game. Due to space limit, this optimization problem is provided in the appendix.

Similarly to normal form games, we can characterize the feasibility of the above Game Modification problem in Markov games, done in the corollary below. Let $\mathcal{I}_h(s) = \text{supp}(\mathbf{p}_h(s))$ and $\mathcal{J}_h(s) = \text{supp}(\mathbf{q}_h(s))$.

Corollary 6.1 (Feasibility of Markov Game Modification)**.** *The Game Modification problem in Definition 6.1 for Markov games is feasible if and only if* $|\mathcal{I}_h(s)| = |\mathcal{J}_h(s)|$ *for every* $h \in [H], s \in \mathcal{S}$*, and* $(-Hb, Hb) \bigcap [\underline{v}, \overline{v}] \neq \emptyset$*.*

The conditions above are sufficient and necessary for feasibility. In particular, sufficiency is proved by explicitly constructing a feasible Markov game, recursively using the Extended Rock-Paper-Scissors game.

To develop an efficient algorithm, we follow a similar strategy as in normal form games: we ignore the INV (invertibility) condition and retain only the linear constraints for the Markov game modification problem, and add small margins ι, λ to the SIISOW and reward bound constraints so that random perturbation can be added later. Doing so leads to a linearly constrained optimization problem, given in (6.15), which generalizes the program (6.10) for normal form games.

Remark 6.2. *If there is no value range constraint and the cost function $\ell(R, R^\circ)$ is decomposable across the states and periods (e.g., L^1 cost), then the program (6.15) can be broken into $H|\mathcal{S}|$ smaller optimization problems, one for each stage game, that can be solved sequentially by backward induction.*

$$
\begin{aligned}
&\min_{R,\nu,\mathbb{Q}} \ \ell\left(R, R^\circ\right) && (6.15)\\
&\text{s.t. } \left[\mathbb{Q}_h\left(s\right)\right]_{\mathcal{J}_h(s)\bullet} \mathbf{q}_h\left(s\right) = \nu_h\left(s\right) 1_{|\mathcal{J}_h(s)|} \\
&\qquad \forall\, h \in [H], s \in \mathcal{S} && [\text{row SII}]\\
&\quad \mathbf{p}_h^\top\left(s\right) \left[\mathbb{Q}_h\left(s\right)\right]_{\bullet \mathcal{J}_h(s)} = \nu_h\left(s\right) 1_{|\mathcal{J}_h(s)|}^\top \\
&\qquad \forall\, h \in [H], s \in \mathcal{S} && [\text{column SII}]\\
&\quad \left[\mathbb{Q}_h\left(s\right)\right]_{\mathcal{A}_1 \setminus \mathcal{J}_h(s)\bullet} \mathbf{q}_h\left(s\right) \leqslant \left(\nu_h\left(s\right) - \iota\right) 1_{|\mathcal{A}_1 \setminus \mathcal{J}_h(s)|} \\
&\qquad \forall\, h \in [H], s \in \mathcal{S} && [\text{row SOW}]\\
&\quad \mathbf{p}_h^\top\left(s\right) \left[\mathbb{Q}_h\left(s\right)\right]_{\bullet \mathcal{A}_2 \setminus \mathcal{J}_h(s)} \geqslant \left(\nu_h\left(s\right) + \iota\right) 1_{|\mathcal{A}_2 \setminus \mathcal{J}_h(s)|}^\top \\
&\qquad \forall\, h \in [H], s \in \mathcal{S} && [\text{column SOW}]\\
&\quad \mathbb{Q}_h\left(s\right) = R_h\left(s\right) + \sum_{s' \in \mathcal{S}} P_h\left(s'|s\right) \nu_{h+1}\left(s'\right) \\
&\qquad \forall\, h \in [H-1], s \in \mathcal{S} && [\text{Bellman}]\\
&\quad \mathbb{Q}_H\left(s\right) = R_H\left(s\right), \forall\, s \in \mathcal{S} \\
&\quad \underline{\nu} \leqslant \sum_{s \in \mathcal{S}} P_0\left(s\right) \nu_1\left(s\right) \leqslant \overline{\nu} && [\text{value range}]\\
&\quad -b + \lambda \leqslant \left[R_h\left(s\right)\right]_{ij} \leqslant b - \lambda \\
&\qquad \forall\ (i,j) \in \mathcal{A}, h \in [H], s \in \mathcal{S} && [\text{reward bound}]
\end{aligned}
$$

We present our algorithm, Relax And Perturb for Markov Games (RAP-MG), in Algorithm 5. The algorithm adds random perturbation to the reward matrix of every stage game. Consequently, the feasibility and optimality results for Algorithm 5 are similar to those for the RAP algorithm

for normal form games in Proposition 6.1, though the proofs are complicated by the dependency across the stage games. These results and proofs are provided in the appendix.

Algorithm 5 Relax And Perturb for Markov Games (RAP-MG)

Input: original game (R°, P), cost function ℓ, target policy $(\mathbf{p}, \mathbf{q})$, target value range $[\underline{v}, \overline{v}]$, reward bound $b \in \mathbb{R}^+ \cup \{\infty\}$.
Parameters: margins $\iota \in \mathbb{R}^+$ and $\lambda \in \mathbb{R}^+$.
Output: modified game (R, P).

1: Solve the problem (6.15). Call the solution R'.
2: **for** $h \in [H]$, $s \in \mathcal{S}$ **do**
3: Sample $\varepsilon \sim \text{uniform}[-\lambda, \lambda]$
4: Perturb the reward matrix in stage (h, s):
5: $R_h(s) = R'_h(s) + \varepsilon R^{\text{eRPS}(\mathbf{p}_h(s), \mathbf{q}_h(s))}$.
6: Return (R, P).

6.5 Experiments

We numerically evaluate our game design algorithms. Additional experiments are given in the appendix.

Two Finger Morra

The simplified two-finger morra game (Good, 1965) is given by the payoff matrix on the left below, which has a unique NE $(\mathbf{p}, \mathbf{q}) = (\frac{7}{12}, \frac{5}{12})$ and value $-\frac{1}{12}$. We aim to modify the game to keep the same unique NE minimally but make the game fair with a value of 0. The redesigned game is given below.

$$\text{Original: } \begin{pmatrix} 2 & -3 \\ -3 & 4 \end{pmatrix} \quad \text{Modified: } \begin{pmatrix} 2.04 & -2.86 \\ -2.86 & 4 \end{pmatrix}$$

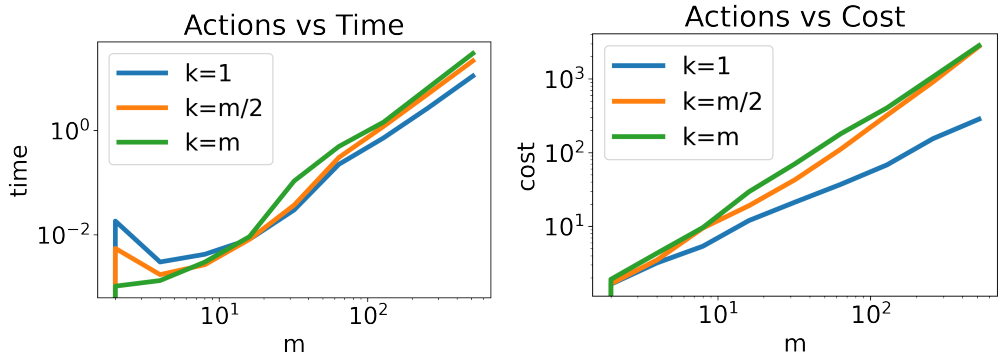


Figure 6.1: Scale Benchmark for Number of Actions

Rock-Paper-Scissors-Fire-Water

Given on the left below is a generalization of the RPS game to five actions (Tagiew, 2009). The unique NE is $\mathbf{p} = \mathbf{q} = (\frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{3}, \frac{1}{3})$ and has value 0. We desire the NE to be simpler for humans, so we redesign the game to have a uniformly mixed NE $\mathbf{p} = \mathbf{q} = (\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$. The resultant game is given below.

$$\begin{array}{cc} \text{Original} & \text{Modified} \\ \begin{pmatrix} 0 & -1 & 1 & -1 & 1 \\ 1 & 0 & -1 & -1 & 1 \\ -1 & 1 & 0 & -1 & 1 \\ 1 & 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 1 & 0 \end{pmatrix} & \begin{pmatrix} 0 & -1 & 1 & -1 & 1 \\ 1 & 0 & -1 & -1 & 1 \\ -1 & 1 & 0 & -1 & 1 \\ 1 & 1 & 1 & 0 & -3 \\ -1 & -1 & -1 & 3 & 0 \end{pmatrix} \end{array}$$

Note that an alternative 5-action game Rock-Paper-Scissors-Spock-Lizard also has the desired NE. However, our modification has a lower modification cost 4, compared to the cost 8 for using the alternative game.

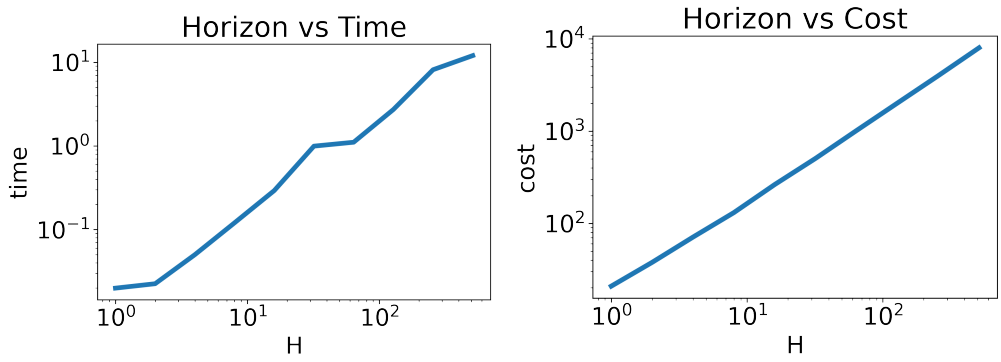


Figure 6.2: Scale Benchmark for Number of Periods

Markov Game Scale Benchmark

We run Algorithm 4 and Algorithm 5 on several games to illustrate the efficacy of our techniques. We know our algorithm succeeds by checking that $(\mathbf{p}, \mathbf{q})$ satisfies the SIISOW and INV (invertibility) conditions for $R^\dagger$. By Theorem 6.1, satisfying these conditions implies $(\mathbf{p}, \mathbf{q})$ is the unique NE for $R^\dagger$.

We first show how our methods scale with the number of actions. For each $m \in \{2, 4, 8, \ldots, 512\}$ we generate $N = 5$ random matrices $R^\circ \sim \text{uniform}[-1, 1]^{m \times m}$. For each matrix, we also generate 3 random $(\mathbf{p}, \mathbf{q}) \sim \text{Dirichlet}(1, \ldots, 1)$ with support size (i) $k = 1$, (ii) $k = m/2$, and (iii) $k = m$ (full support). We run Algorithm 4 on each instance and report the worst running time (in seconds) and the worst cost encountered for each m in Figures 6.1. We see that the solving time grows linearly in the log and so the runtime is polynomial in the actions. Using the Gurobi LP solver, even on a laptop computer the algorithm handles millions of variables (512^2) in roughly 10 seconds. The L^1 costs also appear to grow linearly though with different slopes. We observed that both SIISOW and INV always holds after perturbation.

Next, we show how our methods scale with the horizon. We consider Markov games with $S = 10$, $A = 2$, random transitions and random

reward matrices. Formally, for each $H \in \{1, 2, 4, \ldots, 512\}$, we generate $N = 5$ random Markov games and corresponding target NE pairs with full support. For any fixed H, we generate $R_h(s) \in \text{uniform}[-1, 1]^{2\times 2}$ for each h and s, and choose $P_h(s, a) \sim \text{Dirichlet}(1, \ldots 1)$ for each (h, s, a). We run Algorithm 4 on each instance and report the worst running time and cost encountered for each H in Figures 6.2. We observe the solutions are correct and again the algorithm is efficient.

6.6 Conclusion

Our work points to several directions of future research: (i) It is of interest to study Markov game modification problems where the transition probabilities can also be changed, and to generalize to general-sum, multi-agent games and other equilibrium concepts. (ii) In many games, the rewards are constrained to take integer (e.g., $-1, 0, 1$) or other discrete values. The feasibility and tractability of such constrained game modification problems require further investigation. (iii) Extending our results to data poisoning problems, where the players learn the true game from observational data, lead to interesting theoretical and algorithmic questions.

7 FUTURE WORK

Infinite State and Action Spaces

The approach we used in our work is mostly only applicable to normal-form games or tabular Markov games with finite state and action spaces. One immediate extension is to study adversarial attacks when the number of states or actions is infinite. Installing a dominant strategy equilibrium in the planning setting becomes the following problem,

$$\begin{aligned} \min_{R} \quad & C\left(R, R^{o}\right) \\ \text{s.t.} \quad & R\left(s\right)\left(a_{i}^{\dagger}\left(s\right), a_{-i}\right) > R\left(s\right)\left(a_{i}, a_{-i}\right), \forall\, a_{i} \neq a_{i}^{\dagger}\left(s\right), \forall\, a_{-i}, \forall\, s, i, \end{aligned}$$

which has an infinite number of constraints, and the resulting optimization problem could not be solved efficiently. Installing other equilibrium concepts in the other setting is similarly intractable. One possibility is to make smoothness assumptions and discretize the state and action spaces. Another possibility is to impose further linear or quadratic assumptions on the R as a function of the actions and states, for example,

$$R\left(s\right)\left(a\right) = s^{T} A s + a_{i}^{T} B a_{i} - a_{-i}^{T} C a_{-i},$$

whose Nash equilibria can be characterized by linear functions of s and a, and can be used as constraints in our attacker's problem. More generally, we can paramterize R by θ and a feature function $\varphi\left(s, a\right)$,

$$R_{\theta}\left(s\right)\left(a\right) = \theta^{T} \varphi\left(s, a\right),$$

or more flexibly implement $R_{\theta}\left(s\right)\left(a\right)$ as a non-linear neural network with weights θ. Under convexity assumptions of $\varphi\left(s, a\right)$, we could write the

derivative conditions as the constraints in our attacker's problem,

$$\begin{aligned} &\min_{R} C\left(R, R^{o}\right) \\ &\text{s.t.} \theta^{\top} \frac{\partial}{\partial a_i} \varphi\left(s, a_i = a_i^{\dagger}\left(s\right), a_{-i}\right) = 0, \forall\, i. \end{aligned}$$

Model-Free Victim Algorithms

We also did not directly address the case when the victims do not use model-based algorithms, and we always have the attacker simulate model-based victims who estimate the normal-form or Markov game reward and transition matrices first, then solve for the equilibria based on the estimated game. In the offline setting, we have been using the following form of the attacker's problem,

$$\begin{aligned} &\min_{r} C\left(r, r^{o}\right) \\ &\text{s.t.} \hat{R}\left(r\right) \text{ has a unique equilibrium } \pi^{\dagger}, \end{aligned}$$

where $\hat{R}\left(r\right)$ is the game rewards estimated from r, and we further assume that the estimation function $\hat{R}$ is linear in r. However, some algorithms are model-free, for example, the victims can estimate the value functions or the equilibrium policies directly from the reward data, without estimating the game. In those cases, our approaches may not directly apply. Given that many of these model-free algorithms have high-probability guarantees to find the equilibria of the underlying game that generates the data, it could be possible that we can show our attacks would still be successful if the victims use these algorithms. Alternatively, we could formulate the attack problem without assuming the form of $\hat{R}$, which likely would require more

knowledge of the victim's algorithm,

$$\begin{aligned} &\min_{r} C\left(r, r^{o}\right) \\ &\text{s.t.} \pi^{\star}\left(r\right) = \left\{\pi^{\dagger}\right\}, \end{aligned}$$

where $\pi^{\star}$ is the victims' algorithm to compute the equilibrium based on a dataset r.

Victims with Private Types

We are also interested in solving the adversarial attack problem in which the victims have private information for private types. For example, if the attacker cannot observe the state information in the offline training data or during online training, but the victims can access the information, then the states can be viewed as private types of the victims. The resulting problem under information asymmetry is closely related to dynamic mechanism design, where in addition to incentivizing the victims to use a specific action over others, the attacker must also modify the environment or data in a way that separates the victims with different types. In the private state example, the attacker could compute a modification function ΔR that is independent of the state and solves the following optimization,

$$\begin{aligned} &\min_{\Delta R} \|\Delta R\| \\ &\text{s.t.} \hat{R}\left(s\right)\left(a_i^{\dagger}\left(s\right), a_{-i}\right) + \Delta R\left(a_i^{\dagger}\left(s\right), a_{-i}\right) > \hat{R}\left(s\right)\left(a_i, a_{-i}\right) + \Delta R\left(a_i, a_{-i}\right), \\ &\qquad \forall\, a_i \neq a_i^{\dagger}\left(s\right), \forall\, a_i, \forall\, s, i, \end{aligned}$$

where $\Delta R\left(a, a_{-i}\right)$ specifies the additional transfer (possibly negative) made to the victim i, when the action profile $\left(a, a_{-i}\right)$ is used, that is the same for every state s. This problem is not always feasible, but under some monotonicity conditions of $\hat{R}$, we might be able to apply techniques from mechanism design and solve the problem.

Defense Against Adversarial Attacks

In the longer term, I would investigate the defense problem in the multi-agent reinforcement learning setting. We have pointed out vulnerabilities of these learning algorithms if they were given modified environments and datasets, so it is natural to study possible ways to defend against data poisoning, for example, using more robust learning algorithms that can detect or correct poisoned data. There can be a few directions to approach the defense problem:

- Suppose the victim can inspect a fixed number of entries in the reward matrix or training data at a cost, then the victim might be able to strategically select the number of entries to randomly verify, or strategically specify which entries to verify and confirm the policy is indeed an equilibrium of the original game. The attack fails if the victim inspects an entry that is changed by an amount larger than some threshold. Given the victim's ability to inspect data, the attacker might choose to attack differently to minimize the failure probability, and we can model the interaction as a sequential or Stackelberg game.

- If the victims cannot verify entries, but they have information about the attacker's payoffs from the policies used by the victims, then the victims can select robust policies that are the worst-case best responses based on the attacker's payoff matrices. Even without precise information about the attacker's payoffs, based on the poisoned environment or data, the victims might be able to infer the attacker's payoff structure and react accordingly.

A ONLINE REWARD POISONING FOR BANDIT GAMES TO INSTALL A DOMINANT STRATEGY EQUILIBRIUM

Appendix

Lemma 3.4. *The redesigned game* (3.2) *satisfies:*

1. $\forall i, a, \ell_i(a) \in \tilde{\mathcal{L}}$, *thus* ℓ *is valid.*
2. *For every player* i, *the target action* $a_i^\dagger$ *strictly dominates any other action by* $(1-\frac{1}{M})\rho$, *i.e.,* $\ell_i(a_i, a_{-i}) = \ell_i(a_i^\dagger, a_{-i}) + (1-\frac{1}{M})\rho, \forall i, a_i \neq a_i^\dagger, a_{-i}$.
3. $\ell(a^\dagger) = \ell^o(a^\dagger)$.
4. *If the original loss for the target action profile* $\ell^o(a^\dagger)$ *is zero-sum, then the redesigned game* ℓ *is also zero-sum.*

Proof. The redesigned game (3.2) is given by

$$\forall i, a, \ell_i(a) = \begin{cases} \ell_i^o(a^\dagger) - (1 - \frac{d(a)}{M})\rho & \text{if } a_i = a_i^\dagger, \\ \ell_i^o(a^\dagger) + \frac{d(a)}{M}\rho & \text{if } a_i \neq a_i^\dagger, \end{cases} \tag{A.1}$$

where $d(a) = \sum_{j=1}^{M} \mathbb{1}\left[a_j = a_j^\dagger\right]$.

1. Both branches of $\ell_i(a)$ are lower bounded by L:

$$\ell_i^o(a^\dagger) - (1 - \frac{d(a)}{M})\rho \geqslant \ell_i^o(a^\dagger) - \rho \geqslant L. \tag{A.2}$$

$$\ell_i^o(a^\dagger) + \frac{d(a)}{M}\rho \geqslant \ell_i^o(a^\dagger) \geqslant L. \tag{A.3}$$

   Both branches are upper bounded by U:

$$\ell_i^o(a^\dagger) - (1 - \frac{d(a)}{M})\rho \leqslant \ell_i^o(a^\dagger) \leqslant U. \tag{A.4}$$

$$\ell_i^o(a^\dagger) + \frac{d(a)}{M}\rho \leqslant \ell_i^o(a^\dagger) + \rho \leqslant U. \tag{A.5}$$

Therefore, $\ell_i(a) \in [L, U] = \tilde{\mathcal{L}}$.

2. Fix $i \in [M]$. $\forall a_{-i}$, let $a = (a_i, a_{-i})$ for some $a_i \neq a_i^\dagger$, and $b = (a_i^\dagger, a_{-i})$, then we have $d(b) = d(a) + 1$, thus

$$\begin{aligned} \ell_i(a) - \ell_i(b) &= \ell_i^o(a^\dagger) + \frac{d(a)}{M}\rho - \ell_i^o(a^\dagger) + (1 - \frac{d(b)}{M})\rho \\ &= (1 - \frac{1}{M})\rho. \end{aligned} \tag{A.6}$$

Therefore, for player i the target action $a_i^\dagger$ strictly dominates any other actions by $(1 - \frac{1}{M})\rho$.

3. When $a = a^\dagger$, we have $d(a) = M$, thus by our design, we have $\forall i$,

$$\begin{aligned} \ell_i(a^\dagger) &= \ell_i^o(a^\dagger) - (1 - \frac{d(a)}{M})\rho \\ &= \ell_i^o(a^\dagger) - (1 - \frac{M}{M})\rho = \ell_i^o(a^\dagger). \end{aligned} \tag{A.7}$$

4. Fix a, we sum over all players to obtain

$$\begin{aligned} \sum_{i=1}^{M} \ell_i(a) = &\sum_{i:a_i = a_i^\dagger} \left(\ell_i^o(a^\dagger) - (1 - \frac{d(a)}{M})\rho \right) + \\ &\sum_{i:a_i \neq a_i^\dagger} \left(\ell_i^o(a^\dagger) + \frac{d(a)}{M}\rho \right) \\ &= \sum_i \ell_i^o(a^\dagger) - d(a)(1 - \frac{d(a)}{M})\rho \\ &+ (M - d(a))\frac{d(a)}{M}\rho \\ &= \sum_{i=1}^{M} \ell_i^o(a^\dagger = 0. \end{aligned} \tag{A.8}$$

□

Theorem 3.5. *Using Algorithm 1, the designer can achieve* $\mathbb{E}N^T(a^\dagger) = T - O(MT^\alpha)$ *while incurring expected cumulative design cost* $\mathbb{E}C^T = O(1^\top_{|\mathcal{A}_1|\times 1} M^{1+\frac{1}{p}} T^\alpha)$.

Proof. Since the designer perturbs $\ell^o(\cdot)$ to $\ell(\cdot)$, the players are equivalently running no-regret algorithms under loss function ℓ. Note that according to Lemma 3.4 property 2, $a_i^\dagger$ is the optimal action for player i, and taking a non-target action results in $(1-\frac{1}{M})\rho$ regret regardless of a_{-i}, thus the expected regret of player i is

$$\begin{aligned} \mathbb{E}R_i^T &= \mathbb{E}\sum_{t=1}^{T} \mathbb{1}\left[a_i^t \neq a_i^\dagger\right](1-\frac{1}{M})\rho \\ &= (1-\frac{1}{M})\rho\left(T - \mathbb{E}N_i^T(a_i^\dagger)\right) \end{aligned} \tag{A.9}$$

Rearranging, we have

$$\forall i, \mathbb{E}N_i^T(a_i^\dagger) = T - \frac{M}{(M-1)\rho}\mathbb{E}R_i^T \tag{A.10}$$

Applying a union bound over M players,

$$\begin{aligned}
T - \mathbb{E}N^T(a^\dagger) &= \mathbb{E}\sum_{t=1}^{T} \mathbb{1}\left[a^t \neq a^\dagger\right] \\
&= \mathbb{E}\sum_{t=1}^{T} \mathbb{1}\left[a_j^t \neq a_j^\dagger \text{ for some } j\right] \\
&\leqslant \mathbb{E}\sum_{t=1}^{T}\sum_{j=1}^{M} \mathbb{1}\left[a_j^t \neq a_j^\dagger\right] \\
&= \sum_{j=1}^{M} \mathbb{E}\sum_{t=1}^{T} \mathbb{1}\left[a_j^t \neq a_j^\dagger\right] \\
&= \sum_{j=1}^{M} \left(T - \mathbb{E}N_j(a_j^\dagger)\right) \\
&= \sum_{j=1}^{M} \frac{M}{(M-1)\rho}\mathbb{E}R_i^T \\
&= O(MT^\alpha).
\end{aligned} \tag{A.11}$$

where the second-to-last equation is due to the no-regret assumption of the learner. Therefore, we have $\mathbb{E}N^T(a^\dagger) = T - O(MT^\alpha)$.

Next we bound the expected cumulative design cost. Note that by design $\ell^o(a^\dagger) = \ell(a^\dagger)$, thus when $a^t = a^\dagger$ by our assumption on the cost function we have $C(\ell^o, \ell, a^t) = 0$. On the other hand, when $a^t \neq a^\dagger$ by Lipschitz condition on the cost function we have $C(\ell^o, \ell, a^t) \leqslant 1_{|\mathcal{A}_1|\times 1}^\top M^{\frac{1}{p}}(U-L)$.

Therefore, the expected cumulative design cost is

$$\begin{aligned}
\mathbb{E}C^T &= \mathbb{E}\sum_{t=1}^{T} C(\ell^o, \ell, a^t) \\
&\leqslant 1_{|\mathcal{A}_1|\times 1}^\top M^{\frac{1}{p}}(U-L)\mathbb{E}\sum_{t=1}^{T} \mathbb{1}\left[a^t \neq a^\dagger\right] \\
&= 1_{|\mathcal{A}_1|\times 1}^\top M^{\frac{1}{p}}(U-L)\left(T - \mathbb{E}N^T(a^\dagger)\right) \\
&= 1_{|\mathcal{A}_1|\times 1}^\top M^{\frac{1}{p}}(U-L)\sum_{j=1}^{M}\frac{M}{(M-1)\rho}\mathbb{E}R_i^T \\
&= O(1_{|\mathcal{A}_1|\times 1}^\top M^{1+\frac{1}{p}}T^\alpha),
\end{aligned} \tag{A.12}$$

where the last equality used (A.11). □

Corollary 3.7. *Assume* $M = 2$ *and* ℓ^o *is zero-sum. Then with the redesigned game* (3.2)*, the expected averaged policy* $\mathbb{E}\bar{\pi}_i^T = \mathbb{E}\frac{1}{T}\sum_t \pi_i^t$ *converges to a point mass on* $a_i^\dagger$.

Proof. The new game ℓ is also a two-player zero-sum game. The players applying no-regret algorithm will have their average actions $\mathbb{E}\bar{\pi}^T$ converging to an approximate Nash equilibrium. We use $\pi_i^t(a)$ to denote the probability of player i choosing action a at round t. Next we compute $\mathbb{E}\bar{\pi}_i^T(a^\dagger)$. Note that this expectation is with respect to all the randomness during game playing, including the selected actions $a^{1:T}$ and policies $\pi^{1:T}$. For any t, when we condition on π^t, we have $\mathbb{E}\mathbb{1}\left[a_i^t = a\right] \mid \pi^t = \pi_i^t(a)$. Therefore,

we have $\forall i$

$$\begin{aligned}
\mathbb{E}\bar{\pi}_i^T(a_i^\dagger) &= \frac{1}{T}\mathbb{E}\sum_{t=1}^{T}\pi_i^t(a_i^\dagger) \\
&= \frac{1}{T}\mathbb{E}\pi^{1:T}\sum_{t=1}^{T}\mathbb{E}a^t\mathbb{1}\left[a_i^t = a_i^\dagger\right] \mid \pi^t \\
&= \frac{1}{T}\mathbb{E}\pi^{1:T}\mathbb{E}a^{1:T}\sum_{t=1}^{T}\mathbb{1}\left[a_i^t = a_i^\dagger\right] \mid \pi^{1:T} \\
&= \frac{1}{T}\mathbb{E}\pi^{1:T}\mathbb{E}a^{1:T}N_i^T(a_i^\dagger) \mid \pi^{1:T} \\
&= \frac{1}{T}\mathbb{E}N_i^T(a_i^\dagger) = \frac{T - O(T^\alpha)}{T} \to 1.
\end{aligned} \tag{A.13}$$

Therefore, asymptotically the players believe that $a_i^\dagger, i \in [M]$ form a Nash equilibrium. □

Lemma 3.8. *The redesigned game* (3.3) *satisfies:*

1. $\forall i, a, \ell_i^t(a) \in \tilde{\mathcal{L}}$*, thus the loss function is valid.*
2. *For every player* i*, the target action* $a_i^\dagger$ *strictly dominates any other action by* $(1 - \frac{1}{M})\rho w_t$*, i.e.,* $\ell_i^t(a_i, a_{-i}) = \ell_i^t(a_i^\dagger, a_{-i}) + (1 - \frac{1}{M})\rho w_t, \forall i, t, a_i \neq a_i^\dagger, a_{-i}$.
3. $\forall t, C(\ell^o, \ell^t, a^\dagger) \leqslant 1_{|\mathcal{A}_1|\times 1}^\top (U - L)M^{\frac{1}{p}} w_t$
4. *If the original loss for the target action profile* $\ell^o(a^\dagger)$ *and the vector* v *are both zero-sum, then* $\forall t, \ell^t$ *is zero-sum.*

Proof. The redesigned game (3.3) is given by

$$\ell^t = w_t \underline{\ell} + (1 - w_t)\bar{\ell} \tag{A.14}$$

where

$$w_t = t^{\alpha+\epsilon-1} \tag{A.15}$$

1. Note that $\underline{\ell}$ is valid, as we have proved in Lemma 3.4 property 1, thus $\underline{\ell} \in [L, U]$. Also note that $\overline{\ell} \in [L, U]$. Therefore, $\ell^t = w_t\underline{\ell} + (1-w_t)\overline{\ell} \in [L, U]$.

2. $\forall i$ and $\forall a_{-i}$, let $a = (a_i, a_{-i})$ for some $a_i \neq a_i^\dagger$, and let $b = (a_i^\dagger, a_{-i})$, then according to Lemma 3.4 property 2, we have

$$\underline{\ell}(a) - \underline{\ell}(b) = (1 - \frac{1}{M})\rho. \tag{A.16}$$

   Therefore, we have $\ell^t(a) - \ell^t(b) =$

$$\begin{aligned} &(1-w_t)\overline{\ell}(a) + w_t\underline{\ell}(a) - (1-w_t)\overline{\ell}(b) + w_t\underline{\ell}(b) \\ &= (1-w_t)\ell^o(a^\dagger) + w_t\underline{\ell}(a) - (1-w_t)\ell^o(a^\dagger) + w_t\underline{\ell}(b) \\ &= w_t\left(\underline{\ell}(a) - \underline{\ell}(b)\right) = (1 - \frac{1}{M})\rho w_t. \end{aligned} \tag{A.17}$$

3. Note that we have

$$\begin{aligned} \ell^o(a^\dagger) - \ell^t(a^\dagger) &= \ell^o(a^\dagger) - \left(w_t\underline{\ell}(a^\dagger) + (1-w_t)\ell^o(a^\dagger)\right) \\ &= w_t\left(\ell^o(a^\dagger) - \underline{\ell}(a^\dagger)\right). \end{aligned} \tag{A.18}$$

   Therefore, we have

$$\begin{aligned} C(\ell^o, \ell^t, a^\dagger) &\leqslant 1_{|\mathcal{A}_1|\times 1}^\top \|\ell^o(a^\dagger) - \ell^t(a^\dagger)\|_p \\ &= 1_{|\mathcal{A}_1|\times 1}^\top w_t \|\ell^o(a^\dagger) - \underline{\ell}(a^\dagger)\|_p \\ &\leqslant 1_{|\mathcal{A}_1|\times 1}^\top (U-L)M^{\frac{1}{p}} w_t. \end{aligned} \tag{A.19}$$

4. If the loss vector v is zero-sum, then by Lemma 3.4 property 4 $\underline{\ell}$ is a

zero-sum game. If $\ell^o(a^\dagger)$ is also zero-sum, then we have

$$\begin{aligned}\sum_{i=1}^{M} \ell_i^t(a) &= \sum_{i=1}^{M} \left(w_t \underline{\ell}_i(a) + (1-w_t)\ell_i^o(a^\dagger)\right) \\ &= w_t \sum_{i=1}^{N} \underline{\ell}_i(a) + (1-w_t)\sum_{i=1}^{M} \ell_i^o(a^\dagger) \\ &= 0.\end{aligned} \tag{A.20}$$

□

Theorem 3.9. *Using Algorithm 2, the designer can achieve* $\mathbb{E}N^T(a^\dagger) = T - O(MT^{1-\epsilon})$ *while incurring expected cumulative design cost* $\mathbb{E}C^T = O(M^{1+\frac{1}{p}}T^{1-\epsilon} + M^{\frac{1}{p}}T^{\alpha+\epsilon})$.

Proof. Under game redesign, the players are equivalently running no-regret algorithms over the game sequence $\ell^1, \ldots, \ell^T$ instead of $\ell^o(\cdot)$. By Lemma 3.8 property 2, $a_i^\dagger$ is always the optimal action for player i, and taking a non-target action results in $(1 - 1/M)\rho w_t$ regret regardless of a_{-i}, thus the expected regret of player i is

$$\begin{aligned}\mathbb{E}R_i^T &= \mathbb{E}\sum_{t=1}^{T} \mathbb{1}\left[a_i^t \neq a_i^\dagger\right](1 - \frac{1}{M})\rho w_t \\ &= (1 - \frac{1}{M})\rho \mathbb{E}\sum_{t=1}^{T} \mathbb{1}\left[a_i^t \neq a_i^\dagger\right] w_t.\end{aligned} \tag{A.21}$$

Now note that $w_t = t^{\alpha+\epsilon-1}$ is monotonically decreasing as t grows, thus we have

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{1}\left[a_i^t \neq a_i^\dagger\right] w_t &\geqslant \sum_{t=N_i(a_i^\dagger)+1}^{T} t^{\alpha+\epsilon-1} \\ &= \sum_{t=1}^{T} t^{\alpha+\epsilon-1} - \sum_{t=1}^{N_i(a_i^\dagger)} t^{\alpha+\epsilon-1}.\end{aligned} \tag{A.22}$$

Next, by examining the area under curve, we obtain

$$\sum_{t=1}^{T} t^{\alpha+\epsilon-1} \geqslant \int_{1}^{T} t^{\alpha+\epsilon-1} dt = \frac{1}{\alpha+\epsilon} T^{\alpha+\epsilon} - \frac{1}{\alpha+\epsilon}. \tag{A.23}$$

Similarly, we can also derive

$$\sum_{t=1}^{N_i(a_i^\dagger)} t^{\alpha+\epsilon-1} \leqslant \int_{0}^{N_i(a_i^\dagger)} t^{\alpha+\epsilon-1} dt = \frac{1}{\alpha+\epsilon} \left(N_i^T(a_i^\dagger) \right)^{\alpha+\epsilon}. \tag{A.24}$$

Therefore, we have $\sum_{t=1}^{T} \mathbb{1}\left[a_i^t \neq a_i^\dagger\right] w_t \geqslant$

$$\begin{aligned}
&\frac{1}{\alpha+\epsilon} \left(T^{\alpha+\epsilon} - \left(N_i^T(a_i^\dagger) \right)^{\alpha+\epsilon} \right) - \frac{1}{\alpha+\epsilon} \\
&= \frac{1}{\alpha+\epsilon} T^{\alpha+\epsilon} \left(1 - (1 - \frac{T - N_i^T(a_i^\dagger)}{T})^{\alpha+\epsilon} \right) - \frac{1}{\alpha+\epsilon} \\
&\geqslant \frac{1}{\alpha+\epsilon} T^{\alpha+\epsilon} \frac{T - N_i^T(a_i^\dagger)}{T} (\alpha+\epsilon) - \frac{1}{\alpha+\epsilon} \\
&= T^{\alpha+\epsilon} - T^{\alpha+\epsilon-1} N_i^T(a_i^\dagger) - \frac{1}{\alpha+\epsilon}.
\end{aligned} \tag{A.25}$$

The inequality follows from the fact $(1-x)^c \leqslant 1 - cx$ for $x, c \in (0,1)$. Plug back in (A.21) we have

$$\begin{aligned}
&\mathbb{E} R_i^T = (1 - \frac{1}{M}) \rho \mathbb{E} \sum_{t=1}^{T} \mathbb{1}\left[a_i^t \neq a_i^\dagger\right] w_t \\
&\geqslant (1 - \frac{1}{M}) \rho \mathbb{E} \left(T^{\alpha+\epsilon} - T^{\alpha+\epsilon-1} N_i^T(a_i^\dagger) - \frac{1}{\alpha+\epsilon} \right) \\
&= (1 - \frac{1}{M}) \rho \left(T^{\alpha+\epsilon} - T^{\alpha+\epsilon-1} \mathbb{E} N_i^T(a_i^\dagger) - \frac{1}{\alpha+\epsilon} \right)
\end{aligned} \tag{A.26}$$

As a result, we have $\forall i, \mathbb{E}N_i^T(a_i^\dagger) \geqslant$

$$\begin{aligned}
&T - \frac{M}{(M-1)\rho}\mathbb{E}R_i^T T^{1-\alpha-\epsilon} - \frac{1}{\alpha+\epsilon}T^{1-\alpha-\epsilon} \\
&= T - \frac{M}{(M-1)\rho}O(T^\alpha)T^{1-\alpha-\epsilon} - \frac{1}{\alpha+\epsilon}T^{1-\alpha-\epsilon} \\
&= T - O(T^{1-\epsilon}) - O(T^{1-\alpha-\epsilon}) \\
&= T - O(T^{1-\epsilon}).
\end{aligned} \tag{A.27}$$

By a union bound similar to (A.11), we have $\mathbb{E}N^T(a^\dagger) = T - O(MT^{1-\epsilon})$.

We now analyze the cumulative design cost. Note that by Lemma 3.8 property 3, when $a^t = a^\dagger$, $C(\ell^o, \ell^t, a^t) \leqslant 1_{|\mathcal{A}_1|\times 1}^\top (U - L)M^{\frac{1}{p}} w_t$. On the other hand, when $a^t \neq a^\dagger$, we have

$$C(\ell^o, \ell^t, a^t) \leqslant 1_{|\mathcal{A}_1|\times 1}^\top \|\ell^o(a^t) - \ell^t(a^t)\|_p \leqslant 1_{|\mathcal{A}_1|\times 1}^\top (U - L)M^{\frac{1}{p}}. \tag{A.28}$$

Therefore, the expected cumulative design cost is

$$\begin{aligned}
\mathbb{E}C^T \leqslant & 1_{|\mathcal{A}_1|\times 1}^\top (U - L)M^{\frac{1}{p}}\mathbb{E}\sum_{t=1}^T \mathbb{1}\left[a^t \neq a^\dagger\right] \\
&+ 1_{|\mathcal{A}_1|\times 1}^\top (U - L)M^{\frac{1}{p}}\mathbb{E}\sum_{t=1}^T \mathbb{1}\left[a^t = a^\dagger\right] w_t \\
&\leqslant 1_{|\mathcal{A}_1|\times 1}^\top (U - L)M^{\frac{1}{p}}(T - \mathbb{E}N^T(a^\dagger)) \\
&+ 1_{|\mathcal{A}_1|\times 1}^\top (U - L)M^{\frac{1}{p}}\sum_{t=1}^T w_t.
\end{aligned} \tag{A.29}$$

$T - \mathbb{E}N^T(a^\dagger) = O(MT^{1-\epsilon})$ is already proved. Also note that

$$\sum_{t=1}^T w_t = \sum_{t=1}^T t^{\alpha+\epsilon-1} \leqslant \int_{t=0}^T t^{\alpha+\epsilon-1} = \frac{1}{\alpha+\epsilon}T^{\alpha+\epsilon}. \tag{A.30}$$

Therefore, we have

$$\begin{aligned}\mathbb{E}C^T &\leqslant (U-L)1^\top_{|\mathcal{A}_1|\times 1} M^{\frac{1}{p}} O(MT^{1-\epsilon}) + \frac{1^\top_{|\mathcal{A}_1|\times 1}(U-L)}{\alpha+\epsilon} M^{\frac{1}{p}} T^{\alpha+\epsilon} \\ &= O(M^{1+\frac{1}{p}} T^{1-\epsilon} + M^{\frac{1}{p}} T^{\alpha+\epsilon}).\end{aligned} \tag{A.31}$$

□

A.1 Exact Form of the Theoretical Upper Bounds

According to Theorem 3.4 in Bubeck and Cesa-Bianchi (2012), the EXP3.P achieves expected regret bound

$$\mathbb{E}R^T \leqslant 5.15\sqrt{TA_i \log A_i} + \sqrt{\frac{TA_i}{\log A_i}}. \tag{A.32}$$

where $A_i = |\mathcal{A}_i|$ is the size of the action space of player i. Note that, however, Bubeck and Cesa-Bianchi (2012) assumes the loss takes value in $[0, 1]$, while we assume the loss lies in $[L, U]$. Therefore, the regret bound should boost by $U - L$, i.e., we have

$$\forall i, \mathbb{E}R_i^T \leqslant (U-L)\left(5.15\sqrt{TA_i \log A_i} + \sqrt{\frac{TA_i}{\log A_i}}\right). \tag{A.33}$$

Plug the above regret bound into the proofs of Theorem 3.5 and Theorem 3.9, we obtain the following exact form of the theoretical upper bounds.

For the interior design Algorithm 1, we have

$$\begin{aligned} T - \mathbb{E}N^T(a^\dagger) &\leqslant \sum_{j=1}^{M} \frac{M}{(M-1)\rho} \mathbb{E}R_i^T \\ &= \frac{M(U-L)}{(M-1)\rho} \sum_{i=1}^{M} \left(5.15\sqrt{TA_i \log A_i} + \sqrt{\frac{TA_i}{\log A_i}} \right) \end{aligned} \tag{A.34}$$

and

$$\begin{aligned} \mathbb{E}C^T &\leqslant 1_{|\mathcal{A}_1|\times 1}^\top M^{\frac{1}{p}} (U-L) \sum_{j=1}^{M} \frac{M}{(M-1)\rho} \mathbb{E}R_i^T \\ &= \frac{1_{|\mathcal{A}_1|\times 1}^\top M^{1+\frac{1}{p}} (U-L)^2}{(M-1)\rho} \sum_{i=1}^{M} \left(5.15\sqrt{TA_i \log A_i} + \sqrt{\frac{TA_i}{\log A_i}} \right) \end{aligned} \tag{A.35}$$

For the boundary design, we have $T - \mathbb{E}N^T(a^\dagger) \leqslant$

$$\begin{aligned} &\sum_{i=1}^{M} \left(\frac{M}{(M-1)\rho} \mathbb{E}R_i^T T^{1-\alpha-\epsilon} + \frac{1}{\alpha+\epsilon} T^{1-\alpha-\epsilon} \right) = \\ &\left(\frac{M(U-L)}{(M-1)\rho} \sum_{i=1}^{M} \left(5.15\sqrt{TA_i \log A_i} + \sqrt{\frac{TA_i}{\log A_i}} \right) + \frac{M}{\alpha+\epsilon} \right) T^{1-\alpha-\epsilon}. \end{aligned} \tag{A.36}$$

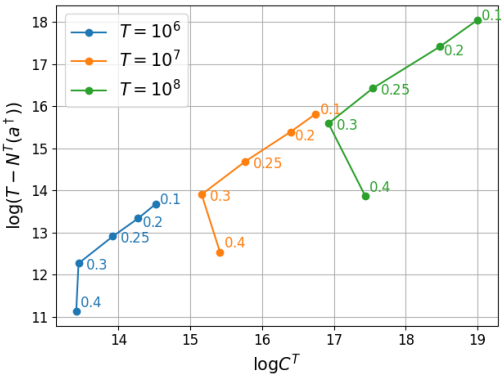


Figure A.1: Number of rounds with $a^t \neq a^\dagger$. The dashed lines are the theoretical upper bound.

and

$$\begin{aligned}\mathbb{E}C^T &\leqslant 1_{|\mathcal{A}_1|\times 1}^\top (U-L)M^{\frac{1}{p}}(T-\mathbb{E}N^T(a^\dagger)) + 1_{|\mathcal{A}_1|\times 1}^\top (U-L)M^{\frac{1}{p}} \sum_{t=1}^{T} w_t \\ &\leqslant 1_{|\mathcal{A}_1|\times 1}^\top (U-L)M^{\frac{1}{p}} \times (A.36) + \frac{1_{|\mathcal{A}_1|\times 1}^\top (U-L)M^{\frac{1}{p}}}{\alpha+\epsilon} T^{\alpha+\epsilon}.\end{aligned} \tag{A.37}$$

A.2 Minimum Cumulative Design Cost

Theorem 3.9 suggests that the minimum cost is achieved at $\epsilon^* = \frac{1-\alpha}{2} = 0.25$, while Figure 3.4b implies that the cost is minimum at some $\epsilon \in (0.3, 0.4)$. We believe the inconsistency is due to not large enough horizon T. We now experiment with slightly larger T for the RPS game with $a^\dagger = (R, P)$. Specifically, we let $T = 10^6, 10^7, 10^8$ and $\epsilon = 0.1, 0.2, 0.25, 0.3, 0.4$. In Figure A.1, we plot $\log(T - N^T(a^\dagger))$ against $\log C^T$ and we marked out the corresponding ϵ values on the curve. Note that for different T, the pattern remains the same – as ϵ grows, $\log(T - \log N^T(a^\dagger))$ decreases monotonically, while $\log C^T$ first reduces and then increases. We also note that as T becomes larger, the ϵ with the minimum cumulative design cost becomes closer to $\epsilon^* = 0.25$. We anticipate that as T grows even larger (e.g., 10^{10}), the cumulative design cost will achieve the minimum at exactly $\epsilon^* = 0.25$.

B OFFLINE REWARD POISONING FOR GENERAL-SUM GAMES TO INSTALL A DOMINANT STRATEGY EQUILIBRIUM

B.1 Compatibility with Pessimistic/Optimistic Offline MARL Algorithms

There is growing literature on offline RL with theoretical guarantees Jin et al. (2021b); Cui and Du (2022b); Zhong et al. (2022). In particular, to address the uncertainty due to the 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 Jin et al. (2021b). Recent work extends the pessimism principle to offline multi-agent RL, focusing on finding the NE in *two-player zero-sum* MG: Cui and Du (2022b) considers tabular setting, while Zhong et al. (2022) 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 continues 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), which is a generalization

of the existing pessimistic algorithms from Jin et al. (2021b); Cui and Du (2022b); Zhong et al. (2022). When specialized to two-player zero-sum MGs, this algorithm class recovers the pessimistic offline algorithms from Cui and Du (2022b); Zhong et al. (2022). We emphasize that our goal is not to provide a 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

$$\left(B^*_{i,h} f\right)(s, a_1) = R^*_{i,h}(s, a_1) + \langle P^*_h(\cdot|s, a_1), f(\cdot)\rangle\,, \tag{B.1}$$

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 $\mathcal{D} = \left\{(s^k_h, a^k_h, r^k_h)\right\}_{h\in[H],k\in[K]}$, the learner constructs the empirical Bellman operator $\widehat{B}_{i,h} : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ by

$$\left(\widehat{B}_{i,h} f\right)(s, a_1) = \widehat{R}_{i,h}(s, a_1) + \left\langle \widehat{P}_h(\cdot|s, a_1), f(\cdot)\right\rangle, \tag{B.2}$$

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.

In the above algorithm, $\Gamma_{i,h}(s, a_1)$ 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 $|\Gamma_{i,h}(s, a_1)| \propto \frac{1}{\sqrt{N_h(s,a_1)+1}}$, where we recall that $N_h(s, a_1)$ is the visit count of the state-action pair (s, a_1). Note that we allow $\Gamma_{i,h}(s, a_1)$ to take an arbitrary sign, with a positive (resp., negative) $\Gamma_{i,h}(s, a_1)$ corresponding to pessimism (resp., optimism).

Recall that in our attack formulation, confidence intervals $CI^R_{i,h}(s, a_1)$

Algorithm 6 Pessimistic-Optimistic Value Iteration (POVI)

Input: Offline dataset $\mathcal{D} = \left\{(s_h^k, a_h^k, r_h^k)\right\}_{h\in[H],k\in[K]}$
  Initialization: $\underline{V}_{i,H+1}(\cdot) \leftarrow 0$ for all i
  for $h = H, H-1, \ldots, 1$ **do**
    Let $\widetilde{Q}_{i,h}(s,a_1) \leftarrow \left(\widehat{B}_{i,h}\underline{V}_{i,h+1}\right)(s,a_1), \quad i \in [n]$ `// empirical Q estimate`
    Let $\underline{Q}_{i,h}(s,a_1) \leftarrow \widetilde{Q}_{i,h}(s,a_1) - \Gamma_{i,h}(s,a_1), \quad i \in [n]$ `// bonus`
    Let $\underline{\boldsymbol{\pi}}_h(\cdot|s) \leftarrow \mathrm{NE}\left(\underline{Q}_{1,h}(s,\cdot), \ldots, \underline{Q}_{n,h}(s,\cdot)\right)$ `// 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 $\underline{\boldsymbol{\pi}} = (\underline{\boldsymbol{\pi}}_h)_{i=1}^H$

and $\mathrm{CI}_h^P(s,a_1)$, with confidence widths $\rho_h^R(s,a_1)$ and $\rho_h^P(s,a_1)$, are constructed for the reward and transition, respectively (cf. Definition 4.2). We impose the following assumption on the relationship between these confidence widths and the bonus $\Gamma_{i,h}(s,a_1)$ used in POVI.

Assumption B.1 (Relationship between CIs)**.** *The above CIs satisfy*

$$|\Gamma_{i,h}(s,a_1)| \leqslant \rho_h^R(s,a_1) + \max_{\substack{u\in\mathbb{R}^{|\mathcal{S}|}:\sum_{s'} u(s')=0 \\ \|u\|_1 \leqslant \rho_h^P(s,a_1)}} \left\langle u, \underline{V}_{i,h+1}\right\rangle$$

for all $(i,s,a_1,h) \in [n]\times\mathcal{S}\times\mathcal{A}\times[H]$.

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 $\pi^\dagger$ as the unique ι-strict MPDSE (hence also the unique NE) for *all* plausible games in the confidence set (see Lemma 4.1). Combined with Theorem B.1, we conclude that our attack will also successfully install $\pi^\dagger$ as the output of any instantiation of POVI.

Theorem B.1 (Compatibility)**.** *Under Assumption B.1, there exist some reward function and transition kernel* $(\mathbf{R}_h, P_h)_{h\in[H]}$ *for which the following hold:*

1. *For all* $(i, s, a_1, h) \in [n] \times \mathcal{S} \times \mathcal{A} \times [H]$*:*

$$R_{i,h}(s, a_1) \in CI^R_{i,h}(s, a_1),$$
$$P_h(\cdot \mid s, a_1) \in CI^P_h(s, a_1).$$

2. *For all* $(i, s, a_1, h) \in [n] \times \mathcal{S} \times \mathcal{A} \times [H]$*:*

$$\underline{Q}_{i,h}(s, a_1) = R_{i,h}(s, a_1) + \left\langle P_h(\cdot \mid s, a_1), \underline{V}_{i,h+1}(\cdot)\right\rangle,$$
$$\underline{V}_{i,h}(s) = \langle \underline{\boldsymbol{\pi}}_h(\cdot|s), \underline{Q}_{i,h}(s, \cdot)\rangle,$$
$$\text{where } \underline{\boldsymbol{\pi}}_h(\cdot|s) = \mathrm{NE}\left(\underline{Q}_{1,h}(s, \cdot), \ldots, \underline{Q}_{n,h}(s, \cdot)\right).$$

   That is, $\underline{\mathbf{Q}}_h$ *and* $\underline{\mathbf{V}}_h$ *are the Q and value functions of the NE of the Markov Game* $G = (\mathcal{S}, \mathcal{A}, \mathbf{R}, P, H)$*.*

Proof of Theorem B.1. Fix an arbitrary tuple $(i, s, a_1, h) \in [n] \times \mathcal{S} \times \mathcal{A} \times [H]$. By Assumption B.1, there exists a $u \in \mathbb{R}$ and a $U \in \mathbb{R}^{|\mathcal{S}|}$ satisfying $|u| \leqslant \rho^R_h(s, a_1)$, $\sum_{s'} U(s') = 0$, $\|U\|_1 \leqslant \rho^P_h(s, a_1)$ and

$$\Gamma_{i,h}(s, a_1) = u + \left\langle U, \underline{V}_{i,h+1}\right\rangle. \tag{B.3}$$

Let

$$R_{i,h}(s, a_1) = \widehat{R}_{i,h}(s, a_1) - u,$$
$$P_h(\cdot \mid s, a_1) = \widehat{P}_h(\cdot \mid s, a_1) - U.$$

By construction it is clear that $R_{i,h}(s, a_1) \in CI^R_{i,h}(s, a_1)$ and $P_h(\cdot \mid s, a_1) \in$

$\mathrm{CI}^{\mathrm{P}}_{h}(s, a_1)$, hence part 1 of the theorem holds. Moreover, we have

$$\begin{aligned}
\underline{Q}_{i,h}(s, a_1) &\overset{\text{(i)}}{=} \left(\widehat{B}_{i,h}\underline{V}_{i,h+1}\right)(s, a_1) - \Gamma_{i,h}(s, a_1) \\
&\overset{\text{(ii)}}{=} \widehat{R}_{i,h}(s, a_1) + \left\langle \widehat{P}_h(\cdot \mid s, a_1), \underline{V}_{i,h+1}(\cdot) \right\rangle \\
&\qquad - u - \left\langle U, \underline{V}_{i,h+1} \right\rangle \\
&\overset{\text{(iii)}}{=} R_{i,h}(s, a_1) + \left\langle P_h(\cdot \mid s, a_1), \underline{V}_{i,h+1}(\cdot) \right\rangle,
\end{aligned}$$

where step (i) follows from Line 4 in Algorithm 6, step (ii) follows from the definition of $\widehat{B}_{i,h}$ in (B.2) and the expression of $\Gamma_{i,h}(s, a_1)$ in (B.3), and step (iii) follows from the above construction of $R_{i,h}(s, a_1)$ and $P_h(\cdot \mid s, a_1)$. This proves the first equation in Part 2 of the theorem. The remaining equations in Part 2 are from the POVI algorithm specification. □

Remark B.1. *Below we discuss when Assumption B.1 holds and how it is related to common choices of the confidence widths $\Gamma_h, \rho^R_h, \rho^P_h$.*

- *A sufficient condition for Assumption B.1 is*

$$|\Gamma_{i,h}(s, a_1)| \leqslant \rho^R_h(s, a_1), \qquad \forall i, s, a_1, h. \tag{B.4}$$

  This condition becomes equivalent to Assumption B.1 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 (B.4) and in turn Assumption B.1 are satisfied for*

the following choices of the bonus term and CI widths:

$$|\Gamma_{i,h}(s, a_1)| = H\sqrt{\frac{\beta}{N_h(s, a_1) + 1}},$$
$$\rho_h^R(s, a_1) = H\sqrt{\frac{\beta}{N_h(s, a_1) + 1}},$$
$$\rho_h^P(s, a_1) = \sqrt{\frac{|\mathcal{S}|\,\beta}{N_h(s, a_1) + 1}}.$$

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, $N := |\mathcal{D}| = \sum_{s,a,h} N_h(s, a_1)$ and δ the desired failure probability. Note that the above choice of $\Gamma_{i,h}(s, a_1)$ is similar to those used in existing work on offline MARL Cui and Du (2022b); Zhong et al. (2022). The above choices of $\rho_h^R(s, a_1)$ and $\rho_h^P(s, a_1)$, given by Hoeffding-type concentration inequalities, are also similar to those typically used in existing model-based RL algorithms.

B.2 Feasibility Proofs

Proof of Proposition 4.1

Proof. When $N_h(s, a_1) = 0$, the learners may assume arbitrary default values for the missing entries, and the attacker has no way of modifying such entries. In particular, the learners may assume values such as b or $-b$, and if the default values for $\hat{R}_{i,h}(s, a_1) = -b$ when $a_1 = \pi_h^\dagger(s)$ or $\hat{R}_{i,h}(s, a_1)$ when $a_1 \neq \pi_h^\dagger(s)$, the attacker will not be able to install $\pi_h^\dagger(s)$ as the dominant-strategy in the stage.

□

Proof of Proposition 4.2

We remark that Proposition 4.2 is a special case of the general Theorem 4.1. We refer the readers to the proof of Theorem 4.1 for details.

Proof of Lemma 4.1

Proof. Given $G \in CI^G$, we have, for every $i \in [n], h \in [H], s \in \mathcal{S}$, and $a_1 \in \mathcal{A}$,

$$\begin{aligned} R_{i,h}(s, a_1) &\in CI_h^{R^\dagger}(s, a_1), \\ P_h(s, a_1) &\in CI_h^P(s, a_1), \end{aligned}$$

where we abuse the notation $R_{i,h}(s, a_1)$ to represent the mean reward after the attack, and we compute the Q values based on $R_{i,h}(s, a_1)$ and the target policy $\pi^\dagger$.

In period H,for every $i \in [n], h \in [H], s \in \mathcal{S}$, and $a_1 \in \mathcal{A}$, we have,

$$Q_{i,H}(s, a_1) = R_{i,H}(s, a_1) \leqslant \max_{R_{i,H} \in CI_{i,H}^{R^\dagger}(s, a_1)} R_{i,H} = \overline{Q}_{i,H}(s, a_1),$$

and

$$Q_{i,H}(s, a_1) = R_{i,H}(s, a_1) \geqslant \min_{R_{i,H} \in CI_{i,H}^{R^\dagger}(s, a_1)} R_{i,H} = \underline{Q}_{i,H}(s, a_1).$$

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

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

Therefore, we have, in period H, and every $s \in \mathcal{S}, \pi_H^\dagger(s)$ is a ι-strict domi-

nant strategy equilibrium.

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

$$Q_{i,h+1}(s, a_1) \in \left[\underline{Q}_{i,h+1}(s, a_1), \overline{Q}_{i,h+1}(s, a_1)\right], \tag{B.5}$$

and $\pi^{\dagger}_{h+1}(s)$ is a ι-strict dominant strategy equilibrium. Then, we have in period h, for every $i \in [n], s \in \mathcal{S}, a_1 \in \mathcal{A}$,

$$\begin{aligned}
& Q_{i,h}(s, a_1) \\
=\ & R_{i,h}(s, a_1) + \sum_{s' \in \mathcal{S}} P_h(s'|s, a_1)\, Q_{i,h+1}\left(s', \pi^{\dagger}_{h+1}(s')\right) \\
\leqslant\ & R_{i,h}(s, a_1) + \sum_{s' \in \mathcal{S}} P_h(s'|s, a_1)\, \overline{Q}_{i,h+1}\left(s', \pi^{\dagger}_{h+1}(s')\right) \\
\leqslant\ & \max_{R_{i,h} \in CI^{R^{\dagger}}_h(s,a_1)} R_{i,h} \\
& \quad + \max_{P_h \in CI^{P}_h(s,a_1)} \sum_{s' \in \mathcal{S}} P_h(s')\, \overline{Q}_{i,h+1}\left(s', \pi^{\dagger}_{h+1}(s')\right) \\
=\ & \overline{Q}_{i,h}(s, a_1),
\end{aligned}$$

and,

$$\begin{aligned}
& Q_{i,h}(s, a_1) \\
=\ & R_{i,h}(s, a_1) + \sum_{s' \in \mathcal{S}} P_h(s'|s, a_1)\, Q_{i,h+1}\left(s', \pi^{\dagger}_{h+1}(s')\right) \\
\geqslant\ & R_{i,h}(s, a_1) + \sum_{s' \in \mathcal{S}} P_h(s'|s, a_1)\, \underline{Q}_{i,h+1}\left(s', \pi^{\dagger}_{h+1}(s')\right) \\
\geqslant\ & \min_{R_{i,h} \in CI^{R^{\dagger}}_h(s,a_1)} R_{i,h} \\
& \quad + \min_{P_h \in CI^{P}_h(s,a_1)} \sum_{s' \in \mathcal{S}} P_h(s')\, \underline{Q}_{i,h+1}\left(s', \pi^{\dagger}_{h+1}(s')\right) \\
=\ & \underline{Q}_{i,h}(s, a_1).
\end{aligned}$$

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

$$\begin{aligned} Q_{i,h}\left(s, \left(\pi^{\dagger}_{i,h}(s), a_{-i}\right)\right) &\geqslant \underline{Q}_{i,h}\left(s, \left(\pi^{\dagger}_{i,h}(s), a_{-i}\right)\right) \\ &\geqslant \overline{Q}_{i,h}(s, (a_i, a_{-i})) + \iota \\ &\geqslant Q_{i,h}(s, (a_i, a_{-i})) + \iota. \end{aligned}$$

Therefore, $\pi^{\dagger}_h(s)$ is the ι-strict dominant strategy equilibrium in period h, state s.

Since $\pi^{\dagger}$ is a Markov policy, it is the ι-strict Markov perfect dominant strategy equilibrium. □

Proof of Theorem 4.1

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

$$R^{\dagger}_{i,h}(s, a_1) = \frac{1}{N_h(s, a_1)} \sum_{k=1}^{K} r^{\dagger,(k)}_{i,h} \mathbf{1}_{\left\{s^{(k)}_h = s, \mathbf{a}^{(k)}_h = a_1\right\}} \tag{B.6}$$
$$\forall\, h, s, i, a_1$$

$$\begin{aligned} \underline{Q}_{i,h}(s, a_1) = &\min_{R_h \in CI^{R^{\dagger}}_h(s,a_1)} R_h \\ &+ \mathbf{1}_{h<H} \min_{P_h \in CI^{P}_h(s,a_1)} \sum_{s' \in \mathcal{S}} P_h(s')\, \underline{Q}_{i,h+1}\left(s', \pi^{\dagger}_{h+1}(s')\right) \end{aligned} \tag{B.7}$$
$$\forall\, h, s, i, a_1,$$

$$\begin{aligned} \overline{Q}_{i,h}(s, a_1) = &\max_{R_h \in CI^{R^{\dagger}}_h(s,a_1)} R_h \\ &+ \mathbf{1}_{h<H} \max_{P_h \in CI^{P}_h(s,a_1)} \sum_{s' \in \mathcal{S}} P_h(s')\, \overline{Q}_{i,h+1}\left(s', \pi^{\dagger}_{h+1}(s')\right), \end{aligned} \tag{B.8}$$
$$\forall\, h, s, i, a_1,$$

$$\underline{Q}_{i,h}\left(s,\left(\pi^{\dagger}_{i,h}(s),a_{-i}\right)\right)\geqslant\overline{Q}_{i,h}(s,(a_i,a_{-i}))+\iota \qquad \forall\, h,s,i,a_{-i},a_i\neq\pi^{\dagger}_{i,h}(s), \tag{B.9}$$

$$r^{\dagger,(k)}_{i,h}\in[-b,b],\forall\, h,k,i. \tag{B.10}$$

Consider the attack defined by,

$$r^{\dagger,(k)}_{i,h}=b\mathbf{1}_{\left\{a^{(k)}_{i,h}=\pi^{\dagger}_{i,h}\left(s^{(k)}\right)\right\}}-b\mathbf{1}_{\left\{a^{(k)}_{i,h}\neq\pi^{\dagger}_{i,h}\left(s^{(k)}\right)\right\}}$$

For each h, k, and i. Given this attack, (B.6) implies,

$$\begin{aligned}R^{\dagger}_{i,h}(s,a_1)&=\frac{1}{N_h(s,a_1)}\sum_{k=1}^{K}r^{\dagger,(k)}_{i,h}\mathbf{1}_{\left\{s^{(k)}_h=s,\mathbf{a}^{(k)}_h=a_1\right\}}\\&=b\mathbf{1}_{\left\{a_i=\pi^{\dagger}_{i,h}(s)\right\}}-b\mathbf{1}_{\left\{a_i\neq\pi^{\dagger}_{i,h}(s)\right\}},\forall\, h,s,i,a_1.\end{aligned}$$

Then (B.7) implies, in period H,

$$\begin{aligned}\underline{Q}_{i,H}\left(s,\pi^{\dagger}_h(s)\right)&=\min_{R_{i,h}\in CI^{R^{\dagger}}_{i,H}\left(s,\pi^{\dagger}_h(s)\right)}R_{i,h}\\&=R^{\dagger}_{i,H}\left(s,\pi^{\dagger}_H(s)\right)-\rho^{(R)}_H\left(s,\pi^{\dagger}_H(s)\right)\\&\geqslant b-\frac{b-\frac{\iota}{2}}{(H+1)/2},\forall\, s,i,\end{aligned} \tag{B.11}$$

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

we have

$$
\begin{aligned}
&\underline{Q}_{i,h}\left(s,\pi_h^\dagger(s)\right)\\
=&\min_{R_{i,h}\in CI_{i,H}^{R^\dagger}\left(s,\pi_h^\dagger(s)\right)} R_{i,h}\\
&+\min_{P_h\in CI_h^P\left(s,\pi_h^\dagger(s)\right)}\sum_{s'\in\mathcal{S}} P_h(s')\,\underline{Q}_{i,h+1}\left(s',\pi_{h+1}^\dagger(s')\right)\\
=&\,R_{i,h}^\dagger\left(s,\pi_h^\dagger(s)\right)-\rho_h^{(R)}\left(s,\pi_h^\dagger(s)\right)\\
&+\min_{P_h\in CI_h^P\left(s,\pi_h^\dagger(s)\right)}\sum_{s'\in\mathcal{S}} P_h(s')\,\underline{Q}_{i,h+1}\left(s',\pi_{h+1}^\dagger(s')\right)\\
\geqslant&\, b-\frac{b-\frac{\iota}{2}}{(H+1)/2}\\
&+\min_{P_h\in CI_h^P\left(s,\pi_h^\dagger(s)\right)}\sum_{s'\in\mathcal{S}} P_h(s')(H-h)\left(b-\frac{b-\frac{\iota}{2}}{(H+1)/2}\right)\\
=&\,(H-h+1)\left(b-\frac{b-\frac{\iota}{2}}{(H+1)/2}\right),\forall\, h,s,i.
\end{aligned}
\tag{B.12}
$$

Similarly, in period H, due to reward bound (B.10),

$$
\begin{aligned}
\overline{Q}_{i,H}\left(s,\pi_H^\dagger(s)\right)&=\max_{R_{i,H}\in CI_{i,H}^{R^\dagger}\left(s,\pi_H^\dagger(s)\right)} R_{i,H}\\
&=\min\left\{b,b+\rho_H^{(R)}\left(s,\pi_H^\dagger(s)\right)\right\}\\
&=b,\forall\, s,i,
\end{aligned}
\tag{B.13}
$$

and for $h<H$, assume $\overline{Q}_{i,h+1}\left(s,\pi_h^\dagger(s)\right)=(H-h)\,b,\forall\ s,i$, using the

reward bound (B.10) again, we have,

$$\begin{aligned}
&\overline{Q}_{i,h}\left(s,\pi_h^\dagger(s)\right)\\
=&\max_{R_{i,h}\in CI_{i,h}^{R^\dagger}\left(s,\pi_H^\dagger(s)\right)} R_{i,h}\\
&+\max_{P_h\in CI_h^P\left(s,\pi_h^\dagger(s)\right)}\sum_{s'\in\mathcal{S}} P_h(s')\,\overline{Q}_{i,h+1}\left(s',\pi_{h+1}^\dagger(s')\right)\\
=&\min\left\{b,b+\rho_h^{(R)}\left(s,\pi_h^\dagger(s)\right)\right\}\\
&+\max_{P_h\in CI_h^P\left(s,\pi_h^\dagger(s)\right)}\sum_{s'\in\mathcal{S}} P_h(s')\,\overline{Q}_{i,h+1}\left(s',\pi_{h+1}^\dagger(s')\right)\\
\geqslant& b+\max_{P_h\in CI_h^P\left(s,\pi_h^\dagger(s)\right)}\sum_{s'\in\mathcal{S}} P_h(s')\,(H-h)\,b\\
=&(H-h+1)\,b.
\end{aligned}$$

On the other hand, (B.8) implies, in period H, $a_i\neq\pi_{i,H}^\dagger(s)$,

$$\begin{aligned}
\overline{Q}_{i,H}(s,a_1)&=\max_{R_{i,H}\in CI_{i,H}^{R^\dagger}(s,a_1)} R_{i,H}\\
&=R_{i,H}^\dagger(s,a_1)+\rho_H^{(R)}(s,a_1)\\
&\leqslant -b+\frac{b-\frac{\iota}{2}}{(H+1)/2},\forall\, s,i,
\end{aligned}\tag{B.14}$$

and for $h < H$,

$$
\begin{aligned}
& \overline{Q}_{i,h}(s, a_1) \\
= & \max_{R_{i,h} \in CI_{i,h}^{R^\dagger}(s,a_1)} R_{i,h} \\
& + \max_{P_h \in CI_h^P(s,a_1)} \sum_{s' \in \mathcal{S}} P_h(s') \overline{Q}_{i,h+1}\left(s', \pi_{h+1}^\dagger(s')\right) \\
= & R_{i,h}^\dagger(s, a_1) + \rho_h^{(R)}(s, a_1) \\
& + \max_{P_h \in CI_h^P(s,a_1)} \sum_{s' \in \mathcal{S}} P_h(s') \overline{Q}_{i,h+1}\left(s', \pi_{h+1}^\dagger(s')\right) \\
\leqslant & -b + \frac{b - \frac{\iota}{2}}{(H+1)/2} \\
& + \max_{P_h \in CI_h^P(s,a_1)} \sum_{s' \in \mathcal{S}} P_h(s')(H-h)b \\
= & (H-h-1)b + \frac{b - \frac{\iota}{2}}{(H+1)/2}, \forall\, h, s, i.
\end{aligned}
\tag{B.15}
$$

Then (B.9) is satisfied since,

$$
\begin{aligned}
&\overline{Q}_{i,h}\left(s,\left(a_i,a_{-i}\right)\right)+\iota\\
&\leqslant\left(H-h-1\right)b+\frac{b-\frac{\iota}{2}}{\left(H+1\right)/2}+\iota\\
&=\left(H-h+1\right)\left(b-\frac{b-\frac{\iota}{2}}{\left(H+1\right)/2}\right)\\
&\quad+\left(H-h+2\right)\frac{b-\frac{\iota}{2}}{\left(H+1\right)/2}-\left(2b-\iota\right)\\
&=\left(H-h+1\right)\left(b-\frac{b-\frac{\iota}{2}}{\left(H+1\right)/2}\right)\\
&\quad+\left(\frac{H-h+2}{H+1}-1\right)\frac{b-\frac{\iota}{2}}{1/2},h\geqslant1\\
&\leqslant\left(H-h+1\right)\left(b-\frac{b-\frac{\iota}{2}}{\left(H+1\right)/2}\right)\\
&\quad+\left(\frac{H+1}{H+1}-1\right)\frac{b-\frac{\iota}{2}}{1/2}\\
&=\left(H-h+1\right)\left(b-\frac{b-\frac{\iota}{2}}{\left(H+1\right)/2}\right)\\
&\leqslant\underline{Q}_{i,h}\left(s,\left(\pi_{i,h}^{\dagger}\left(s\right),a_{-i}\right)\right),\\
&\forall\,h,s,i,a_{-i}\neq\pi_{-i,h}^{\dagger}\left(s\right),a_i\neq\pi_{i,h}^{\dagger}\left(s\right).
\end{aligned}
\tag{B.16}
$$

Finally, (B.10) holds by definition,

$$
\begin{aligned}
r_{i,h}^{\dagger,(k)}&=b\mathbf{1}_{\left\{a_{i,h}^{(k)}=\pi_{i,h}^{\dagger}\left(s^{(k)}\right)\right\}}-b\mathbf{1}_{\left\{a_{i,h}^{(k)}\neq\pi_{i,h}^{\dagger}\left(s^{(k)}\right)\right\}}\\
&\in\left[-b,b\right],\forall\,h,k,i.
\end{aligned}
\tag{B.17}
$$

□

Proof of Corollary 4.1

Proof. Given $\rho_h^{(R)}(s, a_1) = f\left(\frac{1}{N_h(s, a_1)}\right)$ for some strictly increasing function f, the feasibility condition can be written as, for all $h \in [H], s \in \mathcal{S}, a_1 \in \mathcal{A}$,

$$\begin{aligned} & \iota \leqslant 2b - (H+1) f\left(\frac{1}{N_h(s, a_1)}\right). \\ \Rightarrow & f\left(\frac{1}{N_h(s, a_1)}\right) \leqslant \frac{2b - \iota}{H+1} \\ \Rightarrow & \frac{1}{N_h(s, a_1)} \leqslant f^{-1}\left(\frac{2b - \iota}{H+1}\right) \\ \Rightarrow & N_h(s, a_1) \geqslant \left(f^{-1}\left(\frac{2b - \iota}{H+1}\right)\right)^{-1}. \end{aligned}$$

In particular, $\rho_h^{(R)}(s, a_1) = 2b\sqrt{\frac{\log\left(\frac{H|\mathcal{S}||\mathcal{A}|}{\delta}\right)}{\max\{N_h(s, a_1), 1\}}}$ is strictly increasing in $\frac{1}{N_h(s, a_1)}$, we have,

$$\begin{aligned} & 2b\sqrt{\frac{\log\left(\frac{H|\mathcal{S}||\mathcal{A}|}{\delta}\right)}{\max\{N_h(s, a_1), 1\}}} \leqslant \frac{2b - \iota}{H+1} \\ \Rightarrow & N_h(s, a_1) \geqslant \frac{4b^2 (H+1)^2 \log\left(\frac{H|\mathcal{S}||\mathcal{A}|}{\delta}\right)}{(2b - \iota)^2}. \end{aligned}$$

This completes the proof. □

B.3 Linear Program Formulations

Bandit Game Maximum Likelihood Learners

The problem (4.1) can be converted into the following linear program:

$$\begin{aligned}
&\min_{r^{\dagger},t,R^{\dagger}} \sum_{i=1}^{n}\sum_{k=1}^{K} t_i^{(k)} \\
&\text{such that } r_i^{\dagger,(k)} - r_i^{0,(k)} \leqslant t_i^{(k)}, \forall\, k, i \\
&\qquad r_i^{0,(k)} - r_i^{\dagger,(k)} \leqslant -t_i^{(k)}, \forall\, k, i \\
&\qquad R_i^{\dagger}(a_1) = \frac{1}{N(a_1)} \sum_{k=1}^{K} r_i^{\dagger,(k)} \mathbf{1}_{\{\mathbf{a}^{(k)}=a_1\}}, \forall\, a_1, i \\
&\qquad R_i^{\dagger}(a_i, a_{-i}) - R_i^{\dagger}\left(\pi_i^{\dagger}, a_{-i}\right) \leqslant -\iota, \forall\, i, a_{-i}, a_i \neq \pi_i^{\dagger} \\
&\qquad r_i^{\dagger,(k)} \leqslant b, \forall\, k, i \\
&\qquad -r_i^{\dagger,(k)} \leqslant -b, \forall\, k, i.
\end{aligned} \tag{B.18}$$

To linearize the L^1 -norm, we introduce slack variables t, and rewrite the objective $\min_{r^{\dagger}} \left\| r^{\dagger} - r^{0} \right\|_1$ as:

$$\begin{aligned}
&\min_{t,r^{\dagger}} e^{T} t \\
&\text{such that } -t \leqslant r^{\dagger} - r^{0} \leqslant t.
\end{aligned} \tag{B.19}$$

Confidence Bound Learners

Bandit Game Confidence Bound Learners

Proof of Proposition 4.3. The problem (4.2) can be converted into the linear program in (B.20).

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,

$$\max\{-b, R_1 - \rho_1\} \geqslant \min\{b, R_2 + \rho_2\} + \iota,$$

as follows:

$$\begin{aligned}
&\min\{-b, R_1 - \rho_1\} + |R_1 - \rho_1 + b| \\
&\geqslant \max\{b, R_2 + \rho_2\} - |R_2 + \rho_2 - b| + \iota,
\end{aligned}$$

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

$$\begin{aligned}
-b + \overline{m}^- + \overline{m}^+ &\geqslant b - \underline{m}^- - \underline{m}^+ + \iota \\
R_1 - \rho_1 + \overline{m}^- + \overline{m}^+ &\geqslant R_2 + \rho_2 - \underline{m}^- - \underline{m}^+ + \iota \\
-b + \overline{m}^- + \overline{m}^+ &\geqslant R_2 + \rho_2 - \underline{m}^- - \underline{m}^+ + \iota \\
R_1 - \rho_1 + \overline{m}^- + \overline{m}^+ &\geqslant b - \underline{m}^- - \underline{m}^+ + \iota \\
\overline{m}^- &\geqslant -R_1 - \rho_1 - b \\
\overline{m}^+ &\geqslant R_1 - \rho_1 + b \\
\underline{m}^- &\geqslant -R_2 - \rho_2 + b \\
\underline{m}^+ &\geqslant R_2 + \rho_2 - b \\
\overline{m}^-, \overline{m}^+, \underline{m}^-, \underline{m}^+ &\geqslant 0.
\end{aligned} \tag{B.21}$$

We do the same conversion for each $a_1 \in \mathcal{A}$ to obtain the above linear problem.

Markov Game Confidence Bound Learners

Proof of Theorem 4.2. The problem (4.3)– (4.7) can be converted into the linear program in (B.22).

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,

$$\max_{P_h \in CI_h^P(s,a_1)} \sum_{s' \in \mathcal{S}} P_h(s') \overline{Q}_{i,h+1}\left(s', \pi_{h+1}^{\dagger}(s')\right),$$

where,

$$\mathrm{CI}_h^P(s,a_1) = \left\{ P_h \in \Delta(\mathcal{A}) : \left\| P_h - \hat{P}_h(s,a_1) \right\|_1 \leqslant \rho_h^{(P)}(s,a_1) \right\}.$$

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

$$\begin{aligned} \max_{P_h} \quad & P_h \cdot \overline{Q}_{i,h+1}\left(\pi_{h+1}^\dagger\right) \\ \text{such that } & P_h \leqslant \hat{P}_h(s,a_1) + \rho_h^{(P)}(s,a_1) \\ & P_h \geqslant \hat{P}_h(s,a_1) - \rho_h^{(P)}(s,a_1) \\ & P_h \cdot e_{|\mathcal{S}|} = 1 \\ & P_h \geqslant 0, \end{aligned} \tag{B.23}$$

and in the standard form,

$$\begin{aligned} \min_{P_h} \quad & \overline{Q}_{i,h+1}^T\left(\pi_{h+1}^\dagger\right) P_h \\ \text{such that } & \begin{bmatrix} I_{|\mathcal{S}|} \\ -I_{|\mathcal{S}|} \end{bmatrix} P_h \leqslant \begin{bmatrix} \hat{P}_h(s,a_1) + \rho_h^{(P)}(s,a_1) \\ -\hat{P}_h(s,a_1) + \rho_h^{(P)}(s,a_1) \end{bmatrix} \\ & e_{|\mathcal{S}|}^T P_h = 1 \\ & P_h \geqslant 0, \end{aligned} \tag{B.24}$$

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 following dual prob-

lem:

$$\begin{aligned} &\min_{u,v,w} \begin{bmatrix} \hat{P}_h(s,a_1) + \rho_h^{(P)}(s,a_1) \\ -\hat{P}_h(s,a_1) + \rho_h^{(P)}(s,a_1)\, 1^T \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \\ &\text{such that } \begin{bmatrix} I_{|\mathcal{S}|} \\ -I_{|\mathcal{S}|} \\ e_{|\mathcal{S}|}^T \end{bmatrix}^T \begin{bmatrix} u \\ v \\ w \end{bmatrix} \geqslant \overline{Q}_{i,h+1}\left(\pi_{h+1}^{\dagger}\right) \\ &\begin{bmatrix} u \\ v \end{bmatrix} \geqslant 0, \end{aligned} \tag{B.25}$$

which is equivalent to the following:

$$\begin{aligned} &\min_{u,v,w} \sum_{s' \in \mathcal{S}} \hat{P}_h(s'|s,a_1)(u_{s'} - v_{s'} + \rho_h^{(P)}(s,a_1)(u_{s'} + v_{s'}) + w \\ &\text{such that } u_{s'} - v_{s'} + w \geqslant \overline{Q}_{i,h+1}\big(s', \pi_{h+1}^{\dagger}(s')\big), \forall\, s' \in \mathcal{S}. \\ &\qquad u_{s'}, v_{s'} \geqslant 0, \forall\, s' \in \mathcal{S}, \end{aligned} \tag{B.26}$$

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 a_1 in $\mathcal{A}$.

B.4 Optimal Cost Analysis

Outline. In order to understand the attack cost, we make two critical reductions. The first is relating the attack cost for an entire instance I to the attack costs of each period game I_h. Each period game is essentially just a bandit game, so this reduces the task of analyzing the cost for a full Markov Game instance down to the task of analyzing a bandit game instance. Then, we reduce solving a bandit game instance to a mechanism design problem. In particular, we show that the cost of poisoned rewards

$r^\dagger$ is closely related to the cost of its corresponding MLE $R^\dagger$. Thus, we can focus on optimizing the MLE rewards, which is just a mechanism design problem. Most of the results then follow from constructing particular mechanisms for normal-form games; including an optimal mechanism for installing a large margin-DSE in normal-form games. This optimal mechanism is equivalent to solving the bandit attack problem for the special case when $\underline{N} = \overline{N} = 1$, but its cost is much easier to see. We present these ideas in reverse order to build up from the easier problems to the harder ones.

First, note that the L^1 cost function for a specific poisoning $r^\dagger$, $C\left(r^0, r^\dagger\right)$ can be written as,

$$\sum_{h=1}^{H}\sum_{i=1}^{n}\sum_{s\in\mathcal{S}}\sum_{a_1\in\mathcal{A}}\sum_{k=1}^{K}\mathbf{1}_{\left\{s_h^{(k)}=s,\mathbf{a}_h^{(k)}=a_1\right\}}\left|r_{i,h}^{\dagger,(k)}-r_{i,h}^{(k)}\right|$$

When clear from the context, we just refer to this quantity as C. We see that C is defined by sums over the parameters h, i, s, a_1, and k. For any choice of parameters $p_1,\ldots,p_j$, we let $C_{p_1,\ldots,p_j}$ denote the cost function with those parameters fixed. For example, C_{i,h,s,a_1} denotes the innermost sum $\sum_{k=1}^{K}\mathbf{1}_{\left\{s_h^{(k)}=s,\mathbf{a}_h^{(k)}=a_1\right\}}\left|r_{i,h}^{\dagger,(k)}-r_{i,h}^{(k)}\right|$. We then always have that $C=\sum_{p_1}\cdots\sum_{p_j}C_{p_1,\ldots,p_j}$.

Now, we can similarly define the cost associated with changing the MLE rewards from $\hat{R}$ to $R^\dagger$. We denote this cost by $C^M\left(\hat{R},R^\dagger\right)$, which can be written as,

$$\sum_{h=1}^{H}\sum_{i=1}^{n}\sum_{s\in\mathcal{S}}\sum_{a_1\in\mathcal{A}}\left|R_{i,h}^{\dagger}(s,a_1)-\hat{R}_{i,h}(s,a_1)\right|$$

When clear from the context, we just refer to this quantity as C^M. We can partition this cost function by parameter just as we did with C.

We start by showing solving the attack problem can be reduced to

mechanism design.

Lemma B.1. *For each* h*, let* $\mathbf{R}_h^\dagger : \mathcal{S} \times \mathcal{A} \to [-b, b]^n$*. Then, for every choice of bounded rewards* $r^\dagger$ *whose MLE is* $R^\dagger$*, we have that,*

$$\sum_{h=1}^{H} \underline{N}_h C_h^M(\hat{R}, R^\dagger) \leqslant C(r^0, r^\dagger)$$

Also, there exists a choice of $r^\dagger$ *whose MLE is* $R^\dagger$ *and that satisfies,*

$$C(r^\dagger, r^0) \leqslant \sum_{h=1}^{H} \overline{N}_h C_h^M(\hat{R}, R^\dagger)$$

Proof. Fix any h, i, s, a_1, and any rewards $r^\dagger$ whose MLE is $R^\dagger$. By the definition of MLEs and the triangle inequality, we have that,

$$\begin{aligned}
C_{i,h,s,a_1}^M &= \left| R_{i,h}^\dagger(s, a_1) - \hat{R}_{i,h}(s, a_1) \right| \\
&= \Bigg| \frac{1}{N_h(s, a_1)} \sum_{k=1}^{K} \mathbf{1}_{\left\{ s_h^{(k)} = s, \mathbf{a}_h^{(k)} = a_1 \right\}} r_{i,h}^{\dagger,(k)} \\
&\quad - \frac{1}{N_h(s, a_1)} \sum_{k=1}^{K} \mathbf{1}_{\left\{ s_h^{(k)} = s, \mathbf{a}_h^{(k)} = a_1 \right\}} r_{i,h}^{(k)} \Bigg| \\
&= \frac{1}{N_h(s, a_1)} \left| \sum_{k=1}^{K} \mathbf{1}_{\left\{ s_h^{(k)} = s, \mathbf{a}_h^{(k)} = a_1 \right\}} \left(r_{i,h}^{\dagger,(k)} - r_{i,h}^{(k)} \right) \right| \\
&\leqslant \frac{1}{N_h(s, a_1)} \sum_{k=1}^{K} \mathbf{1}_{\left\{ s_h^{(k)} = s, \mathbf{a}_h^{(k)} = a_1 \right\}} \left| r_{i,h}^{\dagger,(k)} - r_{i,h}^{(k)} \right| \\
&= \frac{1}{N_h(s, a_1)} C_{i,h,s,a_1}
\end{aligned}$$

Thus,

$$\begin{aligned} C(r^0, r^\dagger) &= \sum_{i,h,s,a_1} C_{i,h,s,a_1} \\ &\geqslant \sum_{i,h,s,a_1} N_h(s, a_1) C^M_{i,h,s,a_1} \\ &\geqslant \sum_h \underline{N}_h C^M_h(\hat{R}, R^\dagger) \end{aligned}$$

This proves the first claim.

For the second claim, we construct specific rewards $r^\dagger$. Let $E_h(s, a_1) = \{k \mid s_h^{(k)} = s, \mathbf{a}_h^{(k)} = a_1\}$. For each k, s, and a_1 with $s_h^{(k)} = s$ and $\mathbf{a}_h^{(k)} = a_1$, we define $r_{i,h}^{\dagger,(k)}$ depending on the value of $r = R_{i,h}^\dagger(s, a_1) - \hat{R}_{i,h}(s, a_1) + r_{i,h}^{(k)}$,

1. If $r \in [-b, b]$, then we set $r_{i,h}^{\dagger,(k)} := r$
2. If $r > b$, then we set $r_{i,h}^{\dagger,(k)} := b$
3. If $r < -b$, then we set $r_{i,h}^{\dagger,(k)} := -b$

Clearly, this is a feasible choice of rewards. It is easy to see that the MLE of $r^\dagger$ is $R^\dagger$. We also claim that $\max_{k \in E_h(s,a_1)} |r_{i,h}^\dagger - r_{i,h}^0| \leqslant |R_{i,h}^\dagger(s, a_1) - \hat{R}_{i,h}(s, a_1)|$. We show this depending on case.

1. If $r \in [-b, b]$, then by definition we have that,

$$|r_{i,h}^{\dagger,(k)} - r_{i,h}^{(k)}| = |r - r_{i,h}^{(k)}| = |R_{i,h}^\dagger(s, a_1) - \hat{R}_{i,h}(s, a_1)|$$

2. If $r > b$, then by definition we have that,

$$\begin{aligned} &R_{i,h}^\dagger(s, a_1) - \hat{R}_{i,h}(s, a_1) + r_{i,h}^{(k)} > b \\ &\implies b - r_{i,h}^{(k)} < R_{i,h}^\dagger(s, a_1) - \hat{R}_{i,h}(s, a_1) \end{aligned}$$

Also, since all rewards are bounded above by b, we know that $b - r_{i,h}^{(k)} \geqslant 0$ and so $b - r_{i,h}^{(k)} = |b - r_{i,h}^{(k)}|$. Thus,

$$\begin{aligned}|r_{i,h}^{\dagger,(k)} - r_{i,h}^{(k)}| = |b - r_{i,h}^{(k)}| &= b - r_{i,h}^{(k)} \\ &< R_{i,h}^{\dagger}(s, a_1) - \hat{R}_{i,h}(s, a_1) \\ &\leqslant |R_{i,h}^{\dagger}(s, a_1) - \hat{R}_{i,h}(s, a_1)|\end{aligned}$$

3. If $r < -b$, then by definition we have that,

$$\begin{aligned}&R_{i,h}^{\dagger}(s, a_1) - \hat{R}_{i,h}(s, a_1) + r_{i,h}^{(k)} < -b \\ &\implies b + r_{i,h}^{(k)} < \hat{R}_{i,h}(s, a_1) - R_{i,h}^{\dagger}(s, a_1)\end{aligned}$$

Also, since all rewards are bounded below by $-b$, we know that $b + r_{i,h}^{(k)} \geqslant 0$ and so $b + r_{i,h}^{(k)} = |b + r_{i,h}^{(k)}|$. Thus,

$$\begin{aligned}|r_{i,h}^{\dagger,(k)} - r_{i,h}^{(k)}| = |b + r_{i,h}^{(k)}| &= b + r_{i,h}^{(k)} \\ &< \hat{R}_{i,h}(s, a_1) - R_{i,h}^{\dagger}(s, a_1) \\ &\leqslant |R_{i,h}^{\dagger}(s, a_1) - \hat{R}_{i,h}(s, a_1)|\end{aligned}$$

We then see for this choice of $r^{\dagger}$,

$$\begin{aligned}C_{i,h} &= \sum_{s \in \mathcal{S}} \sum_{a_1 \in \mathcal{A}} \sum_{k=1}^{K} \mathbf{1}_{\left\{s_h^{(k)} = s, \mathbf{a}_h^{(k)} = a_1\right\}} \left| r_{i,h}^{\dagger,(k)} - r_{i,h}^{(k)} \right| \\ &\leqslant \sum_{s \in \mathcal{S}} \sum_{a_1 \in \mathcal{A}} N_h(s, a_1) \max_{k \in E_h(s, a_1)} \left| r_{i,h}^{\dagger,(k)} - r_{i,h}^{(k)} \right| \\ &\leqslant \overline{N}_h \sum_{s \in \mathcal{S}} \sum_{a_1 \in \mathcal{A}} |R_{i,h}^{\dagger}(s, a_1) - R_{i,h}(s, a_1)| \\ &= \overline{N}_h C_{i,h}^{M}\end{aligned}$$

Overall, we have,

$$\begin{aligned} C(r^0, r^\dagger) &= \sum_h \sum_i C_{h,i} \\ &\leqslant \sum_h \sum_i \overline{N}_h C^M_{i,h} \\ &= \sum_h \overline{N}_h C^M_h(\hat{R}, R^\dagger) \end{aligned}$$

This proves the second claim.

□

Equipped with this lemma we can focus on poisoning normal-form games and ignore the complexities of the dataset. Define $\epsilon^\iota_{i,h}(s,(a_i,a_{-i})) = \rho^{(R)}_h(s,(a_i,a_{-i})) + \rho^{(R)}_h(s,(\pi^\dagger_{i,h}(s),a_{-i})) + \iota$. Then the dominance gap takes the form,

$$\begin{aligned} d^\iota_{i,h}(s,a_{-i}) := \max_{a_i \neq \pi^\dagger_{i,h}(s)} \Big[&\hat{R}_{i,h}(s,(a_i,a_{-i})) \\ &- \hat{R}_{i,h}\left(s,\left(\pi^\dagger_{i,h}(s),a_{-i}\right)\right) + \epsilon^\iota_{i,h}(s,(a_i,a_{-i}))\Big]_+ \end{aligned}$$

Recall, the intuition behind dominance gaps is that they measure how much $\hat{R}_{i,h}\left(s,\left(\pi^\dagger_{i,h}(s),a_{-i}\right)\right)$ would need to be increased in order to satisfy the ι-dominance constraint. We also define $I^\iota_{i,h}(s,a_{-i}) = \{a_i \mid \hat{R}_{i,h}(s,(a_i,a_{-i})) > b - \epsilon^\iota_{i,h}(s,(a_i,a_{-i}))\}$ and $\delta^\iota_{i,h}(s,a_{-i}) :=$

$$\sum_{\substack{a_i \neq \pi^\dagger_{i,h}(s),\\ a_i \in I^\iota_{i,h}(s,a_{-i})}} \left(\hat{R}_{i,h}(s,(a_i,a_{-i})) - b + \epsilon^\iota_{i,h}(s,(a_i,a_{-i}))\right)$$

In contrast to dominance gaps which focus on increasing the rewards for actions that intersect $\pi^\dagger$, δ measures how much we have to decrease actions disjoint from $\pi^\dagger$ to ensure the ι-dominance constraint holds for extremal reward actions. We then consolidate all these quantities for period h into

the variable $\Delta_h(\iota)$,

$$\Delta_h(\iota) := \sum_{s \in S} \sum_{i=1}^{n} \sum_{a_{-i}} \left(d^{\iota}_{i,h}(s, a_{-i}) + \delta^{\iota}_{i,h}(s, a_{-i}) \right)$$

Now, consider the following algorithm, ATK.

Algorithm 7 ATK(I_h)

1: **for** $s \in \mathcal{S}, i \in [n], a_{-i} \in A_{-i}$ **do**
2: $\quad R^{\dagger}_{i,h}(s, \pi^{\dagger}_{i,h}(s), a_{-i}) \leftarrow \min\{\hat{R}_{i,h}(s, \pi^{\dagger}_{i,h}(s), a_{-i}) + d^{\iota}_{i,h}(s, a_{-i}),\ b\}$
3: $\quad$ **for** $a_i \neq \pi^{\dagger}_{i,h}(s)$ with $\hat{R}_{i,h}(s, (a_i, a_{-i})) > b - \epsilon^{\iota}_{i,h}(s, (a_i, a_{-i}))$ **do**
4: $\quad\quad R^{\dagger}_{i,h}(s, (a_i, a_{-i})) \leftarrow b - \epsilon^{\iota}_{i,h}(s, (a_i, a_{-i}))$
5: **return** $R^{\dagger}_h$

This is the optimal mechanism for poisoning the MLEs of a bandit game and, in particular, for poisoning I_h. We use this algorithm to help prove Lemma 4.3. We also remark that this procedure gives an alternative algorithm to the LP when $\underline{N} = \overline{N} = 1$. The surprising fact that we need to only increase rewards for actions intersecting $\pi^{\dagger}$ and decrease other disjoint rewards is an artifact of using the L^1-norm cost.

Proof of Lemma 4.3.

Proof. We first note that by definition of ATK, $C^{\mathcal{M}}_h(\hat{R}, R^{\dagger}) = \Delta_h(\iota)$. This solution is feasible since line 2 forcibly satisfies the ι-dominance constraint unless the upper bound of b is hit. In which case, we push the reward up to b and decrease the other reward so that the ι-dominance constraint is satisfied. Thus, by Lemma B.1 we have that $C^*(I_h) \leqslant C(r^0, r^{\dagger}) \leqslant \overline{N}_h \Delta_h(\iota)$.

For the lower bound, the intuition is that $R^{\dagger}_{i,h}(s, (\pi^{\dagger}_{i,h}(s), a_{-i}))$ must be increased to be large enough to satisfy the ι-dominance constraint. The amount of increase required is exactly the ι-dominance gap. This gives the first term in $\Delta_h(\iota)$. The second term arises from actions not intersecting

$\pi^\dagger$ that must be decreased since their rewards are too close to b. Fix $s \in \mathcal{S}$, i, and a_{-i}.

We start by showing the cost is lower-bounded by the first term in $\Delta_h(\iota)$, $d^\iota_{i,h}(s, a_{-i})$. We define a^*_i ti be the action,

$$\underset{a_i \neq \pi^\dagger_{i,h}(s)}{\operatorname{argmax}} \left\{ \hat{R}_{i,h}\left(s, (a_i, a_{-i})\right) + \rho^{(R)}_h\left(s, (a_i, a_{-i})\right) + \iota \right\}.$$

We then let,

$$\begin{aligned} a^\star &:= (a^\star_i, a_{-i}), \\ a^\dagger &:= \left(\pi^\dagger_{i,h}(s), a_{-i}\right). \end{aligned}$$

If,

$$\begin{aligned} &\hat{R}_{i,h}\left(s, a^\star\right) + \rho^{(R)}_h\left(s, a^\star\right) \\ &- \hat{R}_{i,h}\left(s, a^\dagger\right) + \rho^{(R)}_h\left(s, a^\dagger\right) + \iota \leqslant 0, \end{aligned}$$

then,

$$d^\iota_{i,h}\left(s, a_{-i}\right) = 0,$$

and we have,

$$\begin{aligned} &\left| R^\dagger_{i,h}\left(s, a^\dagger\right) - \hat{R}_{i,h}\left(s, a^\dagger\right) \right| \\ &+ \left| R^\dagger_{i,h}\left(s, a^\star\right) - \hat{R}_{i,h}\left(s, a^\star\right) \right| \\ &\geqslant 0 = d^\iota_{i,h}\left(s, a_{-i}\right). \end{aligned}$$

Otherwise, the triangle inequality implies,

$$\begin{aligned}
&\left|R^{\dagger}_{i,h}\left(s,a^{\dagger}\right)-\hat{R}_{i,h}\left(s,a^{\dagger}\right)\right|+\left|R^{\dagger}_{i,h}\left(s,a^{\star}\right)-\hat{R}_{i,h}\left(s,a^{\star}\right)\right|\\
&\geqslant\left|R^{\dagger}_{i,h}\left(s,a^{\dagger}\right)-R^{\dagger}_{i,h}\left(s,a^{\star}\right)-\hat{R}_{i,h}\left(s,a^{\dagger}\right)+\hat{R}_{i,h}\left(s,a^{\star}\right)\right|\\
&\geqslant R^{\dagger}_{i,h}\left(s,a^{\dagger}\right)-R^{\dagger}_{i,h}\left(s,a^{\star}\right)-\hat{R}_{i,h}\left(s,a^{\dagger}\right)+\hat{R}_{i,h}\left(s,a^{\star}\right)\\
&\geqslant\hat{R}_{i,h}\left(s,a^{\star}\right)+\rho^{(R)}_{h}\left(s,a^{\star}\right)-\hat{R}_{i,h}\left(s,a^{\dagger}\right)+\rho^{(R)}_{h}\left(s,a^{\dagger}\right)+\iota\\
&=d^{\iota}_{i,h}\left(s,a_{-i}\right).
\end{aligned}$$

The first inequality comes from the triangle inequality, the second comes from, and removing absolute values, and the third from the ι-dominance constraint. Specifically, the ι-dominance constraint is,

$$\begin{aligned}
&R^{\dagger}_{i,h}\left(s,\left(\pi^{\dagger}_{i,h}\left(s\right),a_{-i}\right)\right)-\rho^{(R)}_{h}\left(s,\left(\pi^{\dagger}_{i,h}\left(s\right),a_{-i}\right)\right)\\
&\geqslant R^{\dagger}_{i,h}\left(s,(a_i,a_{-i})\right)+\rho^{(R)}_{h}\left(s,(a_i,a_{-i})\right)+\iota.
\end{aligned}$$

and the third inequality comes from moving the ρ terms to the RHS and moving the $R^{\dagger}$ terms to the LHS. The equality is from the definition of the dominance gap when it is not 0. The takeaway is that in either case, we have,

$$\begin{aligned}
&\left|R^{\dagger}_{i,h}\left(s,a^{\dagger}\right)-\hat{R}_{i,h}\left(s,a^{\dagger}\right)\right|+\left|R^{\dagger}_{i,h}\left(s,a^{\star}\right)-\hat{R}_{i,h}\left(s,a^{\star}\right)\right|\\
&\geqslant d^{\iota}_{i,h}(s,a_{-i})
\end{aligned}$$

Next, we show the cost is lower-bounded by the second term in $\Delta_h(\iota)$, $\delta^{\iota}_{i,h}(s,a_{-i})$. Consider any action $a_i \in I^{\iota}_{i,h}(s,a_{-i})$. Note, by the ι-dominance constraint, it must be the case that $R^{\dagger}_{i,h}(s,a_i,a_{-i}) \leqslant b - \epsilon^{\iota}_{i,h}(s,(a_i,a_{-i}))$, i.e., it was necessary to reduce this reward. Otherwise, we have by the bound constraint $R^{\dagger}_{i,h}(s,\pi^{\dagger}_{i,h}(s),a_{-i}) \leqslant b$ and so $R^{\dagger}_{i,h}(s,\pi^{\dagger}_{i,h}(s),a_{-i}) - R^{\dagger}_{i,h}(s,a_i,a_{-i}) < b - (b - \epsilon^{\iota}_{i,h}(s,(a_i,a_{-i}))) = \epsilon^{\iota}_{i,h}(s,(a_i,a_{-i}))$ which would contradict the ι-dominance constraint. Thus, we have that for such

$a_i \in I^{\iota}_{i,h}(s, a_{-i})$,

$$\begin{aligned}
&|R^{\dagger}_{i,h}(s, a_i, a_{-i}) - \hat{R}_{i,h}(s, (a_i, a_{-i}))| \\
&= \hat{R}_{i,h}(s, (a_i, a_{-i})) - R^{\dagger}_{i,h}(s, a_i, a_{-i}) \\
&\geqslant \hat{R}_{i,h}(s, (a_i, a_{-i})) - b + \epsilon^{\iota}_{i,h}(s, (a_i, a_{-i}))
\end{aligned}$$

Summing over all $a_i \in I^{\iota}_{i,h}(s, a_{-i})$, we have th cost is bounded below by $\delta^{\iota}_{i,h}(s, a_{-i})$.

Lastly, we mention an action could partake in both the $d^{\iota}_{i,h}(s, a_{-i})$ term and $\delta^{\iota}_{i,h}(s, a_{-i})$ part of a term in $\Delta_h(\iota)$. If this were the case, for the ι-dominance constraint to hold it must be the $a^{\dagger}$ part was increased to b and the $a^{\star}$ part was decreased as above and so both lower bounds appear simultaneously (so there is no issue with double counting). Overall, we see that summing over all s, i, and a_{-i} gives $C^{M}_{h}(\hat{R}, R^{\dagger}) \geqslant \Delta_h(\iota)$ for any choice of $R^{\dagger}$. By Lemma B.1 we have that $C^{*}(I_h) \geqslant \underline{N}_h \Delta_h(\iota)$.

□

Now that we have characterized the cost of bandit game instances and better understood their structure through the optimal algorithm, we can move on to general Markov game instances.

Proof of Theorem 4.4.

Proof. For the lower bound, we note that the optimization problem for I includes the optimization problem for optimally poisoning I_H as a special case. Thus, the minimum cost to poison the entire instance I could never be less than the minimum cost to poison the last period instance I_H.

For the upper bound, fix a period h. Fix any minimum cost solution, $r^{\star}_h$, to the attack problem for I_h. Let $\hat{R}^{\star}_h$ be the corresponding MLE. Note for any s, we can increase the rewards $\hat{R}^{\star}_h(s, \pi^{\dagger}_h(s))$ to be $(b, \ldots, b)$ without effecting feasibility. Specifically, if $\pi^{\dagger}_h(s)$ is a ι-DSE in the normal-form

game $\hat{R}^{\star}_h(s,\cdot)$, then it remains one if we increase the rewards for each player in this cell; increasing the rewards only increases the margin of dominance.

To change the data to get such an MLE, we can simply change each of the $N_h(s,\pi^{\dagger}_h(s))$ data points to have reward vector $(b,\dots,b)$. The total cost to do this is at most $2bnN_h(s,\pi^{\dagger}_h(s))$. We may also need to modify each MLE reward by $(H-h+1)\overline{\rho}$ to mitigate accumulating errors. We discuss this more in the next paragraph, but for now, we observe the cost for such changes is at most $\overline{N}H\overline{\rho}nA^n$. Summing over all states, we see that the total cost of this new solution for I_h is $C^*(I_h)+|\mathcal{S}|2bn\overline{N}+\overline{N}H\overline{\rho}|\mathcal{S}|nA^n$. By summing over all h, we get the stated upper bound.

Now, we argue that combining these solutions for each h gives a feasible solution for the overall instance I. Note in this new solution, we always have that $\mathbf{R}^{\dagger}_h(s,\pi^{\dagger}(s))=(b,\dots,b)$. As argued before by induction in the feasibility proof, we have $\overline{Q}_{i,h}(s,a)=(H-h+1)b$. Also, since $\min_{R_{i,h}\in CI^{R^{\dagger}}_{i,h}(s,a_1)} R_{i,h}\geqslant b-\overline{\rho}$, we have that by induction $\underline{Q}_{i,h}(s,(\pi^{\dagger}_{i,h}(s),a_{-i}))\geqslant (H-h+1)(b-\overline{\rho})$. We then have,

$$\begin{aligned}
&\underline{Q}_{i,h}\left(s,a^{\dagger}\right)-\overline{Q}_{i,h}\left(s,a_1\right)\\
&\geqslant \min_{R_{i,h}\in CI^{R^{\dagger}}_{i,h}\left(s,a^{\dagger}\right)} R_{i,h}-\max_{R_{i,h}\in CI^{R^{\dagger}}_{i,h}(s,a_1)} R_{i,h}\\
&+(H-h)(b-\overline{\rho})-(H-h)b\\
&=\min_{R_{i,h}\in CI^{R^{\dagger}}_{i,h}\left(s,a^{\dagger}\right)} R_{i,h}-\max_{R_{i,h}\in CI^{R^{\dagger}}_{i,h}(s,a_1)} R_{i,h}-(H-h)\overline{\rho}
\end{aligned}$$

We also know that for r^*_h the difference will be at least ι. However, to ensure feasibility for the overall instance we need the difference to be at least $\iota+(H-h)\overline{\rho}$. So long as $\overline{\rho}\leqslant\frac{2b-\iota}{H-h+1}$, r^*_h can be modified to attain feasibility for I without breaking the reward bounds, i.e. the dominance margin is attainable. Since this condition is captured by the feasibility condition, we know such a modification is possible. In fact, the most wasteful such option is to simply reduce every action not intersecting

with $\pi^{\dagger}_{h}(s)$ by $(H-h)\overline{\rho}$ without going below $-b$ and similarly increase all entries that do intersect by $(H-h)\overline{\rho}$ without going above b. This can be seen as a feasible solution to I_h that has the larger dominance parameter $\iota + (H-h)\overline{\rho}$ instead of just ι. Hence, there is a feasible solution that has exactly the cost of the above. Consequently, we have that

$$C^*(I) \leqslant \sum_{h=1}^{H} C^*(I_h) + 2bnH|\mathcal{S}|\overline{N} + H^2\overline{\rho}|\mathcal{S}|nA^n\overline{N}$$

□

So, we see a prominent effect from $\overline{\rho}$. If the uncertainty of the learners is small, then poisoning is slightly more than poisoning bandit instances independently. This is desirable since it allows us to solve the much easier bandit instances separately instead of the complicated full LP. On the other hand, if the uncertainty is high, then very little can be said about the cost of the solution through relative bounds. Thus, we turn to the universal bounds.

Proof of Theorem 4.3. We now move to the proof of Theorem 4.3 which gives concrete bounds on the cost of any instance.

Proof. The lower bound follows immediately from using any non-negative cost function, the L^1-norm in our case. For the upper bound, we consider a specific solution. Fix a period h and state s and consider the solution,

1. $R^{\dagger}_{i,h}(s, a_1) = b$, if $a_i = \pi^{\dagger}_{i,h}(s)$
2. $R^{\dagger}_{i,h}(s, a_1) = -b$, if $a_i \neq \pi^{\dagger}_{i,h}(s)$

This is the most extreme attack. If any attack is feasible so is this one. For any feasible solution, we can perturb the solution to this one while maintaining feasibility. Specifically, increasing rewards of dominating

actions can only increase the margin of domination and similarly by reducing dominated actions. Hence, this solution is feasible if I is a feasible instance. The cost of this attack is bound simply. For each h, s, i, and a_1, $|R^\dagger_{i,h}(s,a_1) - \hat{R}_{i,h}(s,a_1)| \leqslant 2b$. Thus, $C^{\mathcal{M}}(\hat{R}, R^\dagger) \leqslant H|\mathcal{S}|nA^n 2b$. Then, applying Lemma B.1 implies that $C^*(I) \leqslant \overline{N}H|\mathcal{S}|nA^n 2b$.

□

Now that we have a good grasp of the cost to poison in general, we look at more structured instances. We show that a very simple structural assumption can tell us a good deal about the cost.

Proof of Lemma 4.2.

Proof. Fix $h < H$, s, i, a_i, and some a_{-i}. We use the notation

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

By assumption, we know that some uniform transition $U \in CI^P_h(s,a_1)$ for each s and a_1. Being uniform, $U(s' \mid s, a_1) = \frac{1}{|\mathcal{S}|}$ for all s'. For the particular choice of $P_h = U$, we get an upper bound on the minimum over all transitions in the equation defining $\underline{Q}$ and thus an upper bound on $\underline{Q}$. Similarly, this choice of P_h gives a lower bound on the max over all transitions in the equation defining $\overline{Q}$ and thus an upper bound on $\overline{Q}$. In symbols, these bounds are,

$$\begin{aligned}\underline{Q}_{i,h}(s,a_1) \leqslant & \min_{R_{i,h} \in CI^{R^\dagger}_{i,h}(s,a_1)} R_{i,h} \\ & + \sum_{s' \in \mathcal{S}} \frac{1}{|\mathcal{S}|} \underline{Q}_{i,h+1}\left(s', \pi^\dagger_{h+1}(s')\right)\end{aligned}$$

$$\overline{Q}_{i,h}(s,a_1) \geqslant \max_{R_{i,h}\in CI^{R^\dagger}_{i,h}(s,a_1)} R_{i,h} + \sum_{s'\in\mathcal{S}} \frac{1}{|\mathcal{S}|}\overline{Q}_{i,h+1}\left(s', \pi^\dagger_{h+1}(s')\right)$$

Consequently, we have that,

$$\begin{aligned}
&\min_{R_{i,h}\in CI^{R^\dagger}_{i,h}(s,a^\dagger)} R_{i,h} - \max_{R_{i,h}\in CI^{R^\dagger}_{i,h}(s,a_1)} R_{i,h} \\
&\geqslant \min_{R_{i,h}\in CI^{R^\dagger}_{i,h}(s,a^\dagger)} R_{i,h} - \max_{R_{i,h}\in CI^{R^\dagger}_{i,h}(s,a_1)} R_{i,h} \\
&+ \sum_{s'\in\mathcal{S}} \frac{1}{|\mathcal{S}|}(\underline{Q}_{i,h+1}(s', \pi^\dagger_{h+1}(s')) - \overline{Q}_{i,h+1}(s', \pi^\dagger_{h+1}(s'))) \\
&= \min_{R_{i,h}\in CI^{R^\dagger}_{i,h}(s,a^\dagger)} R_{i,h} + \sum_{s'\in\mathcal{S}} \frac{1}{|\mathcal{S}|}\underline{Q}_{i,h+1}\left(s', \pi^\dagger_{h+1}(s')\right) \\
&- \max_{R_{i,h}\in CI^{R^\dagger}_{i,h}(s,a_1)} R_{i,h} + \sum_{s'\in\mathcal{S}} \frac{1}{|\mathcal{S}|}\overline{Q}_{i,h+1}\left(s', \pi^\dagger_{h+1}(s')\right) \\
&\geqslant \underline{Q}_{i,h}\left(s, a^\dagger\right) - \overline{Q}_{i,h}(s, a_1) \\
&\geqslant \iota
\end{aligned}$$

The first inequality uses the fact that by definition, $\overline{Q}_{i,h+1}\left(s', \pi^\dagger_{h+1}(s')\right) \geqslant \underline{Q}_{i,h+1}\left(s', \pi^\dagger_{h+1}(s')\right)$ for all s' and so we have added a non-positive term. The second inequality uses the bounds on $\underline{Q}_{i,h}$ and $\overline{Q}_{i,h}$ we derived above. The last inequality uses the ι-dominance constraint of the attack problem.

For the last period H, the definition of the attack problem is exactly the definition of the attack problem for I_H. Thus, we see that any feasible solution to the instance I implies all the constraints defining the attack problem for the period instance I_h are satisfied; namely, the ι-dominance constraint and the reward bound constraints. This implies that a feasible

$\mathcal{A}_1 \setminus \mathcal{A}_2$	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

solution to I consists of feasible solutions to each I_h. Consequently,

$$C^*(I) \geqslant \sum_{h=1}^{H} C^*(I_h)$$

as was to be shown.

□

We see that uniform transitions cause the instance to effectively decouple into independent period instances. In fact, when $\rho^{(R)} = \rho^{(P)} = 0$ and the MLE transition of the data is uniform, the proof above along with the proof of the instance-dependent upper bound actually implies that the minimum cost is exactly the sum of the minimum costs of the period instances. Aside from that observation, we can use the above result to derive a particularly high-cost instance.

Proof of Theorem 4.5.

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, with uniform transitions, meaning,

$$\hat{P}_h(s'|s, a_1) = \frac{1}{|\mathcal{S}|}, \forall\, h \in [H], s', s \in \mathcal{S}, a_1 \in \mathcal{A}.$$

Given $\rho_h^{(R)}(s, a_1) = \rho$ for each $h \in [H]$, $s \in \mathcal{S}$ and $a_1 \in \mathcal{A}$, $d_{i,h}^{\iota}(s, a_{-i})$ is,

$$\begin{aligned}
&\max_{a_i \neq \pi_{i,h}^{\dagger}(s)} \left\{ \begin{array}{l} \hat{R}_{i,h}(s, (a_i, a_{-i})) + \rho_h^{(R)}(s, (a_i, a_{-i})) \\ -\hat{R}_{i,h}\left(s, \left(\pi_{i,H}^{\dagger}(s), a_{-i}\right)\right) \\ +\rho_h^{(R)}\left(s, \left(\pi_{i,H}^{\dagger}(s), a_{-i}\right)\right) + \iota \end{array} \right\} \\
&= \max_{a_i \neq \pi_{i,h}^{\dagger}(s)} \{b + \rho - (-b) + \rho + \iota\} \\
&= 2b + 2\rho + \iota,
\end{aligned}$$

and thus, by Lemma 4.2 and Lemma 4.3,

$$\begin{aligned}
C^*(I) &\geqslant \sum_{h=1}^{H} C^*(I_h) \\
&\geqslant \sum_{h=1}^{H} \underline{N}_h \Delta_h(\iota) \\
&\geqslant \underline{N} \sum_{h=1}^{H} \sum_{s \in \mathcal{S}} \sum_{i=1}^{n} \sum_{a_{-i} \in \mathcal{A}_{-i}} (2b + 2\rho + \iota) \\
&= H|\mathcal{S}|\underline{N} n A^{n-1}(2b + 2\rho + \iota).
\end{aligned}$$

□

In fact, we can generalize this example to a family of instances having high costs. We can shift each entry from b to some $b - \epsilon$ and still maintain exponential cost. Even further, it can be shown that uniformly random games have exponential costs in expectation. This follows since uniformly random games have a large "spread" of values and so many entries must be shifted significantly to ensure dominance. Formally, consider a normal-form matrix game and note that $\mathbb{E}[|x-y| \mid x > y] = \frac{1}{3}b$ when $x, y \sim U[-b, b]$. Also, $Pr[x > y] = \frac{1}{2}$. The expected dominance gap for some i and a_{-i} is at

least

$$\mathbb{E}[|x - y| \mid x > y]\Pr[x > y] + 0\Pr[x \leqslant y] = \frac{1}{6}b$$

So, appealing to the lower bound on poisoning bandit games via dominance gaps, we have the expected cost for an ι-DSE attack on a uniformly random matrix game is at least

$$\sum_{i=1}^{n} \sum_{a^{-i}} \frac{1}{6}b = n|A|^{n-1}\frac{b}{6}$$

Thus, many games are costly to attack using the strong solution concept of ι-MPDSE. We encourage that in future work, different solution concepts be considered, such as unique Nash equilibrium.

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

$$\begin{aligned}
\text{such that } & r_i^{\dagger,(k)} - r_i^{0,(k)} \leqslant t_i^{(k)}, \forall\, k, i \\
& r_i^{0,(k)} - r_i^{\dagger,(k)} \leqslant -t_i^{(k)}, \forall\, k, i \\
& R_i^{\dagger}(a_1) = \frac{1}{N(a_1)} \sum_{k=1}^{K} r_i^{\dagger,(k)} \mathbf{1}_{\{\mathbf{a}^{(k)}=a_1\}}, \forall\, a_1, i \\
& -\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}) \\
& \qquad \leqslant -2b - \iota, \forall\, i, a_{-i}, a_i \neq \pi_i^{\dagger} \\
& -\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}) \\
& \qquad + R_i^{\dagger}\left(\pi_i^{\dagger}, a_{-i}\right) - R_i^{\dagger}(a_i, a_{-i}) \\
& \qquad \leqslant -\rho^{(R)}\left(\pi_i^{\dagger}, a_{-i}\right) - \rho^{(R)}(a_i, a_{-i}) - \iota, \forall\, i, a_{-i}, a_i \neq \pi_i^{\dagger} \\
& -\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) \\
& \qquad \leqslant -b - \rho^{(R)}\left(\pi_i^{\dagger}, a_{-i}\right) - \iota, \forall\, i, a_{-i}, a_i \neq \pi_i^{\dagger} \\
& -\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}) \\
& \qquad \leqslant -\rho^{(R)}(a_i, a_{-i}) - b - \iota, \forall\, i, a_{-i}, a_i \neq \pi_i^{\dagger} \\
& -\overline{m}_i^{-}\left(\pi_i^{\dagger}, a_{-i}\right) - R_i^{\dagger}\left(\pi_i^{\dagger}, a_{-i}\right) \leqslant \rho^{(R)}\left(\pi_i^{\dagger}, a_{-i}\right) - b, \forall\, i, a_{-i}, a_i \neq \pi_i^{\dagger} \\
& -\overline{m}_i^{+}\left(\pi_i^{\dagger}, a_{-i}\right) + R_i^{\dagger}\left(\pi_i^{\dagger}, a_{-i}\right) \leqslant -\rho^{(R)}\left(\pi_i^{\dagger}, a_{-i}\right) + b, \forall\, i, a_{-i}, a_i \neq \pi_i^{\dagger} \\
& -\underline{m}_i^{-}(a_i, a_{-i}) + R_i^{\dagger}\left(\pi_i^{\dagger}, a_{-i}\right) \leqslant -\rho^{(R)}(a_i, a_{-i}) + b, \forall\, i, a_{-i}, a_i \neq \pi_i^{\dagger} \\
& -\underline{m}_i^{+}(a_i, a_{-i}) - R_i^{\dagger}\left(\pi_i^{\dagger}, a_{-i}\right) \leqslant \rho^{(R)}(a_i, a_{-i}) - b, \forall\, i, a_{-i}, a_i \neq \pi_i^{\dagger} \\
& r_i^{\dagger,(k)} \leqslant b, \forall\, k, i \\
& -r_i^{\dagger,(k)} \leqslant -b, \forall\, k, i \\
& \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}) \geqslant 0, \forall\, i, a_{-i}, a_i \neq \pi_i^{\dagger}.
\end{aligned} \tag{B.20}$$

$$
\begin{aligned}
&\min_{r^{\dagger},t,u,v,w} \sum_{i=1}^{n} \sum_{k=1}^{K} \sum_{k=1}^{K} t_{i,h}^{(k)} \\
&\text{such that } r_{i,h}^{\dagger,(k)} - r_{i,h}^{(k)} \leqslant t_{i,h}^{(k)}, \forall\, h, k, i \\
&\qquad r_{i,h}^{(k)} \leqslant -t_{i,h}^{(k)}, \forall\, h, k, i \\
&\qquad R_{i,h}^{\dagger}(s, a_1) = \frac{1}{N_h(s, a_1)} \sum_{k=1}^{K} r_{i,h}^{\dagger,(k)} \mathbf{1}_{\left\{s_h^{(k)} = s, \mathbf{a}_h^{(k)} = a_1\right\}}, \forall\, h, s, i, a_1 \\
&\qquad \underline{Q}_{i,H}(s, a_1) = R_{i,H}^{\dagger}(s, a_1) - \rho_H^{(R)}(s, a_1), \forall\, s, i, a_1 \\
&\qquad \overline{Q}_{i,H}(s, a_1) = R_{i,H}^{\dagger}(s, a_1) + \rho_H^{(R)}(s, a_1), \forall\, s, i, a_1 \\
&\qquad \underline{Q}_{i,h}(s, a_1) = R_{i,h}^{\dagger}(s, a_1) - \rho_h^{(R)}(s, a_1) \\
&\qquad\qquad - \sum_{s' \in \mathcal{S}} \hat{P}_h(s'|s, a_1) \left[\underline{u}_{i,h}(s, a_1) - \underline{v}_{i,h}(s, a_1)\right]_{s'} \\
&\qquad\qquad - \left\{ \sum_{s' \in \mathcal{S}} \rho_h^{(P)}(s, a_1) \left[\underline{u}_{i,h}(s, a_1) + \underline{v}_{i,h}(s, a_1)\right]_{s'} \right\} - \underline{w}_{i,h}(s, a_1), \forall\, a_1 \\
&\qquad \overline{Q}_{i,h}(s, a_1) = R_{i,h}^{\dagger}(s, a_1) + \rho_h^{(R)}(s, a_1) \\
&\qquad\qquad + \sum_{s' \in \mathcal{S}} \hat{P}_h(s'|s, a_1) \left[\overline{u}_{i,h}(s, a_1) - \overline{v}_{i,h}(s, a_1)\right]_{s'} \\
&\qquad\qquad + \left\{ \sum_{s' \in \mathcal{S}} \rho_h^{(P)}(s, a_1) \left[\overline{u}_{i,h}(s, a_1) + \overline{v}_{i,h}(s, a_1)\right]_{s'} \right\} + \overline{w}_{i,h}(s, a_1), \forall\, a_1 \\
&\qquad -\underline{Q}_{i,h+1}\left(s', \pi_{h+1}^{\dagger}(s')\right) \leqslant \left[\underline{u}_{i,h}(s, a_1) - \underline{v}_{i,h}(s, a_1) + \underline{w}_{i,h}(s, a_1)\right]_{s'}, \forall\, h, s, s', i, a_1 \\
&\qquad \overline{Q}_{i,h+1}\left(s', \pi_{h+1}^{\dagger}(s')\right) \leqslant \left[\overline{u}_{i,h}(s, a_1) - \overline{v}_{i,h}(s, a_1) + \overline{w}_{i,h}(s, a_1)\right]_{s'}, \forall\, h, s, s', i, a_1 \\
&\qquad \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) \leqslant -\iota, \forall\, h, s, i, a_{-i}, a_i \neq \pi_{i,h}^{\dagger}(s) \\
&\qquad r_{i,h}^{\dagger,(k)} \leqslant b, \forall\, h, k, i \\
&\qquad -r_{i,h}^{\dagger,(k)} \leqslant -b, \forall\, h, k, i \\
&\qquad \left[\underline{u}_{i,h}(s, a_1)\right]_{s'}, \left[\underline{v}_{i,h}(s, a_1)\right]_{s'}, \left[\overline{u}_{i,h}(s, a_1)\right]_{s'}, \left[\overline{v}_{i,h}(s, a_1)\right]_{s'} \geqslant 0, \forall\, h, s, s', i, a_1
\end{aligned}
\tag{B.22}
$$

C OFFLINE REWARD POISONING FOR ZERO-SUM GAMES TO INSTALL A NASH EQUILIBRIUM

C.1 Supplementary Material

Proof of Proposition 5.1 and Theorem 5.1

We show that for zero-sum games, strict MPEs are MPEs and they are unique. We use the following definition of MPE and strict MPE for zero-sum games rewritten in terms of Q functions. Proposition 5.1 is a special case of Theorem 5.1 with $H = |\mathcal{S}| = 1$.

Definition C.1. (*Markov Perfect Equilibrium for Zero-sum Games*)$\pi^\dagger$ *is a MPE if for each* $h \in [H], s \in \mathcal{S}$,

$$Q_h^{\pi^\dagger}\left(s, \pi_h^\dagger(s)\right) \geqslant Q_h^{\pi^\dagger}\left(s, \left(a_1, \pi_{2,h}^\dagger(s)\right)\right), \forall\, a_1 \neq \pi_{1,h}^\dagger(s), \tag{C.1}$$

$$Q_h^{\pi^\dagger}\left(s, \pi_h^\dagger(s)\right) \leqslant Q_h^{\pi^\dagger}\left(s, \left(\pi_{1,h}^\dagger(s), a_2\right)\right), \forall\, a_2 \neq \pi_{2,h}^\dagger(s). \tag{C.2}$$

Definition C.2. (*Strict Markov Perfect Equilibrium for Zero-sum Games*)$\pi^\dagger$ *is a strict MPE if for each* $h \in [H], s \in \mathcal{S}$,

$$Q_h^{\pi^\dagger}\left(s, \pi_h^\dagger(s)\right) > Q_h^{\pi^\dagger}\left(s, \left(a_1, \pi_{2,h}^\dagger(s)\right)\right), \forall\, a_1 \neq \pi_{1,h}^\dagger(s), \tag{C.3}$$

$$Q_h^{\pi^\dagger}\left(s, \pi_h^\dagger(s)\right) < Q_h^{\pi^\dagger}\left(s, \left(\pi_{1,h}^\dagger(s), a_2\right)\right), \forall\, a_2 \neq \pi_{2,h}^\dagger(s). \tag{C.4}$$

Proof. Fix a period $h \in [H]$, and assume in periods $h+1, h+2, ..., H, \pi^\dagger$ is the unique NE in every state $s \in \mathcal{S}$. This is vacuously true in period H. First, $\pi_h^\dagger(s)$ is a NE since (C.3) implies (C.1) and (C.4) implies (C.2). Now, for a contradiction, assume $\left(a_1', a_2'\right) \neq \pi_h^\dagger(s)$ is another NE in the

stage game in period h in some state $s \in \mathcal{S}$, then,

$$Q_h^{\pi^\dagger}\left(s, \left(a_1', a_2'\right)\right) \geqslant Q_h^{\pi^\dagger}\left(s, \left(\pi_{1,h}^\dagger(s), a_2'\right)\right), \tag{C.5}$$

$$Q_h^{\pi^\dagger}\left(s, \left(a_1', a_2'\right)\right) \leqslant Q_h^{\pi^\dagger}\left(s, \left(a_1', \pi_{2,h}^\dagger(s)\right)\right). \tag{C.6}$$

From the strict MPE conditions,

$$Q_h^{\pi^\dagger}\left(s, \pi_h^\dagger(s)\right) \overset{(C.3)}{>} Q_h^{\pi^\dagger}\left(s, \left(\pi_{1,h}^\dagger(s), a_2'\right)\right), \tag{C.7}$$

$$Q_h^{\pi^\dagger}\left(s, \pi_h^\dagger(s)\right) \overset{(C.4)}{<} Q_h^{\pi^\dagger}\left(s, \left(a_1', \pi_{2,h}^\dagger(s)\right)\right). \tag{C.8}$$

Combine the above inequalities, we get,

$$Q_h^{\pi^\dagger}\left(s, \pi_h^\dagger(s)\right) \overset{(C.5),\,(C.7)}{>} Q_h^{\pi^\dagger}\left(s, \left(a_1', a_2'\right)\right), \tag{C.9}$$

$$Q_h^{\pi^\dagger}\left(s, \pi_h^\dagger(s)\right) \overset{(C.6),\,(C.8)}{<} Q_h^{\pi^\dagger}\left(s, \left(a_1', a_2'\right)\right), \tag{C.10}$$

which is a contradiction.
Therefore, $\pi^\dagger$ is the unique NE in period h, state s. Since h and s are arbitrary, $\pi^\dagger$ is the unique MPE.

□

Proof of Proposition 5.2 and Theorem 5.2

We first write out the complete optimization problem for (5.23) in Example 5.9, then we show that the optimization is a relaxation by showing for any $Q^{\pi^\dagger} \in \left[\underline{Q}^{\pi^\dagger}, \overline{Q}^{\pi^\dagger}\right]$ elememtwise, $\pi^\dagger$ is a strict MPE, and as a result Theorem 5.1 implies its uniqueness. The proof that the problem can be converted into a linear program is similar to LP conversions in (Wu et al., 2023b). We do not write out the complete LP, and instead we show that each constraint can be converted into a linear constraint. Theorem 5.2 is a special case of (5.23) with given $\underline{Q}^{\pi^\dagger}$ and $\overline{Q}^{\pi^\dagger}$ that are not derived

from the rewards and transitions, and Proposition 5.2 is a special case of Theorem 5.2 when $H = |\mathcal{S}| = 1$.

$$\min_{r^\dagger \in [0,1]^{HK}} \sum_{k=1}^{K} \sum_{h=1}^{H} \left| r_h^{\dagger,(k)} - r_h^{(k)} \right|$$

$$\text{subject to } R_h(s,\mathbf{a}) = \frac{\sum_{k=1}^{K} \sum_{h=1}^{H} r_h^{\dagger,(k)} \mathbb{I}_{\left\{s_h^{(k)}=s, \mathbf{a}_h^{(k)}=\mathbf{a}\right\}}}{\max\{N_h(s,\mathbf{a}), 1\}}, \tag{C.11}$$

$$\forall\, h \in [H], s \in \mathcal{S}, \mathbf{a} \in \mathcal{A}, \tag{C.12}$$

$$P_h(s'|s,\mathbf{a}) = \frac{\sum_{k=1}^{K} \mathbb{I}_{\left\{s_{h+1}^{(k)}=s', s_h^{(k)}=s, \mathbf{a}_h^{(k)}=\mathbf{a}\right\}}}{N_h(s,\mathbf{a})} \text{ or } \frac{1}{|\mathcal{S}|} \text{ if } N_h(s,\mathbf{a}) = 0, \tag{C.13}$$

$$\forall\, h \in [H], s \in \mathcal{S}, \mathbf{a} \in \mathcal{A}, \tag{C.14}$$

$$\underline{Q}_h^{\pi^\dagger}(s,\mathbf{a}) = \min_{R \in \mathcal{C}_{i,h}^{(R)}(s,\mathbf{a})} R + \min_{P \in \mathcal{C}_h^{(P)}(s,\mathbf{a})} \sum_{s' \in \mathcal{S}} P(s')\, \underline{Q}_{h+1}^{\pi^\dagger}\left(s', \pi_{h+1}^\dagger(s')\right), \tag{C.15}$$

$$\forall\, h \in [H], s \in \mathcal{S}, \mathbf{a} \in \mathcal{A}, \tag{C.16}$$

$$\overline{Q}_h^{\pi^\dagger}(s,\mathbf{a}) = \max_{R \in \mathcal{C}_{i,h}^{(R)}(s,\mathbf{a})} R + \max_{P \in \mathcal{C}_h^{(P)}(s,\mathbf{a})} \sum_{s' \in \mathcal{S}} P(s')\, \overline{Q}_{h+1}^{\pi^\dagger}\left(s', \pi_{h+1}^\dagger(s')\right), \tag{C.17}$$

$$\forall\, h \in [H], s \in \mathcal{S}, \mathbf{a} \in \mathcal{A}, \tag{C.18}$$

$$\underline{Q}_{H+1}^{\pi^\dagger}(s,\mathbf{a}) = \overline{Q}_{H+1}^{\pi^\dagger}(s,\mathbf{a}) = 0, \qquad \forall\, s \in \mathcal{S}, \mathbf{a} \in \mathcal{A},$$

$$\underline{Q}_h^{\pi^\dagger}\left(s, \pi_h^\dagger(s)\right) \geqslant \overline{Q}_h^{\pi^\dagger}\left(s, \left(a_1, \pi_{2,h}^\dagger(s)\right)\right) + \iota, \tag{C.19}$$

$$\forall\, h \in [H], s \in \mathcal{S}, a_1 \neq \pi_{1,h}^\dagger(s), \tag{C.20}$$

$$\overline{Q}_h^{\pi^\dagger}\left(s, \pi_h^\dagger(s)\right) \leqslant \underline{Q}_h^{\pi^\dagger}\left(s, \left(\pi_{1,h}^\dagger(s), a_2\right)\right) - \iota, \tag{C.21}$$

$$\forall\, h \in [H], s \in \mathcal{S}, a_2 \neq \pi_{2,h}^\dagger(s). \tag{C.22}$$

Since we evaluate the $\underline{Q}$ and $\overline{Q}$ functions on the policy $\pi^\dagger$, we add superscript $\pi^\dagger$ on $\underline{Q}$ and $\overline{Q}$ inside the optimization for clarity.

Proof. Take any $R \in \mathcal{C}^{(R)}$ and $P \in \mathcal{C}^{(P)}$, due to the definition of $\underline{Q}^{\pi^\dagger}$ and $\overline{Q}^{\pi^\dagger}$, which are replicated in (C.16) and (C.18), we know that, for each $h \in [H], s \in \mathcal{S}, \mathbf{a} \in \mathcal{A}$,

$$\underline{Q}_h^{\pi^\dagger}(s, \mathbf{a}) \leqslant Q_h^{\pi^\dagger}(s, \mathbf{a}) \leqslant \overline{Q}_h^{\pi^\dagger}(s, \mathbf{a}). \tag{C.23}$$

Fix period $h \in [H]$, and assume in periods $h+1, h+2, ..., H$, $\pi^\dagger$ is the Nash equilibrium in every state $s \in \mathcal{S}$. This is vacuously true in period H.
For a fixed $s \in \mathcal{S}$, for any $a_1 \neq \pi_{1,h}^\dagger(s)$,

$$\begin{aligned} Q_h^{\pi^\dagger}\left(s, \pi_h^\dagger(s)\right) &\overset{(C.23)}{\geqslant} \underline{Q}_h^{\pi^\dagger}\left(s, \pi_h^\dagger(s)\right) \\ &\overset{(C.20)}{\geqslant} \overline{Q}_h^{\pi^\dagger}\left(s, \left(a_1, \pi_{2,h}^\dagger(s)\right)\right) + \iota \\ &\overset{(C.23)}{\geqslant} Q_h^{\pi^\dagger}\left(s, \left(a_1, \pi_{2,h}^\dagger(s)\right)\right) + \iota, \end{aligned} \tag{C.24}$$

and for any $a_2 \neq \pi_{2,h}^\dagger(s)$,

$$\begin{aligned} Q_h^{\pi^\dagger}\left(s, \pi_h^\dagger(s)\right) &\overset{(C.23)}{\leqslant} \overline{Q}_h^{\pi^\dagger}\left(s, \pi_h^\dagger(s)\right) \\ &\overset{(C.22)}{\leqslant} \underline{Q}_h^{\pi^\dagger}\left(s, \left(\pi_{1,h}^\dagger(s), a_2\right)\right) - \iota \\ &\overset{(C.23)}{\geqslant} Q_h^{\pi^\dagger}\left(s, \left(\pi_{2,h}^\dagger(s), a_2\right)\right) - \iota, \end{aligned} \tag{C.25}$$

(C.24) and (C.25) imply that $\pi_h^\dagger(s)$ is the Nash equilibrium in period h state s.
Therefore, $Q^{\pi^\dagger} \in \underline{\mathcal{U}}\left(\pi^\dagger; \iota\right)$, and by Theorem 5.1, $\pi^\dagger$ is the unique MPE.
Now, to show that the problem can be converted into an LP, we note that (C.12) is linear in $r^\dagger$, (C.14) is independent of $r^\dagger$, (C.20) and (C.22) are

linear in $\underline{Q}$ and $\overline{Q}$. Therefore, we only have to convert (C.16) and (C.18), which define $\underline{Q}$ and $\overline{Q}$ into linear constraints in $r^{\dagger}$, in particular, we convert the following linear program, for some $h \in [H], s \in \mathcal{S}, \mathbf{a} \in \mathcal{A}$,

$$\begin{aligned}
&\min_{P} \sum_{s' \in \mathcal{S}} P(s') \, \underline{Q}_{h+1}^{\pi^{\dagger}} \left(s', \pi_{h+1}^{\dagger}(s')\right) \\
&\text{subject to } P(s') \leqslant \hat{P}_h(s'|s, \mathbf{a}) + \rho_h^{(P)}(s, \mathbf{a}), \forall\, s' \in \mathcal{S}, \\
&\qquad P(s') \geqslant \hat{P}_h(s'|s, \mathbf{a}) - \rho^{(P)}(s, \mathbf{a}), \forall\, s' \in \mathcal{S}, \\
&\qquad \sum_{s' \in \mathcal{S}} P(s') = 1, \\
&\qquad P(s') \geqslant 0, \forall\, s' \in \mathcal{S},
\end{aligned}$$

into its dual problem,

$$\begin{aligned}
&\max_{\underline{u} \in \mathbb{R}^{\mathcal{S}}, \underline{v} \in \mathbb{R}^{\mathcal{S}}, \underline{w} \in \mathbb{R}} \sum_{s' \in \mathcal{S}} \hat{P}_h(s'|s, \mathbf{a}) (\underline{u}_{s'} - \underline{v}_{s'}) + \rho_h^{(P)}(s, \mathbf{a}) (\underline{u}_{s'} + \underline{v}_{s'}) + \underline{w} \\
&\text{subject to } \underline{u}_{s'} - \underline{v}_{s'} + \underline{w} \geqslant -\underline{Q}_{h+1}^{\pi^{\dagger}} \left(s', \pi_{h+1}^{\dagger}(s')\right), \forall\, s' \in \mathcal{S}, \\
&\qquad \underline{u}_{s'} \geqslant 0, \underline{v}_{s'} \geqslant 0, \forall\, s' \in \mathcal{S}.
\end{aligned}$$

Therefore, (C.16) can be rewritten as the following linear constraints,

$$\begin{aligned}
&\underline{Q}_h^{\pi^{\dagger}}(s, \mathbf{a}) = R_h(s, \mathbf{a}) - \rho_h^{(R)}(s, \mathbf{a}) + \sum_{s' \in \mathcal{S}} \hat{P}_h(s'|s, \mathbf{a}) (\underline{u}_{s'} - \underline{v}_{s'}) + \rho_h^{(P)}(s, \mathbf{a}) (\underline{u}_{s'} + \underline{v}_{s'}) + \underline{w}, \\
&\underline{u}_{s'} - \underline{v}_{s'} + \underline{w} \geqslant -\underline{Q}_{h+1}^{\pi^{\dagger}} \left(s', \pi_{h+1}^{\dagger}(s')\right), \forall\, s' \in \mathcal{S}, \\
&\qquad \underline{u}_{s'} \geqslant 0, \underline{v}_{s'} \geqslant 0, \forall\, s' \in \mathcal{S}.
\end{aligned}$$

The similar dual problem can be written out for the $\overline{Q}$ to replace (C.18),

$$\overline{Q}_h^{\pi^\dagger}(s,\mathbf{a}) = R_h(s,\mathbf{a}) + \rho_h^{(R)}(s,\mathbf{a}) + \sum_{s'\in\mathcal{S}} \hat{P}_h(s'|s,\mathbf{a})(\overline{u}_{s'} - \overline{v}_{s'}) + \rho_h^{(P)}(s,\mathbf{a})(\overline{u}_{s'} + \overline{v}_{s'}) + \overline{w},$$

$$\overline{u}_{s'} - \overline{v}_{s'} + \overline{w} \geqslant \overline{Q}_{h+1}^{\pi^\dagger}\left(s', \pi_{h+1}^\dagger(s')\right), \forall\, s' \in \mathcal{S},$$

$$\overline{u}_{s'} \geqslant 0, \overline{v}_{s'} \geqslant 0, \forall\, s' \in \mathcal{S}.$$

The linearization of the other $\underline{Q}$ and $\overline{Q}$ constraints are similar.

□

Proof of Theorem 5.3

Again, we write the proof for (5.23) in Example 5.9, and Theorem 5.3 is a special case with given $\underline{Q}^{\pi^\dagger}$ and $\overline{Q}^{\pi^\dagger}$ that are not derived from the rewards and transitions. In particular, setting $\rho^{(Q)} = \rho^{(R)}$ and $\rho^{(P)} = 0$ would like to the result stated in Theorem 5.3. We first provide the intuition behind the proofs. The proof is at the end of this subsection.

Suppose the target action profile is $(1,1)$ in some state s in period h, we show that the target action profile $(1,1)$ is the unique NE for any $Q_h(s,\cdot) \in \left[\underline{Q}_h(s,\cdot), \overline{Q}_h(s,\cdot)\right]$ under the following attack,

$$r_h^{\dagger,(k)} = \begin{cases} -b & \text{if } a_{1,h}^{(k)} \neq \pi_{1,h}^\dagger\left(s_h^{(k)}\right), a_{2,h}^{(k)} = \pi_{2,h}^\dagger\left(s_h^{(k)}\right) \\ 0 & \text{if } a_{1,h}^{(k)} = \pi_{1,h}^\dagger\left(s_h^{(k)}\right), a_{2,h}^{(k)} = \pi_{2,h}^\dagger\left(s_h^{(k)}\right) \\ b & \text{if } a_{1,h}^{(k)} = \pi_{1,h}^\dagger\left(s_h^{(k)}\right), a_{2,h}^{(k)} \neq \pi_{2,h}^\dagger\left(s_h^{(k)}\right) \\ r_h^{(k)} & \text{otherwise} \end{cases}. \tag{C.26}$$

To simplify the notations, we define the bounds on the cumulative Q value

in period $h+1, h+2, ..., H$ as,

$$\underline{S}_h = \sum_{h'=h+1}^{H} \min_{s' \in \mathcal{S}} \underline{Q}_{h'} \left(s', \pi^{\dagger}_{h'}(s') \right)$$
$$\overline{S}_h = \sum_{h'=h+1}^{H} \max_{s' \in \mathcal{S}} \overline{Q}_{h'} \left(s', \pi^{\dagger}_{h'}(s') \right)$$

$\underline{Q}_h(s)$ is lower bounded by,

$\mathcal{A}_1 \setminus \mathcal{A}_2$	1	2	...	$\lvert\mathcal{A}_2\rvert$
1	$0 - \rho^{(R)}_h(s,(1,1)) + \underline{S}_h$	$b - \rho^{(R)}_h(s,(1,2)) + \underline{S}_h$	...	$b - \rho^{(R)}_h(s,(1,\lvert\mathcal{A}_2\rvert)) + \underline{S}_h$
2	$-b - \rho^{(R)}_h(s,(2,1)) + \underline{S}_h$	?	...	?
...	...	...	...	...
$\lvert\mathcal{A}_1\rvert$	$-b - \rho^{(R)}_h(s,(\lvert\mathcal{A}_1\rvert,1)) + \underline{S}_h$	?	...	?

$\overline{Q}_h(s)$ is upper bounded by,

$\mathcal{A}_1 \setminus \mathcal{A}_2$	1	2	...	$\lvert\mathcal{A}_2\rvert$
1	$0 + \rho^{(R)}_h(s,(1,1)) + \overline{S}_h$	$b + \rho^{(R)}_h(s,(1,2)) + \overline{S}_h$	...	$b + \rho^{(R)}_h(s,(1,\lvert\mathcal{A}_2\rvert)) + \overline{S}_h$
2	$-b + \rho^{(R)}_h(s,(2,1)) + \overline{S}_h$	?	...	?
...	...	...	...	...
$\lvert\mathcal{A}_1\rvert$	$-b + \rho^{(R)}_h(s,(\lvert\mathcal{A}_1\rvert,1)) + \overline{S}_h$	?	...	?

For $(1,1)$ to be the ι strict, thus unique, Nash equilibrium for all $Q \in \left[\underline{Q}, \overline{Q}\right]$, sufficient conditions are, for $a_1 \neq 1$ and $a_2 \neq 1$,

$$-\rho^{(R)}_h(s,(1,1)) + \underline{S}_h - \frac{\iota}{2} \geqslant -\frac{b}{2H}(H-h+1) \geqslant -b + \rho^{(R)}_h(s,(a_1,1)) + \overline{S}_h + \frac{\iota}{2},$$
$$\rho^{(R)}_h(s,(1,1)) + \overline{S}_h + \frac{\iota}{2} \leqslant \frac{b}{2H}(H-h+1) \leqslant b - \rho^{(R)}_h(s,(1,a_2)) + \underline{S}_h - \frac{\iota}{2},$$

which would be true in period 1 if the following is satisfied for $\mathbf{a}$ such that either $a_1 = \pi^{\dagger}_{1,h}(s)$ or $a_2 = \pi^{\dagger}_{2,h}(s)$,

$$\rho^{(R)}_h(s,\mathbf{a}) \leqslant \frac{b-\iota}{4H} \leqslant \frac{b}{2H} - \frac{\iota}{2},$$

which in turn implies,

$$\underline{S}_h \geqslant -\frac{b}{2H}(H-h+1)+\frac{b}{4H},$$
$$\overline{S}_h \leqslant \frac{b}{2H}(H-h+1)-\frac{b}{4H}.$$

We provide the formal proof below.

Proof. We assume is satisfied, meaning, for each $h \in [H], s \in \mathcal{S}, \mathbf{a} \in \mathcal{A}$,

$$\rho_h^{(R)}(s,\mathbf{a}) \leqslant \frac{b-\iota}{4H} \leqslant \frac{b}{2H}-\frac{\iota}{2}. \tag{C.27}$$

In addition, take $R \in \mathcal{C}^{(R)}$, based on (C.26), we can compute $\hat{R}$ using (C.12), and for each $h \in [H], s \in \mathcal{S}$,

$$-\rho_h^{(R)}\left(s,\pi_h^\dagger(s)\right) \leqslant R_h\left(s,\pi_h^\dagger(s)\right) \leqslant \rho_h^{(R)}\left(s,\pi_h^\dagger(s)\right), \tag{C.28}$$

$$\begin{aligned} -b-\rho_h^{(R)}\left(s,\left(a_1,\pi_{2,h}^\dagger(s)\right)\right) &\leqslant R_h\left(s,\left(a_1,\pi_{2,h}^\dagger(s)\right)\right) \\ &\leqslant -b+\rho_h^{(R)}\left(s,\left(a_1,\pi_{2,h}^\dagger(s)\right)\right), \end{aligned} \tag{C.29}$$

$$\begin{aligned} b-\rho_h^{(R)}\left(s,\left(\pi_{1,h}^\dagger(s),a_2\right)\right) &\leqslant R_h\left(s,\left(\pi_{1,h}^\dagger(s),a_2\right)\right) \\ &\leqslant b+\rho_h^{(R)}\left(s,\left(\pi_{1,h}^\dagger(s),a_2\right)\right). \end{aligned} \tag{C.30}$$

We proceed by induction. In period H, for $a_1 \neq \pi^{\dagger}_{1,H}(s)$,

$$\begin{aligned} Q^{\pi^{\dagger}}_{H}\left(s, \pi^{\dagger}_{H}(s)\right) - \frac{\iota}{2} &= R_H\left(s, \pi^{\dagger}_{H}(s)\right) - \frac{\iota}{2} \\ &\overset{(C.28)}{\geqslant} -\rho^{(R)}_{h}\left(s, \pi^{\dagger}_{H}(s)\right) - \frac{\iota}{2} \\ &\overset{(C.27)}{\geqslant} -\frac{b}{2H} \\ &\geqslant -b + \frac{b}{2H} \\ &\overset{(C.27)}{\geqslant} -b + \rho^{(R)}_{h}\left(s, \left(a_1, \pi^{\dagger}_{2,H}(s)\right)\right) + \frac{\iota}{2} \\ &\overset{(C.29)}{\geqslant} R_H\left(s, \left(a_1, \pi^{\dagger}_{2,H}(s)\right)\right) + \frac{\iota}{2} \\ &= Q^{\pi^{\dagger}}_{h}\left(s, \left(a_1, \pi^{\dagger}_{2,H}(s)\right)\right) + \frac{\iota}{2}, \end{aligned} \tag{C.31}$$

and for $a_2 \neq \pi^{\dagger}_{2,H}(s)$,

$$\begin{aligned} Q^{\pi^{\dagger}}_{H}\left(s, \pi^{\dagger}_{H}(s)\right) + \frac{\iota}{2} &= R_H\left(s, \pi^{\dagger}_{H}(s)\right) + \frac{\iota}{2} \\ &\overset{(C.28)}{\leqslant} \rho^{(R)}_{h}\left(s, \pi^{\dagger}_{H}(s)\right) + \frac{\iota}{2} \\ &\overset{(C.27)}{\leqslant} \frac{b}{2H} \\ &\leqslant b - \frac{b}{2H} \\ &\overset{(C.27)}{\leqslant} b - \rho^{(R)}_{h}\left(s, \left(a_1, \pi^{\dagger}_{2,H}(s)\right)\right) - \frac{\iota}{2} \\ &\overset{(C.29)}{\leqslant} R_H\left(s, \left(\pi^{\dagger}_{1,H}(s), a_2\right)\right) - \frac{\iota}{2} \\ &= Q^{\pi^{\dagger}}_{H}\left(s, \left(\pi^{\dagger}_{1,H}(s), a_2\right)\right) - \frac{\iota}{2}. \end{aligned} \tag{C.32}$$

Now, fix a period $h < H$, we assume in periods $h' \in \{h+1, h+2, ..., H\}$,

in every state $s \in \mathcal{S}$, $\pi^\dagger$ is the Nash equilibrium, and,

$$-\frac{b}{2}\left(H-h'+1\right) \leqslant Q_{h'}^{\pi^\dagger}\left(s, \pi_{h'}^\dagger\left(s\right)\right) \leqslant \frac{b}{2}\left(H-h'+1\right). \tag{C.33}$$

This is true in period H due to (C.31) and (C.32).
Now in period h, for a fixed $s \in \mathcal{S}$, for any $a_1 \neq \pi_{1,h}^\dagger\left(s\right)$,

$$\begin{aligned}
&Q_h^{\pi^\dagger}\left(s, \pi_h^\dagger\left(s\right)\right) - \frac{\iota}{2} \\
&\quad = R_h\left(s, \pi_h^\dagger\left(s\right)\right) + \sum_{s' \in \mathcal{S}} P_h\left(s' | s, \pi_h^\dagger\left(s\right)\right) Q_{h+1}^{\pi^\dagger}\left(s', \pi_{h+1}^\dagger\left(s'\right)\right) - \frac{\iota}{2} \\
&\quad \geqslant R_h\left(s, \pi_h^\dagger\left(s\right)\right) + \min_{s' \in \mathcal{S}} Q_{h+1}^{\pi^\dagger}\left(s', \pi_{h+1}^\dagger\left(s'\right)\right) - \frac{\iota}{2} \\
&\quad \overset{(C.33)}{\geqslant} R_h\left(s, \pi_h^\dagger\left(s\right)\right) - \frac{b}{2}\left(H-h\right) - \frac{\iota}{2} \\
&\quad \overset{(C.28)}{\geqslant} -\rho_h^{(R)}\left(s, \pi_h^\dagger\left(s\right)\right) - \frac{b}{2H}\left(H-h\right) - \frac{\iota}{2} \\
&\quad \overset{(C.27)}{\geqslant} -\frac{b}{2H} - \frac{b}{2H}\left(H-h\right) \\
&\quad \geqslant -\frac{b}{2H}\left(H-h+1\right) \qquad\qquad (C.34) \\
&\quad \geqslant -b + \frac{b}{2H} + \frac{b}{2H}\left(H-h\right) \\
&\quad \overset{(C.27)}{\geqslant} -b + \rho_h^{(R)}\left(s, \left(a_1, \pi_{2,h}^\dagger\left(s\right)\right)\right) + \frac{b}{2H}\left(H-h\right) + \frac{\iota}{2} \\
&\quad \overset{(C.29)}{\geqslant} R_h\left(s, \left(a_1, \pi_{2,h}^\dagger\left(s\right)\right)\right) + \frac{b}{2H}\left(H-h\right) + \frac{\iota}{2} \\
&\quad \overset{(C.33)}{\geqslant} R_h\left(s, \left(a_1, \pi_{2,h}^\dagger\left(s\right)\right)\right) + \max_{s' \in \mathcal{S}} Q_{h+1}^{\pi^\dagger}\left(s', \left(a_1, \pi_{h+1}^\dagger\left(s'\right)\right)\right) + \frac{\iota}{2} \\
&\quad \geqslant R_h\left(s, \left(a_1, \pi_{2,h}^\dagger\left(s\right)\right)\right) + \sum_{s' \in \mathcal{S}} P_h\left(s' | s, \left(a_1, \pi_{2,h}^\dagger\left(s\right)\right)\right) Q_{h+1}^{\pi^\dagger}\left(s', \left(a_1, \pi_{h+1}^\dagger\left(s'\right)\right)\right) + \frac{\iota}{2} \\
&\quad = Q_h^{\pi^\dagger}\left(s, \left(a_1, \pi_{2,h}^\dagger\left(s\right)\right)\right) + \frac{\iota}{2},
\end{aligned}$$

and for $a_2 \neq \pi^\dagger_{2,h}(s)$,

$$
\begin{aligned}
&Q^{\pi^\dagger}_h\left(s, \pi^\dagger_h(s)\right) + \frac{\iota}{2} \\
&\quad = R_h\left(s, \pi^\dagger_h(s)\right) + \sum_{s' \in \mathcal{S}} P_h\left(s' | s, \pi^\dagger_h(s)\right) Q^{\pi^\dagger}_{h+1}\left(s', \pi^\dagger_{h+1}(s')\right) + \frac{\iota}{2} \\
&\quad \leqslant R_h\left(s, \pi^\dagger_h(s)\right) + \max_{s' \in \mathcal{S}} Q^{\pi^\dagger}_{h+1}\left(s', \pi^\dagger_{h+1}(s')\right) + \frac{\iota}{2} \\
&\quad \overset{(C.33)}{\leqslant} R_h\left(s, \pi^\dagger_h(s)\right) + \frac{b}{2H}(H-h) + \frac{\iota}{2} \\
&\quad \overset{(C.28)}{\leqslant} \rho^{(R)}_h\left(s, \pi^\dagger_h(s)\right) + \frac{b}{2H}(H-H) + \frac{\iota}{2} \\
&\quad \overset{(C.27)}{\leqslant} \frac{b}{2H} + \frac{b}{2H}(H-h) \\
&\quad = \frac{b}{2H}(H-h+1) \qquad \text{(C.35)} \\
&\quad \leqslant b - \frac{b}{2H} - \frac{b}{2H}(H-h) \\
&\quad \overset{(C.27)}{\leqslant} b + \rho^{(R)}_h\left(s, \left(a_1, \pi^\dagger_{2,h}(s)\right)\right) - \frac{b}{2H}(H-h) - \frac{\iota}{2} \\
&\quad \overset{(C.29)}{\leqslant} R_h\left(s, \left(\pi^\dagger_{1,h}(s), a_2\right)\right) - \frac{b}{2H}(H-h) - \frac{\iota}{2} \\
&\quad \overset{(C.33)}{\leqslant} R_h\left(s, \left(\pi^\dagger_{1,h}(s), a_2\right)\right) + \min_{s' \in \mathcal{S}} Q^{\pi^\dagger}_{h+1}\left(s', \left(\pi^\dagger_{1,h}(s), a_2\right)\right) - \frac{\iota}{2} \\
&\quad \leqslant R_h\left(s, \left(\pi^\dagger_{1,h}(s), a_2\right)\right) + \sum_{s' \in \mathcal{S}} P_h\left(s' | s, \left(\pi^\dagger_{1,h}(s), a_2\right)\right) Q^{\pi^\dagger}_{h+1}\left(s', \left(\pi^\dagger_{1,h}(s), a_2\right)\right) - \frac{\iota}{2} \\
&\quad = Q^{\pi^\dagger}_h\left(s, \left(\pi^\dagger_{1,h}(s), a_2\right)\right) - \frac{\iota}{2}.
\end{aligned}
$$

Therefore, $\pi^\dagger$ is the Nash equilibrium in period h state s, and (C.34) and (C.35) are consistent (C.33). By induction, $\pi^\dagger$ is a strict, thus unique, Nash equilibrium in every stage game, making $\pi^\dagger$ the unique MPE.

□

Code Details

We conducted our experiments using standard python3 libraries. The only exception being we used the gurobi LP solver. We provide our code in a jupyter notebook with an associated database file so that our experiments can be easily reproduced. The notebook already reads in the database by default so no file management is needed. Simply ensure the notebook is in the same directory as the database folder.

D PLANNING SETTING, REWARD POISONING FOR ZERO-SUM GAMES TO INSTALL A MIXED-STRATEGY NASH EQUILIBRIUM

D.1 Appendix

In this appendix we provide omitted proofs and additional experiments.

Proof of Theorem 6.1

Proof. Theorem 6.1 states that the SIISOW and INV conditions are sufficient and necessary for $(\mathbf{p}, \mathbf{q})$ to be the unique NE of the game R. We prove sufficiency and necessity separately.

Conditions $\Rightarrow$ unique NE: We have already argued that $(\mathbf{p}, \mathbf{q})$ is an NE; see the discussion after the definition of SIISOW. Suppose $(\mathbf{r}, \mathbf{s})$ is another NE. We show that it must be the case $\mathbf{r} = \mathbf{p}, \mathbf{s} = \mathbf{q}$.

First of all, it is easy to see that $supp(\mathbf{r}) \subseteq \mathcal{I}, supp(\mathbf{s}) \subseteq \mathcal{J}$. Suppose there is a violation $\exists i \in supp(\mathbf{r}), i \notin \mathcal{I}$. By (6.6), $\mathbf{e}_i^\top R\mathbf{q} < \mathbf{p}^\top R\mathbf{q} = v^*$ which leads to $\mathbf{r}^\top R\mathbf{q} < v^*$. But since $(\mathbf{r}, \mathbf{s})$ is another NE in a two-player zero-sum game, $(\mathbf{r}, \mathbf{q})$ is a third NE with $\mathbf{r}^\top R\mathbf{q} = v^*$, a contradiction. The case for $\mathbf{s}$ is similar.

Because $(\mathbf{r}, \mathbf{s})$ is an NE, it satisfies the primal-dual LP in Definition 6.4. Now with the support constraints, they satisfy the reduced LPs where the vectors and matrices are restricted to the appropriate support:

$$\max_{\mathbf{r'}_{\mathcal{I}} \in \Delta_{\mathcal{I}}, v} v \qquad \text{s.t. } \mathbf{r'}_{\mathcal{I}}^\top R_{\mathcal{I}\cdot} \geqslant v 1_{|\mathcal{J}|}^\top \tag{D.1}$$

$$\min_{\mathbf{s'}_{\mathcal{J}} \in \Delta_{\mathcal{J}}, v} v \qquad \text{s.t. } R_{\cdot\mathcal{J}} \mathbf{s'}_{\mathcal{J}} \leqslant v 1_{|\mathcal{I}|}. \tag{D.2}$$

We now show this must mean $\mathbf{s} = \mathbf{q}$. Consider two cases on the dual restricted LP:

(Case 1) At the solution $(\mathbf{s}, v^*)$, all constraints in $R_{\mathcal{J}\mathcal{J}}\mathbf{s}_{\mathcal{J}} \leqslant v^*$ are active, i.e. they are equalities $R_{\mathcal{J}\mathcal{J}}\mathbf{s}_{\mathcal{J}} = v^*$. Also $\mathbf{s}_{\mathcal{J}}$ sums to 1. We may write the two as a linear system:

$$\begin{bmatrix} R_{\mathcal{J}\mathcal{J}} & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{s}_{\mathcal{J}} \\ v^* \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{D.3}$$

By the invertability condition, $\mathbf{s}_{\mathcal{J}}$ has a unique solution and it must equal $\mathbf{q}_{\mathcal{J}}$ because $\mathbf{q}_{\mathcal{J}}$ is also a solution to this linear system. The rest of $\mathbf{s}$ and $\mathbf{q}$ are both zeros. Thus $\mathbf{s} = \mathbf{q}$.

(Case 2) At least one constraint in $R_{\mathcal{J}\mathcal{J}}\mathbf{s}_{\mathcal{J}} \leqslant v^*$ is inactive. Then there exists slack variables $\xi \in \mathbb{R}^{|\mathcal{J}|}$, $\xi \geqslant 0$ with at least one positive entry, such that

$$R_{\mathcal{J}\mathcal{J}}\mathbf{s}_{\mathcal{J}} = v^* 1 - \xi.$$

Recall $(\mathbf{p}, \mathbf{q})$ is an NE. By the assumption that $(\mathbf{r}, \mathbf{s})$ is an NE, and the property of two-player zero-sum games, $(\mathbf{p}, \mathbf{s})$ is also an NE with the same value v^*. But $\mathbf{p}^\top R\mathbf{s} = \mathbf{p}_{\mathcal{J}}^\top R_{\mathcal{J}\mathcal{J}}\mathbf{s}_{\mathcal{J}} = v^* - \mathbf{p}_{\mathcal{J}}^\top \xi < v^*$, because all terms in $\mathbf{p}_{\mathcal{J}}$ are positive and at least one term in ξ is positive. This is a contradiction. So case 2 will not happen.

Taken together, $\mathbf{s} = \mathbf{q}$. Similarly, one can show $\mathbf{r} = \mathbf{p}$.

Unique NE $\Rightarrow$ conditions: Let $(\mathbf{p}, \mathbf{q})$ be the unique NE of R with value v^*, and let $\mathcal{I}, \mathcal{J}$ be their support.

We first show SIISOW. Equations (6.4) and (6.5) are immediate from NE definition. Since $(\mathbf{p}, \mathbf{q})$ is the only NE of the game, it satisfies Goldman and Tucker Corollary 3A. The corollary states that

$$\forall i \in \mathcal{A}_1, (\mathbf{e}_i^\top R\mathbf{q} = v^*) \Rightarrow (i \in \mathcal{I}) \tag{D.4}$$

$$\forall j \in \mathcal{A}_2, (\mathbf{p}^\top R\mathbf{e}_j = v^*) \Rightarrow (j \in \mathcal{J}). \tag{D.5}$$

Their contraposition is

$$\forall i \in \mathcal{A}_1, (i \notin \mathcal{I}) \Rightarrow (\mathbf{e}_i^\top R\mathbf{q} \neq v^*) \tag{D.6}$$

$$\forall j \in \mathcal{A}_2, (j \notin \mathcal{J}) \Rightarrow (\mathbf{p}^\top R\mathbf{e}_j \neq v^*). \tag{D.7}$$

But since v^* is the NE game value, these imply

$$\forall i \in \mathcal{A}_1, (i \notin \mathcal{I}) \Rightarrow (\mathbf{e}_i^\top R\mathbf{q} < v^*) \tag{D.8}$$

$$\forall j \in \mathcal{A}_2, (j \notin \mathcal{J}) \Rightarrow (\mathbf{p}^\top R\mathbf{e}_j < v^*). \tag{D.9}$$

Therefore, $(\mathbf{p}, \mathbf{q})$ satisfies the SIISOW condition.

We next show invertability by contradition. Suppose the matrix in Definition 6.2 is not invertable. Then either (i) $|\mathcal{I}| < |\mathcal{J}|$, (ii) $|\mathcal{I}| > |\mathcal{J}|$, or (iii) $|\mathcal{I}| = |\mathcal{J}| \geqslant 2$. Case (iii) is due to the fact that should $|\mathcal{I}| = |\mathcal{J}| = 1$, $R_{\mathcal{I}\mathcal{J}}$ is a scalar and the matrix $\begin{bmatrix} R_{\mathcal{I}\mathcal{J}} & -1 \\ 1 & 0 \end{bmatrix}$ with determinant 1 is always invertible. We show that any one of the three cases leads to a second NE, contradicting the uniqueness of $(\mathbf{p}, \mathbf{q})$. In what follows we give the proof for (i) or (iii); case (ii) is similar to (i) but with respect to $R_{\mathcal{I}\mathcal{J}}^\top$ and $\mathbf{p}$, and is omitted.

In cases (i) or (iii) the following homogeneous linear system has a nonzero solution:

$$\begin{bmatrix} R_{\mathcal{I}\mathcal{J}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix} \begin{bmatrix} \delta \\ x \end{bmatrix} = 0, \tag{D.10}$$

where $\delta \in \mathbb{R}^{|\mathcal{J}|}, x \in \mathbb{R}$. This nonzero solution (δ, x) has some useful properties:

- δ sums to zero:

$$1^\top \delta = 0. \tag{D.11}$$

  This follows directly from the second equality of (D.10).

- $\delta \neq 0$. This follows from the first equality of (D.10)

$$R_{\mathcal{J}\mathcal{J}}\delta = x1, \tag{D.12}$$

otherwise both δ and x would be zero, contradicting a nonzero solution.

- $x = 0$ and

$$R_{\mathcal{J}\mathcal{J}}\delta = 0. \tag{D.13}$$

We first show $x = 0$. Consider

$$\begin{aligned} & \mathbf{p}^\top R \begin{bmatrix} \delta \\ 0_{|\mathcal{A}_2|-|\mathcal{J}|} \end{bmatrix} && \text{(D.14)} \\ = & \sum_{j\in\mathcal{J}} \mathbf{p}^\top R\mathbf{e}_j\delta_j && \text{(D.15)} \\ = & \sum_{j\in\mathcal{J}} \nu^*\delta_j && \text{(D.16)} \\ = & \ 0, && \text{(D.17)} \end{aligned}$$

where the second equality follows from the SIISOW condition $\mathbf{p}^\top R\mathbf{e}_j = \mathbf{p}^\top R\mathbf{q} = \nu^*,\ \forall j \in \mathcal{J}$. But at the same time, by the support of $\mathbf{p}$

$$\begin{aligned} & \mathbf{p}^\top R \begin{bmatrix} \delta \\ 0_{|\mathcal{A}_2|-|\mathcal{J}|} \end{bmatrix} && \text{(D.18)} \\ = & \ \mathbf{p}_{\mathcal{J}}^\top R_{\mathcal{J}\mathcal{J}}\delta && \text{(D.19)} \\ = & \ \mathbf{p}_{\mathcal{J}}^\top x1 = x. && \text{(D.20)} \end{aligned}$$

Therefore $x = 0$. Then use (D.12) to obtain (D.13).

We use this δ to construct another NE with the following steps:

1. We scale δ so its magnitute is sufficiently small. The desired scale is determined by two constants:

a) Since we are under cases (i) or (iii), $|\mathcal{J}| \geqslant 2$. Thus the entries of $\mathbf{q}_{\mathcal{J}}$ cannot be 0 or 1: $\exists c_1 > 0 : c_1 \leqslant q_j \leqslant 1 - c_1, \forall j \in \mathcal{J}$.

b) By the SIISOW condition, $\mathbf{e}_i^\top R\mathbf{q} < v^*$ for $i \notin \mathcal{I}$. Let $c_2 = v^* - \max_{i \notin \mathcal{I}} \mathbf{e}_i^\top R\mathbf{q}$.

We choose the scale

$$c = \min\left(\frac{c_1}{\|\delta\|_\infty}, \min_{i \notin I} \frac{c_2}{|R_{i\mathcal{J}}\delta|}\right). \tag{D.21}$$

2. Set $\mathbf{r} = \mathbf{q} + \begin{bmatrix} c\delta \\ 0 \end{bmatrix}$.

We claim $(\mathbf{p}, \mathbf{r})$ is another NE:

- Since δ sums to zero, $\mathbf{q}_{\mathcal{J}} + c\delta$ remains normalized; since $c \leqslant \frac{c_1}{\|\delta\|_\infty}$, all entries of $\mathbf{q}_{\mathcal{J}} + c\delta$ remains in $[0, 1]$. Therefore $\mathbf{r} \in \Delta_{\mathcal{A}_2}$ is a proper strategy.

- $\mathbf{r}$ is a best response to $\mathbf{p}$:

$$\mathbf{p}^\top R\mathbf{r} = \mathbf{p}^\top R\mathbf{q} + \mathbf{p}^\top R \begin{bmatrix} c\delta \\ 0 \end{bmatrix} = v^*, \tag{D.22}$$

where we used (D.14). Therefore, $\mathbf{p}^\top R\mathbf{r} = v^* \leqslant \mathbf{p}^\top R\mathbf{q}', \forall \mathbf{q}' \in \Delta_{\mathcal{A}_2}$ because $\mathbf{p}$ is part of an NE.

- $\mathbf{p}$ is a best response to $\mathbf{r}$: $\forall \mathbf{p}' \in \Delta_{\mathcal{A}_1}$,

$$
\begin{aligned}
& \mathbf{p}'^{\top} R \mathbf{r} && \text{(D.23)} \\
= & \sum_{i \in \mathcal{I}} p'_i \mathbf{e}_i^{\top} R \mathbf{q} + \mathbf{p}'^{\top}_{\mathcal{J}} R_{\mathcal{J}\mathcal{J}} c\delta + \sum_{i \notin \mathcal{I}} p'_i \left(\mathbf{e}_i^{\top} R \mathbf{q} + \mathbf{e}_i^{\top} R \begin{bmatrix} c\delta \\ 0 \end{bmatrix} \right) \\
= & \sum_{i \in \mathcal{I}} p'_i \mathbf{e}_i^{\top} R \mathbf{q} + \sum_{i \notin \mathcal{I}} p'_i \left(\mathbf{e}_i^{\top} R \mathbf{q} + \mathbf{e}_i^{\top} R \begin{bmatrix} c\delta \\ 0 \end{bmatrix} \right) \\
= & \sum_{i \in \mathcal{I}} p'_i v^* + \sum_{i \notin \mathcal{I}} p'_i \left(\mathbf{e}_i^{\top} R \mathbf{q} + \mathbf{e}_i^{\top} R \begin{bmatrix} c\delta \\ 0 \end{bmatrix} \right) \\
\leqslant & \sum_{i \in \mathcal{I}} p'_i v^* + \sum_{i \notin \mathcal{I}} p'_i \left(v^* - c_2 + \mathbf{e}_i^{\top} R \begin{bmatrix} c\delta \\ 0 \end{bmatrix} \right) \\
= & \sum_{i \in \mathcal{I}} p'_i v^* + \sum_{i \notin \mathcal{I}} p'_i \left(v^* - c_2 + c R_{i\mathcal{J}} \delta \right). && \text{(D.24)}
\end{aligned}
$$

  where the second equality follows from (D.13), the next two lines from SIISOW. Because $c \leqslant \min_{i \notin I} \frac{c_2}{|R_{i\mathcal{J}}\delta|}$,

$$
\begin{aligned}
& \mathbf{p}'^{\top} R \mathbf{r} && \text{(D.25)} \\
\leqslant & \sum_{i \in \mathcal{I}} p'_i v^* + \sum_{i \notin \mathcal{I}} p'_i \left(v^* - c_2 + c_2 \right) = v^* = \mathbf{p}^{\top} R \mathbf{r}.
\end{aligned}
$$

Because $\delta \neq 0$, $\mathbf{r} \neq \mathbf{q}$. Thus $(\mathbf{p}, \mathbf{r}) \neq (\mathbf{p}, \mathbf{q})$ is indeed a second NE, contradicting uniqueness.

□

Proof of Lemma 6.1

For ease of understanding, in Table D.1 we illusrate the reward matrix $R^{\text{eRPS}(\mathbf{p},\mathbf{q})}$ for the extended Rock-Paper-Scissors game when $k \geqslant 2$.

$\mathcal{A}_1 \setminus \mathcal{A}_2$	0	1	2	3	...	$k-2$	$k-1$	k	...	$\lvert\mathcal{A}_2\rvert-1$
0	0	$-\frac{c}{p_0 q_1}$	$\frac{c}{p_0 q_2}$	0	...	0	0	1	...	1
1	0	0	$-\frac{c}{p_1 q_2}$	$\frac{c}{p_1 q_3}$	...	0	0	1	...	1
2	0	0	0	$-\frac{c}{p_2 q_3}$	...	0	0	1	...	1
3	0	0	0	0	...	0	0	1	...	1
...	...	...	...	...	...	...	...	...	...	...
$k-2$	$\frac{c}{p_{k-2} q_0}$	0	0	0	...	0	$-\frac{c}{p_{k-2} q_{k-1}}$	1	...	1
$k-1$	$-\frac{c}{p_{k-1} q_0}$	$\frac{c}{p_{k-1} q_1}$	0	0	...	0	0	1	...	1
k	-1	-1	-1	-1	...	-1	-1	0	...	0
...	...	...	...	...	...	...	...	...	...	...
$\lvert\mathcal{A}_1\rvert-1$	-1	-1	-1	-1	...	-1	-1	0	...	0

Table D.1: The R^{eRPS} game when $k \geqslant 2$, i.e. $(\mathbf{p}, \mathbf{q})$ is a mixed strategy

Proof. To show uniqueness, we check that the conditions in Theorem 6.1 is satisfied,

$$\begin{aligned}
&\mathbf{e}_i^\top R\mathbf{q} = 0 = \mathbf{p}^\top R\mathbf{q}, \forall\, i \in \mathcal{I},\\
&\mathbf{e}_i^\top R\mathbf{q} = -1 < 0 = \mathbf{p}^\top R\mathbf{q}, \forall\, i \notin \mathcal{I},\\
&\mathbf{p}^\top R\mathbf{e}_j = 0 = \mathbf{p}^\top R\mathbf{q}, \forall\, j \in \mathcal{J},\\
&\mathbf{p}^\top R\mathbf{e}_j = 1 > 0 = \mathbf{p}^\top R\mathbf{q}, \forall\, j \notin \mathcal{J},
\end{aligned} \tag{D.26}$$

and we have $\begin{bmatrix} R_{\mathcal{I}\mathcal{J}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix}$ is invertible.

To simplify the notations, we omit the modulo k operation for the indices of $\mathbf{p}$ and $\mathbf{q}$. Observe that

$$\begin{aligned}
&\mathbf{e}_i^\top R\mathbf{q} = -\frac{c}{p_i q_{i+1}}\mathbf{q}_{i+1} + \frac{c}{p_i q_{i+2}}\mathbf{q}_{i+2} = 0, \forall\, i \in \mathcal{I},\\
&\mathbf{e}_i^\top R\mathbf{q} = \sum_{j \in \mathcal{J}} -1\mathbf{q}_j = -1, \forall\, i \notin \mathcal{I},
\end{aligned} \tag{D.27}$$

and similarly,

$$\begin{aligned}\mathbf{p}^\top R\mathbf{e}_j &= \frac{c}{\mathbf{p}_{j-2}\mathbf{q}_j}\mathbf{p}_{j-2} - \frac{c}{\mathbf{p}_{j-1}q_j}\mathbf{p}_{j-1} = 0, \forall\, j \in \mathcal{J},\\ \mathbf{p}^\top R\mathbf{e}_j &= \sum_{i\in\mathcal{J}} 1\mathbf{p}_i = 1, \forall\, j \notin \mathcal{J}.\end{aligned} \tag{D.28}$$

In addition, we have,

$$\mathbf{p}^\top R\mathbf{q} = \sum_{i\in\mathcal{J}} \mathbf{p}_i \left(\mathbf{e}_i^\top R\mathbf{q}\right) = 0. \tag{D.29}$$

Therefore, the SIISOW conditions are satisfied.

We now turn to the invertibility condition. For $k = 1$, $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ is invertible. For fixed $\mathbf{p}, \mathbf{q}$, for $k = 2$, we have,

$$\begin{aligned}&\det \begin{bmatrix} \frac{c}{\mathbf{p}_0\mathbf{q}_0} & -\frac{c}{\mathbf{p}_0\mathbf{q}_1} & -1 \\ -\frac{c}{\mathbf{p}_1\mathbf{q}_0} & \frac{c}{\mathbf{p}_1\mathbf{q}_1} & -1 \\ 1 & 1 & 0 \end{bmatrix}\\ &= \det \begin{bmatrix} \frac{1}{\mathbf{p}_0} & 0 & 0 \\ 0 & \frac{1}{\mathbf{p}_1} & 0 \\ 0 & 0 & \frac{1}{c} \end{bmatrix} \det \begin{bmatrix} 1 & -1 & \mathbf{p}_0 \\ -1 & 1 & \mathbf{p}_1 \\ \mathbf{q}_0 & \mathbf{q}_1 & 0 \end{bmatrix} \det \begin{bmatrix} \frac{1}{\mathbf{q}_0} & 0 & 0 \\ 0 & \frac{1}{\mathbf{q}_1} & 0 \\ 0 & 0 & \frac{1}{c} \end{bmatrix}\\ &= c\left(\mathbf{p}_0 + \mathbf{p}_1\right)\frac{\mathbf{q}_0 + \mathbf{q}_1}{\mathbf{p}_0\mathbf{p}_1\mathbf{q}_0\mathbf{q}_1}\\ &> 0,\end{aligned} \tag{D.30}$$

therefore it is invertible, similarly for $k = 3$,

$$
\begin{aligned}
&\det \begin{bmatrix} 0 & -\frac{c}{p_0 q_1} & \frac{c}{p_0 q_2} & -1 \\ \frac{c}{p_1 q_0} & 0 & -\frac{c}{p_1 q_2} & -1 \\ -\frac{c}{p_2 q_0} & \frac{c}{p_2 q_1} & 0 & -1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \\
&= \det \begin{bmatrix} \frac{1}{p_0} & 0 & 0 & 0 \\ 0 & \frac{1}{p_1} & 0 & 0 \\ 0 & 0 & \frac{1}{p_2} & 0 \\ 0 & 0 & 0 & \frac{1}{c} \end{bmatrix} \det \begin{bmatrix} 0 & -1 & 1 & -p_0 \\ 1 & 0 & -1 & -p_1 \\ -1 & 1 & 0 & -p_2 \\ q_0 & q_1 & q_2 & 0 \end{bmatrix} \det \begin{bmatrix} \frac{1}{q_0} & 0 & 0 & 0 \\ 0 & \frac{1}{q_1} & 0 & 0 \\ 0 & 0 & \frac{1}{q_2} & 0 \\ 0 & 0 & 0 & \frac{1}{c} \end{bmatrix} \\
&= c^2 \frac{(p_0 + p_1 + p_2)(q_0 + q_1 + q_2)}{p_0 p_1 p_2 q_0 q_1 q_2} \\
&> 0,
\end{aligned}
\tag{D.31}
$$

and for $k = 4$,

$$\det \begin{bmatrix} 0 & -\frac{c}{p_0 q_1} & \frac{c}{p_0 q_2} & 0 & -1 \\ 0 & 0 & -\frac{c}{p_1 q_2} & \frac{c}{p_1 q_3} & -1 \\ \frac{c}{p_2 q_0} & 0 & 0 & -\frac{c}{p_2 q_3} & -1 \\ -\frac{c}{p_3 q_0} & \frac{c}{p_3 q_1} & 0 & 0 & -1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$= \det \begin{bmatrix} \frac{1}{p_0} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{p_1} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{p_2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{p_3} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{c} \end{bmatrix} \det \begin{bmatrix} 0 & -1 & 1 & 0 & -p_0 \\ 0 & 0 & -1 & 1 & -p_1 \\ 1 & 0 & 0 & -1 & -p_2 \\ -1 & 1 & 0 & 0 & -p_3 \\ q_0 & q_1 & q_2 & q_3 & 0 \end{bmatrix} \det \begin{bmatrix} \frac{1}{q_0} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{q_1} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{q_2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{q_3} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{c} \end{bmatrix}$$

$$= c^3 \frac{(p_0 + p_1 + p_2 + p_3)(q_0 + q_1 + q_2 + q_3)}{p_0 p_1 p_2 p_3 q_0 q_1 q_2 q_3}$$

$$> 0, \tag{D.32}$$

and in general, we can write $\begin{bmatrix} R_{\mathcal{I}\mathcal{J}} & -1_{|\mathcal{I}|} \\ 1^\top_{|\mathcal{J}|} & 0 \end{bmatrix}$ as the product of $\operatorname{diag}\left(\frac{1}{p_1}, \frac{1}{p_2}, ..., \frac{1}{p_k}, \frac{1}{c}\right)$, $\begin{bmatrix} R' & \mathbf{p} \\ \mathbf{q}^\top & 0 \end{bmatrix}$, and $\operatorname{diag}\left(\frac{1}{q_1}, \frac{1}{q_2}, ..., \frac{1}{q_k}, \frac{1}{c}\right)$, where R' is a matrix with entries,

$$R'_{ij} = \begin{cases} -1 & \text{if } j = (i+1) \mod k \\ 1 & \text{if } j = (i+2) \mod k \\ 0 & \text{otherwise} \end{cases}, \tag{D.33}$$

with the above examples provided for $k = 2, 3, 4$,

and the determinant is given by,

$$
\begin{aligned}
&\det \begin{bmatrix} R_{\mathcal{I}\mathcal{J}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix} \\
&= \det\ \mathrm{diag}\left(\frac{1}{\mathbf{p}_1}, \frac{1}{\mathbf{p}_2}, ..., \frac{1}{\mathbf{p}_k}, \frac{1}{c}\right) \det \begin{bmatrix} R' & \mathbf{p} \\ \mathbf{q}^\top & 0 \end{bmatrix} \det\ \mathrm{diag}\left(\frac{1}{\mathbf{q}_1}, \frac{1}{\mathbf{q}_2}, ..., \frac{1}{\mathbf{q}_k}, \frac{1}{c}\right) \\
&= c^{k-1} \frac{\sum_{i=1}^{k} \mathbf{p}_i \sum_{j=1}^{k} \mathbf{q}_j}{\prod_{i=1}^{k} \mathbf{p}_i \prod_{j=1}^{k} \mathbf{q}_j} \\
&> 0.
\end{aligned}
\tag{D.34}
$$

This verifies the INV condition and completes the proof. □

The Markov Game Modification Problem as An Optimization Problem

Here we instantiate the general Game Modification problem (Definition 6.1) to Markov games as an optimization problem.

Definition D.1 (Game Modification for Two-Player Zero-Sum Markov Game)**.** *Given the cost function ℓ, the target policy $(\mathbf{p}, \mathbf{q})$ with supports $\mathcal{I}, \mathcal{J}$, target value range $[\underline{v}, \overline{v}]$, the game modification for Markov games can be written*

as the following optimization problem,

$$
\begin{aligned}
&\inf_{R,\nu,\mathbb{Q}} \ \ell\left(R, R^{\circ}\right) \\
&\text{s.t. } \left[\mathbb{Q}_h\left(s\right)\right]_{\mathcal{I}_h(s)\bullet} \mathbf{q}_h\left(s\right) = \nu_h\left(s\right) 1_{|\mathcal{I}_h(s)|} \\
&\qquad \forall\, h \in [H], s \in \mathcal{S} \\
&\quad \mathbf{p}_h^{\top}\left(s\right) \left[\mathbb{Q}_h\left(s\right)\right]_{\bullet \mathcal{J}_h(s)} = \nu_h\left(s\right) 1_{|\mathcal{J}_h(s)|}^{\top} \\
&\qquad \forall\, h \in [H], s \in \mathcal{S} \\
&\quad \left[\mathbb{Q}_h\left(s\right)\right]_{\mathcal{A}_1 \setminus \mathcal{I}_h(s)\bullet} \mathbf{q}_h\left(s\right) < \nu_h\left(s\right) 1_{|\mathcal{A}_1 \setminus \mathcal{I}_h(s)|} \\
&\qquad \forall\, h \in [H], s \in \mathcal{S} \\
&\quad \mathbf{p}_h^{\top}\left(s\right) \left[\mathbb{Q}_h\left(s\right)\right]_{\bullet \mathcal{A}_2 \setminus \mathcal{J}_h(s)} > \nu_h\left(s\right) 1_{|\mathcal{A}_2 \setminus \mathcal{J}_h(s)|}^{\top} \\
&\qquad \forall\, h \in [H], s \in \mathcal{S} \\
&\quad \sigma_{\min}\left(\begin{bmatrix} \left[\mathbb{Q}_h\left(s\right)\right]_{\mathcal{I}_h(s)\mathcal{J}_h(s)} & -1_{|\mathcal{I}_h(s)|} \\ 1_{|\mathcal{J}_h(s)|}^{\top} & 0 \end{bmatrix}\right) > 0 \qquad \text{(D.35)} \\
&\qquad \forall\, h \in [H], s \in \mathcal{S} \\
&\quad \mathbb{Q}_h\left(s\right) = R_h\left(s\right) + \sum_{s' \in \mathcal{S}} P_h\left(s'|s\right) \nu_{h+1}\left(s'\right) \\
&\qquad \forall\, h \in [H-1], s \in \mathcal{S} \\
&\quad \mathbb{Q}_H\left(s\right) = R_H\left(s\right), \forall\, s \in \mathcal{S} \\
&\quad \underline{\nu} \leqslant \sum_{s \in \mathcal{S}} P_0\left(s\right) \nu_1\left(s\right) \leqslant \overline{\nu} \\
&\quad -b \leqslant \left[R_h\left(s\right)\right]_{ij} \leqslant b \\
&\qquad \forall\ (i,j) \in \mathcal{A}, h \in [H], s \in \mathcal{S}.
\end{aligned}
$$

Proof of Theorem 6.2 and Corollary 6.1

Theorem 6.2 concerns the feasibility of modifying normal-form games in Definition 6.5, and Corollary 6.1 concerns the feasibility of modifying H-period Markov games in Definition D.1. Below we prove Corollary 6.1, from which Theorem 6.2 follows as an special case with $H = 1$.

Direction $\Rightarrow$. If $\pi = (\mathbf{p}, \mathbf{q})$ is the unique Nash in stage game in period $h \in [H]$, state $s \in \mathcal{S}$, then by Theorem 6.1, $\begin{bmatrix} R_{\mathcal{I}_h(s)\mathcal{J}_h(s)} & -1_{\mathcal{I}_h(s)} \\ 1^{\top}_{\mathcal{J}_h(s)} & 0 \end{bmatrix}$ is an invertible square matrix, therefore, $|\mathcal{I}_h(s)| = |\mathcal{J}_h(s)|$.
Now to show that $(-Hb, Hb) \cap [\underline{v}, \overline{v}] =$ empty leads to infeasibility, note that either,

$$\overline{v} \geqslant Hb, \tag{D.36}$$

or,

$$\underline{v} \leqslant -Hb, \tag{D.37}$$

meaning the value of at least one stage game at least b or at most $-b$, and the SIISOW conditions imply that there are some entries of $R_h(s)$ that are strictly larger than b or strictly smaller than $-b$, which contradicts the reward bound conditions.

Direction $\Leftarrow$. Fix a stage game in period $h \in [H]$, state $s \in \mathcal{S}$, if $|\mathcal{I}_h(s)| = |\mathcal{J}_h(s)| = k$ for some k, then without loss of generality, we can rename the actions so that $\mathcal{I}_h(s) = \mathcal{J}_h(s) = \{0, 1, 2, ..., k-1\}$ and Lemma 6.1 provides a game with the unique Nash equilibrium $(\mathbf{p}_h(s), \mathbf{q}_h(s))$. Note that since the value of R^{eRPS} is 0, all stage games have value 0, so we have, for every $h \in [H], s \in \mathcal{S}$,

$$Q_h(s) = R_h(s). \tag{D.38}$$

The $(-Hb, Hb) \cap [\underline{v}, \overline{v}] \neq \emptyset$ condition guarantees the existence of some $v^\star \in [\underline{v}, \overline{v}]$ that satisfies,

$$-Hb < v^\star < Hb. \tag{D.39}$$

Now consider the Markov game (R, P) with rewards defined by,

$$R_h(s) = R^{\text{eRPS}(\mathbf{p}_h(s), \mathbf{q}_h(s))} + \frac{1}{H} v^\star. \tag{D.40}$$

This implies that the Q matrices can be computed as recursively for $h =$

$H-1, H-2, ..., 1,$

$$\begin{aligned} v_h(s) &= \frac{H-h+1}{H} v^\star, \\ Q_h(s) &= R_h(s) + \sum_{s' \in \mathcal{S}} P_h(s'|s)\, v_{h+1}(s) \\ &= R_h(s) + \sum_{s' \in \mathcal{S}} P_h(s'|s) \frac{H-h}{H} v^\star \\ &= R_h(s) + \frac{H-h}{H} v^\star \\ &= R^{\text{eRPS}(\mathbf{p}_h(s), \mathbf{q}_h(s))} + \frac{H-h+1}{H} v^\star, \end{aligned} \tag{D.41}$$

which is an affine transformation of R^{eRPS}, so it has unique Nash $(\mathbf{p}_h(s), \mathbf{q}_h(s))$ with value $\frac{H-h+1}{H} v^\star$. In particular, the value of this game is given by,

$$\begin{aligned} v_0 &\coloneqq \sum_{s \in \mathcal{S}} P_0(s)\, v_1(s) \\ &= \sum_{s \in \mathcal{S}} P_0(s) \frac{H-1+1}{H} v^\star \\ &= v^\star, \end{aligned} \tag{D.42}$$

which satisfies the value range constraint.

Proof of Feasibility/Optimality for RAP and RAP-MG Algorithms (Proposition 6.1 and Corollary D.1)

Proposition 6.1 concerns the feasibility and optimality of the RAP algorithm for normal form games. This result is a special case of Corollary D.1 below for the RAP-MG algorithm for Markov games.

Corollary D.1 (Feasibility and Optimality of the RAP-MG Algorithm)**.**

The output $R(\iota, \lambda) = R' + \varepsilon R^{\text{eRPS}}$ *of Algorithm 5 with parameters* ι, λ *satisfying,*

$$\lambda + \iota < \frac{1}{H} \min\{Hb + \overline{v}, Hb - \underline{v}\} \tag{D.43}$$

has the following properties,

- (***Existence***) *The solution* R' *to* (6.15) *exists.*
- (***Feasibility***) $R(\iota, \lambda)$ *is feasible for the original game modification problem in Definition 6.1 with probability* 1.
- (***Optimality***) *Under the additional assumption that* ℓ *is Lipschitz with constant* L*,*

  $$|\ell(R, R^\circ) - \ell(R', R^\circ)| \leqslant L \|R - R'\|_1, \tag{D.44}$$

  $R(\iota, \lambda)$ *is near-optimal in the following sense,*

  $$\lim_{\max\{\iota,\lambda\}\to 0} \ell(R(\iota, \lambda), R^\circ) = C^\star, \tag{D.45}$$

  where $C^\star$ *is the optimal objective value in Definition D.1.*

Proof. We show the general result for H-period Markov games, and Theorem 6.1 is the special case when $H = 1$.

Existence. Existence of a solution is implied by Corollary 6.1 with value bounds $[-Hb + H\lambda, Hb - H\lambda]$, and due to (D.43), we have,

$$(-Hb + H\lambda, Hb - H\lambda) \cap [\underline{v}, \overline{v}] \neq \emptyset, \tag{D.46}$$

and therefore, Corollary 6.1 implies the feasible of the problem thus existence of a solution.

Feasibility. We only have to check the INV constraints since $\iota, \lambda > 0$ implies that the other constraints in the original problem are satisfied. We check that for every stage game $\mathbb{Q}$ in period $h \in [H]$, $s \in \mathcal{S}$, we have

$\mathbb{Q}_h(s) = \mathbb{Q}'_h(s) + \varepsilon R^{\text{eRPS}}(\mathbf{p}_h(s), \mathbf{q}_h(s))$ satisfies INV, where $\mathbb{Q}'_h(s)$ is the solution to the optimization. To simplify the notations, we drop the (h, s) indices.

We use the following properties of R^{eRPS} from the proof of Lemma 6.1,

$$\begin{aligned} R^{\text{eRPS}}_{\mathcal{I}\bullet}\mathbf{q} &= 0_{|\mathcal{I}|}, \\ \mathbf{p}^\top R^{\text{eRPS}}_{\bullet\mathcal{J}} &= 0_{|\mathcal{J}|}, \\ R^{\text{eRPS}}_{\mathcal{A}_1\setminus\mathcal{I}\bullet}\mathbf{q} &= -1_{|\mathcal{A}_1\setminus\mathcal{I}|}, \\ \mathbf{p}^\top R^{\text{eRPS}}_{\bullet\mathcal{A}_2\setminus\mathcal{J}} &= 1_{|\mathcal{A}_2\setminus\mathcal{J}|}. \end{aligned} \tag{D.47}$$

Now we check the three conditions of the attacker's problem are satisfied. We have

$$\begin{aligned} \mathbb{Q}_{\mathcal{I}\bullet}\mathbf{q} &= \mathbb{Q}'_{\mathcal{I}}\mathbf{q} + \varepsilon R^{\text{eRPS}}_{\mathcal{I}\bullet}\mathbf{q} \\ &= \nu 1_{|\mathcal{I}|} \\ &= \nu' 1_{|\mathcal{I}|}, \end{aligned} \tag{D.48}$$

and similarly,

$$\begin{aligned} \mathbf{p}^\top \mathbb{Q}_{\bullet\mathcal{J}} &= \mathbf{p}^\top \mathbb{Q}'_{\bullet\mathcal{J}} + \varepsilon \mathbf{p} R^{\text{eRPS}}_{\bullet\mathcal{J}} \\ &= \nu 1_{|\mathcal{J}|} \\ &= \nu' 1_{|\mathcal{J}|}. \end{aligned} \tag{D.49}$$

We also have

$$\begin{aligned} \mathbb{Q}_{\mathcal{A}_1\setminus\mathcal{I}\bullet}\mathbf{q} &= \mathbb{Q}'_{\mathcal{A}_1\setminus\mathcal{I}\bullet}\mathbf{q} + \varepsilon R^{\text{eRPS}}_{\mathcal{A}_1\setminus\mathcal{I}}\mathbf{q} \\ &< \nu 1_{|\mathcal{A}_1\setminus\mathcal{I}|} - \varepsilon 1_{|\mathcal{A}_1\setminus\mathcal{I}|} \\ &< \nu' 1_{|\mathcal{A}_1\setminus\mathcal{I}|}. \end{aligned} \tag{D.50}$$

and similarly,

$$\begin{aligned} \mathbf{p}^\top \mathbb{Q}_{\bullet\mathcal{A}_2\setminus\mathcal{J}} &= \mathbf{p}^\top \mathbb{Q}'_{\bullet\mathcal{A}_2\setminus\mathcal{J}} + \varepsilon R^{\text{eRPS}}_{\mathcal{A}_2\setminus\mathcal{J}}\mathbf{q} \\ &> \nu 1_{|\mathcal{A}_2\setminus\mathcal{J}|} + \varepsilon 1_{|\mathcal{A}_2\setminus\mathcal{J}|} \\ &> \nu' 1_{|\mathcal{A}_2\setminus\mathcal{J}|}. \end{aligned} \tag{D.51}$$

Next we show that $\begin{bmatrix} \mathbb{Q}_{\mathcal{I}\mathcal{J}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix}$ is invertible with probability 1, in particular, since $\begin{bmatrix} R^{\text{eRPS}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix}$ is invertible by Lemma 6.1, we can write its singular value decomposition,

$$\begin{bmatrix} R^{\text{eRPS}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix} = U\Sigma V^\top, \tag{D.52}$$

for some orthonormal $U, V \in \mathbb{R}^{(|\mathcal{I}|+1)\times(|\mathcal{J}|+1)}$ and nonsingular diagonal matrix $\Sigma \in \mathbb{R}^{(|\mathcal{I}|+1)\times(|\mathcal{J}|+1)}$. Consider the event $\begin{bmatrix} \mathbb{Q}_{\mathcal{I}\mathcal{J}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix}$ is singular.

Then the following matrix is also singular:

$$
\begin{aligned}
&\Sigma^{-1/2} U^\top \begin{bmatrix} \mathbb{Q}_{\mathcal{I}\mathcal{J}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix} V \Sigma^{-1/2} \\
&= \Sigma^{-1/2} U^\top \left(\begin{bmatrix} \mathbb{Q}'_{\mathcal{I}\mathcal{J}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix} + \varepsilon \begin{bmatrix} R^{\,\mathrm{eRPS}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix} \right) V \Sigma^{-1/2} \\
&= \Sigma^{-1/2} U^\top \begin{bmatrix} \mathbb{Q}'_{\mathcal{I}\mathcal{J}} + \varepsilon R^{\,\mathrm{eRPS}} & -(1+\varepsilon)\, 1_{|\mathcal{I}|} \\ (1+\varepsilon)\, 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix} V \Sigma^{-1/2} \\
&= \Sigma^{-1/2} U^\top \begin{bmatrix} \mathbb{Q}'_{\mathcal{I}\mathcal{J}} + \dfrac{\varepsilon'}{1-\varepsilon'} R^{\,\mathrm{eRPS}} & -\dfrac{1}{1-\varepsilon'} 1_{|\mathcal{I}|} \\ \dfrac{1}{1-\varepsilon'} 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix} V \Sigma^{-1/2} \\
&\qquad \text{where } \varepsilon' \coloneqq \frac{\varepsilon}{1+\varepsilon} = 1 - \frac{1}{1+\varepsilon}, \qquad (D.53) \\
&\qquad \text{which implies } \varepsilon = \frac{1}{1-\varepsilon'} - 1 = \frac{\varepsilon'}{1-\varepsilon'}, \\
&= \frac{1}{1-\varepsilon'} \Sigma^{-1/2} U^\top \begin{bmatrix} (1-\varepsilon')\, \mathbb{Q}'_{\mathcal{I}\mathcal{J}} + \varepsilon' R^{\,\mathrm{eRPS}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix} V \Sigma^{-1/2} \\
&= \Sigma^{-1/2} U^\top \begin{bmatrix} \mathbb{Q}'_{\mathcal{I}\mathcal{J}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix} V \Sigma^{-1/2} \\
&\qquad + \frac{\varepsilon'}{1-\varepsilon'} \Sigma^{-1/2} U^\top \begin{bmatrix} R^{\mathrm{eRPS}}_{\mathcal{I}\mathcal{J}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix} V \Sigma^{-1/2} \\
&= \Sigma^{-1/2} U^\top \begin{bmatrix} \mathbb{Q}'_{\mathcal{I}\mathcal{J}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix} V \Sigma^{-1/2} + \varepsilon I.
\end{aligned}
$$

Consequently, there exists a nonzero vector $x \in \mathbb{R}^{|\mathcal{I}|+1} = \mathbb{R}^{|\mathcal{J}|+1}$ such that,

$$
\Sigma^{-1/2} U^\top \begin{bmatrix} \mathbb{Q}'_{\mathcal{I}\mathcal{J}} & -1_{|\mathcal{I}|} \\ 1_{|\mathcal{J}|}^\top & 0 \end{bmatrix} V \Sigma^{-1/2} x = -\varepsilon x. \qquad (D.54)
$$

This means that $-\varepsilon$ is an eigenvalue of the following deterministic matrix,

$$\Sigma^{-1/2} U^\top \begin{bmatrix} \mathbb{Q}'_{\mathcal{I}\mathcal{J}} & -1_{|\mathcal{I}|} \\ 1^\top_{|\mathcal{J}|} & 0 \end{bmatrix} V \Sigma^{-1/2}, \tag{D.55}$$

which happens with probability 0 since $\varepsilon \sim \text{ Unif } [-\lambda, \lambda]$ is continuous.

Optimality. Fix $\varepsilon > 0$. Consider a feasible solution to (D.35), $\left(R^{(\varepsilon)}, \nu^{(\varepsilon)}\right)$, that satisfies

$$\ell\left(R^{(\varepsilon)}, R^\circ\right) - C^\star < \frac{\varepsilon}{2}. \tag{D.56}$$

In particular, feasibility of $R^{(\varepsilon)}$ implies, for every $h \in [H], s \in \mathcal{S}$,

$$\begin{aligned}
\left[\mathbb{Q}_h^{(\varepsilon)}(s)\right]_{\mathcal{I}\bullet} \mathbf{q} &= \nu_h^{(\varepsilon)}(s)\, 1_{|\mathcal{I}|} \\
\mathbf{p}^\top \left[\mathbb{Q}_h^{(\varepsilon)}(s)\right]_{\bullet\mathcal{J}} &= \nu_h^{(\varepsilon)}(s)\, 1^\top_{|\mathcal{J}|} \\
\left[\mathbb{Q}_h^{(\varepsilon)}(s)\right]_{\mathcal{A}_1\setminus\mathcal{I}\bullet} \mathbf{q} &< \nu_h^{(\varepsilon)}(s)\, 1_{|\mathcal{A}_1\setminus\mathcal{I}|} \\
\mathbf{p}^\top \left[\mathbb{Q}_h^{(\varepsilon)}(s)\right]_{\bullet\mathcal{A}_2\setminus\mathcal{J}} &> \nu_h^{(\varepsilon)}(s)\, 1_{|\mathcal{A}_2\setminus\mathcal{J}|} \\
\sigma_{\min}\left(\begin{bmatrix} \mathbb{Q}_h^{(\varepsilon)}(s) & \mathcal{I}\mathcal{J} \\ -1_{|\mathcal{I}|} & 1_{|\mathcal{J}|} \end{bmatrix} 0\right) &> 0 \\
\mathbb{Q}_h^{(\varepsilon)}(s) &= R_h^{(\varepsilon)}(s) + \sum_{s'\in\mathcal{S}} P_h(s'|s)\, \nu_{h+1}^{(\varepsilon)}(s') \\
-b \leqslant \left[R_h^{(\varepsilon)}(s)\right]_{ij} &\leqslant b, \forall\ (i,j) \in \mathcal{A}.
\end{aligned} \tag{D.57}$$

Due to the strict SOW inequality in (D.35), we can find the $\iota^{(\varepsilon)} > 0$ such that the SOW conditions in (6.15) is also satisfied,

$$\begin{aligned}
\iota^{(\varepsilon)} \coloneqq \min_{(h\in[H], s\in\mathcal{S})} \Bigg\{ & \nu_h^{(\varepsilon)}(s)\, 1_{|\mathcal{A}_1\setminus\mathcal{I}|} - \left[\mathbb{Q}_h^{(\varepsilon)}(s)\right]_{\mathcal{A}_1\setminus\mathcal{I}\bullet} \mathbf{q}, \\
& \mathbf{p}^\top \left[\mathbb{Q}_h^{(\varepsilon)}(s)\right]_{\bullet\mathcal{A}_2\setminus\mathcal{J}} - \nu_h^{(\varepsilon)}(s)\, 1_{|\mathcal{A}_2\setminus\mathcal{J}|} \Bigg\},
\end{aligned} \tag{D.58}$$

where the min is element-wise for the vectors.

Since $\nu^{(\varepsilon)} \in (-Hb, Hb)$, we can find the value gap $\lambda^{(\varepsilon)} > 0$,

$$\lambda^{(\varepsilon)} \coloneqq b - \min_{h\in[H],s\in\mathcal{S},(i,j)\in\mathcal{A}} \left|\nu_h(s) - P_{ij}(s'|s)\,\nu_{h+1}(s')\right|, \tag{D.59}$$

by noting that if $\lambda^{(\varepsilon)} = 0$, then $\left|\nu^{(\varepsilon)}\right| \geqslant Hb$ which contradicts our assumption.

Now we define the following δ,

$$\delta \coloneqq \min\left\{\frac{\iota^{(\varepsilon)}}{2}, \frac{\varepsilon\lambda^{(\varepsilon)}}{2LbH(H+1)|\mathcal{S}||\mathcal{A}|}\right\}. \tag{D.60}$$

Note that $R^{(\varepsilon)}$ does not satisfy (6.15) due the tighter bounds on the entries, meaning $-b+\lambda \leqslant \left[R_h^{(\varepsilon)}(s)\right]_{ij} \leqslant b-\lambda$ may not be satisfied for some $h \in [H], s \in \mathcal{S}, (i,j) \in \mathcal{A}$. We define $R'^{(\varepsilon)}$ as follows and show that $\left(R'^{(\varepsilon)}, \nu^{(\varepsilon)}\right)$ is feasible to (6.15), for every $h \in [H], s \in \mathcal{S}, (i,j) \in \mathcal{A}$,

$$\left[R_h'^{(\varepsilon)}(s)\right]_{ij} \coloneqq \begin{cases} \left(1-\dfrac{\delta}{\lambda^{(\varepsilon)}}\right)\left[\mathbb{Q}_h^{(\varepsilon)}(s)\right]_{ij} + \dfrac{\nu_h^{(\varepsilon)}(s)\,\delta}{\lambda^{(\varepsilon)}} \\ \qquad -\displaystyle\sum_{s'\in\mathcal{S}} \left[P_h(s'|s)\right]_{ij} \nu_{h+1}^{(\varepsilon)}(s') & \text{if } i \in \mathcal{I}_h(s), j \in \mathcal{J}_h(s). \\ \min\left\{\max\left\{\left[R_h^{(\varepsilon)}(s)\right]_{ij}, -b+\delta\right\}, b-\delta\right\} & \text{otherwise} \end{cases} \tag{D.61}$$

In particular, we have for $i \in \mathcal{I}_h(s), j \in \mathcal{J}_h(s)$,

$$\left[\mathbb{Q}_h'^{(\varepsilon)}(s)\right]_{ij} = \left(1-\frac{\delta}{\lambda^{(\varepsilon)}}\right)\left[\mathbb{Q}_h^{(\varepsilon)}(s)\right]_{ij} + \frac{\nu_h^{(\varepsilon)}(s)\,\delta}{\lambda^{(\varepsilon)}}. \tag{D.62}$$

Now, to check the feasibility of $\left(R'^{(\varepsilon)}, \nu^{(\varepsilon)}\right)$ to (6.15), fix $h \in [H], s \in \mathcal{S}$. To

simplify the notations, we drop the (h, s) indices. Observe that

$$\begin{aligned}
\mathbb{Q}'^{(\varepsilon)}_{\mathcal{I}\bullet}\mathbf{q} &= \left(\left(1-\frac{\delta}{\lambda^{(\varepsilon)}}\right)\mathbb{Q}^{(\varepsilon)}_{\mathcal{I}\bullet} + \frac{\nu^{(\varepsilon)}\delta}{\lambda^{(\varepsilon)}}\right)\mathbf{q}, \text{ since } \mathbf{q}_{\mathcal{A}_2\setminus\mathcal{J}} = 0_{|\mathcal{A}_2\setminus\mathcal{J}|} \\
&= \left(1-\frac{\delta}{\lambda^{(\varepsilon)}}\right)\mathbb{Q}^{(\varepsilon)}_{\mathcal{I}\bullet}\mathbf{q} + \frac{\nu^{(\varepsilon)}\delta}{\lambda^{(\varepsilon)}}1_{\mathcal{I}\mathcal{J}}\mathbf{q} \\
&= \left(1-\frac{\delta}{\lambda^{(\varepsilon)}}\right)\nu^{(\varepsilon)}1_{|\mathcal{I}|} + \frac{\nu^{(\varepsilon)}\delta}{\lambda^{(\varepsilon)}}1_{\mathcal{I}\mathcal{J}}\mathbf{q}, \text{ since } \left(\mathsf{R}^{(\varepsilon)}, \nu^{(\varepsilon)}\right) \text{ is feasible} \\
&= \left(1-\frac{\delta}{\lambda^{(\varepsilon)}}\right)\nu^{(\varepsilon)} + \frac{\nu^{(\varepsilon)}\delta}{\lambda^{(\varepsilon)}} \\
&= \nu^{(\varepsilon)},
\end{aligned} \tag{D.63}$$

and similarly,

$$\begin{aligned}
\mathbf{p}^\top\mathbb{Q}'^{(\varepsilon)}_{\bullet\mathcal{J}} &= \mathbf{p}^\top\left(\left(1-\frac{\delta}{\lambda^{(\varepsilon)}}\right)\mathbb{Q}^{(\varepsilon)}_{\bullet\mathcal{J}} + \frac{\nu^{(\varepsilon)}\delta}{\lambda^{(\varepsilon)}}\right) \\
&= \left(1-\frac{\delta}{\lambda^{(\varepsilon)}}\right)\nu^{(\varepsilon)} + \frac{\nu^{(\varepsilon)}\delta}{\lambda^{(\varepsilon)}} \\
&= \nu^{(\varepsilon)}.
\end{aligned} \tag{D.64}$$

Consider any $\iota < \delta$, we have,

$$\begin{aligned}
\mathbb{Q}'^{(\varepsilon)}_{\mathcal{A}_1\setminus\mathcal{I}\bullet}\mathbf{q} &\leqslant \left(\mathbb{Q}^{(\varepsilon)}_{\mathcal{A}_1\setminus\mathcal{I}\bullet} + \frac{\iota^{(\varepsilon)}}{2}\right)\mathbf{q} \\
&\leqslant \mathbb{Q}^{(\varepsilon)}_{\mathcal{A}_1\setminus\mathcal{I}\bullet}\mathbf{q} + \frac{\iota^{(\varepsilon)}}{2}1_{|\mathcal{A}_1\setminus\mathcal{I}|} \\
&\leqslant \left(\nu^{(\varepsilon)} - \iota^{(\varepsilon)}\right)1_{|\mathcal{A}_1\setminus\mathcal{I}|} + \frac{\iota^{(\varepsilon)}}{2}1_{|\mathcal{A}_1\setminus\mathcal{I}|} \\
&\leqslant \left(\nu^{(\varepsilon)} - \frac{\iota^{(\varepsilon)}}{2}\right)1_{|\mathcal{A}_1\setminus\mathcal{I}|} \\
&\leqslant \left(\nu^{(\varepsilon)} - \delta\right)1_{|\mathcal{A}_1\setminus\mathcal{I}|} \\
&\leqslant \left(\nu^{(\varepsilon)} - \iota\right)1_{|\mathcal{A}_1\setminus\mathcal{I}|},
\end{aligned} \tag{D.65}$$

and similarly,

$$\begin{aligned}
\mathbf{p}^\top \mathbb{Q}'^{(\varepsilon)}_{\bullet \mathcal{A}_2 \setminus \mathcal{J}} &\geqslant \mathbf{p}^\top \left(\mathbb{Q}^{(\varepsilon)}_{\bullet \mathcal{A}_2 \setminus \mathcal{J}} - \frac{\iota^{(\varepsilon)}}{2} \right) \\
&\geqslant \left(\nu^{(\varepsilon)} + \iota^{(\varepsilon)} \right) 1_{|\mathcal{A}_2 \setminus \mathcal{J}|} - \frac{\iota^{(\varepsilon)}}{2} 1_{|\mathcal{A}_2 \setminus \mathcal{J}|} \\
&\geqslant \left(\nu^{(\varepsilon)} + \frac{\iota^{(\varepsilon)}}{2} \right) 1_{|\mathcal{A}_2 \setminus \mathcal{J}|} \\
&\geqslant \left(\nu^{(\varepsilon)} + \iota \right) 1_{|\mathcal{A}_2 \setminus \mathcal{J}|}.
\end{aligned} \tag{D.66}$$

Now to show that $\begin{bmatrix} \mathbb{Q}'^{(\varepsilon)}_{\mathcal{J}\mathcal{J}} & -1_{|\mathcal{J}|} \\ 1^\top_{|\mathcal{J}|} & 0 \end{bmatrix}$ is invertible, since $\begin{bmatrix} \mathbb{Q}^{(\varepsilon)}_{\mathcal{J}\mathcal{J}} & -1_{|\mathcal{J}|} \\ 1^\top_{|\mathcal{J}|} & 0 \end{bmatrix}$ is invert-

ible, there exists vector $\begin{bmatrix} x \\ t \end{bmatrix} \neq 0_{|\mathcal{J}|+1}$, such that

$$
\begin{aligned}
&\begin{bmatrix} Q_{\mathcal{J}\mathcal{J}}^{(\varepsilon)} & -1_{|\mathcal{J}|} \\ 1_{|\mathcal{J}|}^{\top} & 0 \end{bmatrix} \begin{bmatrix} x \\ t \end{bmatrix} = 0_{|\mathcal{J}|+1} \\
&\Rightarrow \begin{cases} Q_{\mathcal{J}\mathcal{J}}^{(\varepsilon)} x - t 1_{|\mathcal{J}|} = 0_{|\mathcal{J}|} \\ 1_{|\mathcal{J}|}^{\top} x = 0 \end{cases} \\
&\Rightarrow \begin{cases} \left(1 - \frac{\delta}{\lambda^{(\varepsilon)}}\right) Q_{\mathcal{J}\mathcal{J}}^{(\varepsilon)} x - \left(1 - \frac{\delta}{\lambda^{(\varepsilon)}}\right) t 1_{|\mathcal{J}|} = 0_{|\mathcal{J}|} \\ 1_{|\mathcal{J}|}^{\top} x = 0 \end{cases} \\
&\Rightarrow \begin{cases} \left(1 - \frac{\delta}{\lambda^{(\varepsilon)}}\right) Q_{\mathcal{J}\mathcal{J}}^{(\varepsilon)} x + \frac{\nu^{(\varepsilon)}\delta}{\lambda^{(\varepsilon)}} 1_{\mathcal{J}\mathcal{J}} x - \left(1 - \frac{\delta}{\lambda^{(\varepsilon)}}\right) t 1_{|\mathcal{J}|} = 0_{|\mathcal{J}|} \\ 1_{|\mathcal{J}|}^{\top} x = 0 \end{cases} \quad , \text{ since } 1_{|\mathcal{J}|}^{\top} x = 0 \\
&\Rightarrow \begin{cases} \left(\left(1 - \frac{\delta}{\lambda^{(\varepsilon)}}\right) Q_{\mathcal{J}\mathcal{J}}^{(\varepsilon)} + \frac{\nu^{(\varepsilon)}\delta}{\lambda^{(\varepsilon)}}\right) x - \left(1 - \frac{\delta}{\lambda^{(\varepsilon)}}\right) t 1_{|\mathcal{J}|} = 0_{|\mathcal{J}|} \\ 1_{|\mathcal{J}|}^{\top} x = 0 \end{cases} \\
&\Rightarrow \begin{cases} Q_{\mathcal{J}\mathcal{J}}^{\prime(\varepsilon)} x - \left(1 - \frac{\delta}{\lambda^{(\varepsilon)}}\right) t 1_{|\mathcal{J}|} = 0_{|\mathcal{J}|} \\ 1_{|\mathcal{J}|}^{\top} x = 0 \end{cases} \\
&\Rightarrow \begin{bmatrix} Q_{\mathcal{J}\mathcal{J}}^{\prime(\varepsilon)} & -1_{|\mathcal{J}|} \\ 1_{|\mathcal{J}|}^{\top} & 0 \end{bmatrix} \begin{bmatrix} x \\ \left(1 - \frac{\delta}{\lambda^{(\varepsilon)}}\right) t \end{bmatrix} = 0_{|\mathcal{J}|+1}.
\end{aligned}
\tag{D.67}
$$

Since $\begin{bmatrix} x \\ \left(1 - \frac{\delta}{\lambda^{(\varepsilon)}}\right) t \end{bmatrix} \neq 0_{|\mathcal{J}|+1}$, we have $\begin{bmatrix} Q_{\mathcal{J}\mathcal{J}}^{\prime(\varepsilon)} & -1_{|\mathcal{J}|} \\ 1_{|\mathcal{J}|}^{\top} & 0 \end{bmatrix}$ is invertible.

Since we did not change the value $\nu^{(\varepsilon)}$, the value range constraint is still satisfied,

$$
\underline{\nu} \leqslant \nu^{(\varepsilon)} \leqslant \overline{\nu}. \tag{D.68}
$$

For the range condition, we use the short-hand notation,

$$\Delta_{ij} v_h^{(\varepsilon)}(s) \coloneqq v_h^{(\varepsilon)}(s) - \sum_{s' \in \mathcal{S}} \left[P_h(s'|s) \right]_{ij} v_{h+1}^{(\varepsilon)}(s'). \tag{D.69}$$

Note that we have,

$$\begin{aligned}
R_h'^{(\varepsilon)}(s) &= Q_h'^{(\varepsilon)}(s) - \sum_{s' \in \mathcal{S}} P_h(s'|s)\, v_{h+1}^{(\varepsilon)}(s') \\
&= \left(1 - \frac{\delta}{\lambda^{(\varepsilon)}}\right) Q_h^{(\varepsilon)}(s) + v_h^{(\varepsilon)}(s) \frac{\delta}{\lambda^{(\varepsilon)}} - \sum_{s' \in \mathcal{S}} P_h(s'|s)\, v_{h+1}^{(\varepsilon)}(s') \\
&= \left(1 - \frac{\delta}{\lambda^{(\varepsilon)}}\right) \left(R_h^{(\varepsilon)}(s) + \sum_{s' \in \mathcal{S}} P v_{h+1}^{(\varepsilon)}(s') \right) \\
&\qquad + v_h^{(\varepsilon)}(s) \frac{\delta}{\lambda^{(\varepsilon)}} - \sum_{s' \in \mathcal{S}} P_h(s'|s)\, v_{h+1}^{(\varepsilon)}(s') \\
&= \left(1 - \frac{\delta}{\lambda^{(\varepsilon)}}\right) R^{(\varepsilon)} + \left(v_h^{(\varepsilon)}(s) - \sum_{s' \in \mathcal{S}} P_h(s'|s)\, v_{h+1}^{(\varepsilon)}(s') \right) \frac{\delta}{\lambda^{(\varepsilon)}} \\
&= \left(1 - \frac{\delta}{\lambda^{(\varepsilon)}}\right) R^{(\varepsilon)} + \Delta v_h^{(\varepsilon)}(s) \frac{\delta}{\lambda^{(\varepsilon)}},
\end{aligned} \tag{D.70}$$

where we drop the indices (h, s) as before. Now for any $\lambda < \delta$, we have,

for every $i \in \mathcal{I}, j \in \mathcal{J}$,

$$
\begin{aligned}
&-b \leqslant R_{ij}^{(\varepsilon)} \leqslant b \\
&\Rightarrow \left(1-\frac{\delta}{\lambda^{(\varepsilon)}}\right)(-b)+\frac{\delta}{\lambda^{(\varepsilon)}}\Delta_{ij}\nu^{(\varepsilon)} \leqslant \left(1-\frac{\delta}{\lambda^{(\varepsilon)}}\right) R_{ij}^{(\varepsilon)}+\frac{\delta}{\lambda^{(\varepsilon)}}\Delta_{ij}\nu^{(\varepsilon)} \\
&\qquad \leqslant \left(1-\frac{\delta}{\lambda^{(\varepsilon)}}\right) b+\frac{\delta}{\lambda^{(\varepsilon)}}\Delta_{ij}\nu^{(\varepsilon)} \\
&\Rightarrow -b+\delta\frac{b+\Delta_{ij}\nu^{(\varepsilon)}}{\lambda^{(\varepsilon)}} \leqslant R_{ij}^{\prime(\varepsilon)} \leqslant b-\delta\frac{b-\Delta_{ij}\nu^{(\varepsilon)}}{\lambda^{(\varepsilon)}} \\
&\Rightarrow -b+\delta\frac{b+\Delta_{ij}\nu^{(\varepsilon)}}{b-\min\limits_{i'j'}\left|\Delta_{i'j'}\nu^{(\varepsilon)}\right|} \leqslant R_{ij}^{\prime(\varepsilon)} \leqslant b-\delta\frac{b-\Delta_{ij}\nu^{(\varepsilon)}}{b-\min\limits_{i'j'}\left|\Delta_{i'j'}\nu^{(\varepsilon)}\right|} \\
&\Rightarrow -b+\delta \leqslant R_{ij}^{\prime(\varepsilon)} \leqslant b-\delta, \text{ since } b+\Delta_{ij}\nu^{(\varepsilon)} \geqslant b-\min_{i'j'}\left|\Delta_{i'j'}\nu^{(\varepsilon)}\right| \geqslant b-\Delta_{ij}\nu^{(\varepsilon)}, \\
&\Rightarrow -b+\lambda \leqslant R_{ij}^{\prime(\varepsilon)} \leqslant b-\lambda,
\end{aligned}
\tag{D.71}
$$

and for any other $(i,j) \in \mathcal{A}$,

$$
\begin{aligned}
&-b+\delta \leqslant \min\left\{\max\left\{R_{ij}^{(\varepsilon)}, -b+\delta\right\}, b-\delta\right\} \leqslant b-\delta \\
&\Rightarrow -b+\delta \leqslant R_{ij}^{\prime(\varepsilon)} \leqslant b-\delta \\
&\Rightarrow -b+\lambda \leqslant R_{ij}^{\prime(\varepsilon)} \leqslant b-\lambda.
\end{aligned}
\tag{D.72}
$$

In addition, we show that each entry changes by less than $\dfrac{\varepsilon}{2LH|\mathcal{S}||\mathcal{A}|}$, for

$i \in \mathcal{I}, j \in \mathcal{J}$. In particular, we have

$$
\begin{aligned}
&\left|R_{ij}'^{(\varepsilon)} - R_{ij}^{(\varepsilon)}\right| \\
&\leqslant \left|Q_{ij}'^{(\varepsilon)} - Q_{ij}^{(\varepsilon)}\right| \\
&\leqslant \left|\left(1 - \frac{\delta}{\lambda^{(\varepsilon)}}\right) Q_{ij}^{(\varepsilon)} + \frac{\nu^{(\varepsilon)}\delta}{\lambda^{(\varepsilon)}} - Q_{ij}^{(\varepsilon)}\right| \\
&= \left|-\frac{\delta}{\lambda^{(\varepsilon)}} Q_{ij}^{(\varepsilon)} + \frac{\nu^{(\varepsilon)}\delta}{\lambda^{(\varepsilon)}}\right| \\
&\leqslant \left|\frac{\delta}{\lambda^{(\varepsilon)}} Q_{ij}^{(\varepsilon)}\right| + \left|\frac{\nu^{(\varepsilon)}\delta}{\lambda^{(\varepsilon)}}\right| \\
&\leqslant \left|\frac{bH\delta}{\lambda^{(\varepsilon)}}\right| + \left|\frac{b\delta}{\lambda^{(\varepsilon)}}\right| \\
&\leqslant \frac{(H+1)\,b}{\lambda^{(\varepsilon)}} \frac{\varepsilon}{LH\,(H+1)\,|\mathcal{S}|\,|\mathcal{A}|} \frac{1}{2} \frac{\lambda^{(\varepsilon)}}{b}, \text{ due to the definition of } \delta, \\
&= \frac{\varepsilon}{2LH\,|\mathcal{S}|\,|\mathcal{A}|},
\end{aligned}
\tag{D.73}
$$

and for other $(i, j) \in \mathcal{A}$,

$$
\begin{aligned}
&\left|R_{ij}'^{(\varepsilon)} - R_{ij}^{(\varepsilon)}\right| \\
&\leqslant \left|\min\left\{\max\left\{R_{ij}^{(\varepsilon)}, -b\right\}, b\right\} - R_{ij}^{(\varepsilon)}\right| \\
&\leqslant \delta \\
&\leqslant \frac{\varepsilon\lambda^{(\varepsilon)}}{2LbH\,(H+1)\,|\mathcal{S}|\,|\mathcal{A}|} \\
&\leqslant \frac{\varepsilon}{2LH\,|\mathcal{S}|\,|\mathcal{A}|}, \text{ since } \lambda^{(\varepsilon)} \leqslant b.
\end{aligned}
\tag{D.74}
$$

Therefore we have,

$$\begin{aligned}
\ell\left(R^{\star}\right)-C^{\star} &\leqslant \ell\left(R'^{(\varepsilon)}\right)-C^{\star} \\
&\leqslant \ell\left(R'^{(\varepsilon)}-R^{(\varepsilon)}+R^{(\varepsilon)}\right)-C^{\star} \\
&\leqslant \ell\left(R^{(\varepsilon)}\right)-C^{\star}+L\left\|R'^{(\varepsilon)}-R^{(\varepsilon)}\right\|_{1} \\
&\leqslant \frac{\varepsilon}{2}-C^{\star}+LH|\mathcal{S}||\mathcal{A}|\frac{\varepsilon}{2LH|\mathcal{S}||\mathcal{A}|} \\
&\leqslant \frac{\varepsilon}{2}+L\frac{\varepsilon}{2L} \\
&=\varepsilon,
\end{aligned} \tag{D.75}$$

which concludes the proof. □

Additional Experiments

Code Details. We conducted our experiments using standard python3 libraries and the gurobi optimization package. We provide our code in a jupyter notebook with an associated database folder so that our experiments can be easily reproduced. The notebook already reads in the database by default so no file management is needed. Simply ensure the notebook is in the same directory as the database folder (like we have arranged in our uploaded zip). We note that for our benchmark tests, the database was too large to upload directly. Instead we will upload that database on github. However, the scale experiments can be reproduced by using the generation code we included in the notebook.

Classic Two-finger Morra. Consider the classic Two-finger Morra game. The game's payoff matrix is described in (D.76). Note that this game is different from the simplified two-finger morra game considered in the

main text.

$$\mathsf{TFM} := \begin{pmatrix} 0 & 2 & -3 & 0 \\ -2 & 0 & 0 & 3 \\ 3 & 0 & 0 & -4 \\ 0 & -3 & 4 & 0 \end{pmatrix} \tag{D.76}$$

TFW has infinitely many NEs: each player's strategy can be any convex combination of $(0, 4/7, 3/7, 0)^\top$ and $(0, 3/5, 2/5, 0)^\top$. Since people often naively use uniform mixing, it may be desirable to derive a similar game where uniform mixing is NE. Applying Algorithm 4 with $p = q = (1/4, 1/4, 1/4, 1/4)^\top$ produces the new payoff matrix (D.77).

$$\mathsf{TFM}^\dagger := \begin{pmatrix} 0 & 2 & -3 & 0 \\ -2 & 0 & -2 & 3 \\ 3 & 0 & 0 & -4 \\ -2 & -3 & 4 & 0 \end{pmatrix} \tag{D.77}$$

Observe that $\mathsf{TFW}^\dagger$ is an unfair game with value $-.25$, unlike the original game whose value was 0. The total cost for the change was 4.

5-action RPSSL. Consider the generalization of the rock-paper-scissors (RPS) game where each player now has 5 strategies rock, paper, scissors, spock, and lizard (RPSSL) that we mentioned in the main text. The game's payoff matrix is described in (D.78). Note that this game is different from the 5-action Rock-Paper-Scissor-Fire-Water (RPSFW) game considered in the main text.

$$\mathsf{RPSSL} := \begin{pmatrix} 0 & -1 & 1 & -1 & 1 \\ 1 & 0 & -1 & 1 & -1 \\ -1 & 1 & 0 & -1 & 1 \\ 1 & -1 & 1 & 0 & -1 \\ -1 & 1 & -1 & 1 & 0 \end{pmatrix} \tag{D.78}$$

Similar to RPS, the unique NE for RPSSL is the uniformly mixed strat-

egy pair $\mathbf{p} = \mathbf{q} = (1/5, 1/5, 1/5, 1/5, 1/5)^\top$. Suppose that instead, we wish to skew the distribution to favor the new actions, spock and lizard. Specifically, if $\mathbf{p} = \mathbf{q} = (1/9, 1/9, 1/9, 1/3, 1/3)^\top$, running Algorithm 4 produces the new payoff matrix (D.79).

$$\mathsf{RPSSL}^\dagger := \begin{pmatrix} 0 & -1 & 1 & -1 & 1 \\ 1 & 0 & -1 & 1 & -1 \\ -1 & 1 & 0 & -1 & 1 \\ 1 & -1 & 1 & 0 & -1/3 \\ -1 & 1 & -1 & 1/3 & 0 \end{pmatrix} \tag{D.79}$$

We observe the resultant NE is fair with value 0. The total cost for the change is 1.33.

REFERENCES

Abbeel, Pieter, and Andrew Y Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on machine learning*, 1.

Adler, Andy. 2005. Vulnerabilities in biometric encryption systems. In *International conference on audio-and video-based biometric person authentication*, 1100–1109. Springer.

Akchurina, Natalia. 2009. Multiagent reinforcement learning: algorithm converging to nash equilibrium in general-sum discounted stochastic games. In *Proceedings of the 8th international conference on autonomous agents and multiagent systems-volume 2*, 725–732.

Anderson, Ashton, Yoav Shoham, and Alon Altman. 2010. Internal implementation. In *Proceedings of the 9th international conference on autonomous agents and multiagent systems: volume 1-volume 1*, 191–198. Citeseer.

Appa, Gautam. 2002. On the uniqueness of solutions to linear programs. *Journal of the Operational Research Society* 53:1127–1132.

Auer, Peter, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. 2002. The nonstochastic multiarmed bandit problem. *SIAM journal on computing* 32(1):48–77.

Bab, Avraham, and Ronen I Brafman. 2008. Multi-agent reinforcement learning in common interest and fixed sum stochastic games: An experimental study. *Journal of Machine Learning Research* 9(12).

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

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

Barreno, Marco, Blaine Nelson, Anthony D Joseph, and J Doug Tygar. 2010. The security of machine learning. *Machine Learning* 81:121–148.

Behzadan, Vahid, and Arslan Munir. 2017. Vulnerability of deep reinforcement learning to policy induction attacks. In *International conference on machine learning and data mining in pattern recognition*, 262–275. Springer.

Bergemann, Dirk, and Juuso Välimäki. 2019. Dynamic mechanism design: An introduction. *Journal of Economic Literature* 57(2):235–274.

Berner, Christopher, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Dkebiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*.

Biggio, Battista, Blaine Nelson, and Pavel Laskov. 2011. Support vector machines under adversarial label noise. In *Asian conference on machine learning*, 97–112. PMLR.

Biggio, Battista, Ignazio Pillai, Samuel Rota Bulò, Davide Ariu, Marcello Pelillo, and Fabio Roli. 2013. Is data clustering in adversarial settings secure? In *Proceedings of the 2013 acm workshop on artificial intelligence and security*, 87–98.

Biggio, Battista, Konrad Rieck, Davide Ariu, Christian Wressnegger, Igino Corona, Giorgio Giacinto, and Fabio Roli. 2014. Poisoning behavioral malware clustering. In *Proceedings of the 2014 workshop on artificial intelligent and security workshop*, 27–36.

Blum, Avrim, and Yishay Mansour. 2007. Learning, regret minimization, and equilibria. *Algorithmic Game Theory*.

Bogunovic, Ilija, Arpan Losalka, Andreas Krause, and Jonathan Scarlett. 2021. Stochastic linear bandits robust to adversarial attacks. In *International conference on artificial intelligence and statistics*, 991–999. PMLR.

Bouville, Mathieu. 2008. Crime and punishment in scientific research. `0803.4058`.

Bowling, Michael. 2000. Convergence problems of general-sum multiagent reinforcement learning. In *Icml*, 89–94.

Bowling, Michael, and Manuela Veloso. 2001. Rational and convergent learning in stochastic games. In *International joint conference on artificial intelligence*, vol. 17, 1021–1026. Lawrence Erlbaum Associates Ltd.

Brandfonbrener, David, Will Whitney, Rajesh Ranganath, and Joan Bruna. 2021. Offline rl without off-policy evaluation. *Advances in Neural Information Processing Systems* 34:4933–4946.

Brown, Noam, and Tuomas Sandholm. 2019. Superhuman ai for multiplayer poker. *Science* 365(6456):885–890.

Brown, Noam, Tuomas Sandholm, and Strategic Machine. 2017. Libratus: The superhuman ai for no-limit poker. In *Ijcai*, 5226–5228.

Bubeck, Sébastien, and Nicolò Cesa-Bianchi. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning* 5(1):1–122.

Carlini, Nicholas, and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, 39–57. Ieee.

Choi, Jae-Deug, and Kee-Eung Kim. 2011. Inverse reinforcement learning in partially observable environments. *Journal of Machine Learning Research* 12:691–730.

Clancey, William J. 1979. Transfer of Rule-Based Expertise through a Tutorial Dialogue. Ph.D. diss., Dept. of Computer Science, Stanford Univ., Stanford, Calif.

———. 1983. Communication, Simulation, and Intelligent Agents: Implications of Personal Intelligent Machines for Medical Education. In *Proceedings of the eighth international joint conference on artificial intelligence* (*IJCAI-83*), 556–560. Menlo Park, Calif: IJCAI Organization.

———. 1984. Classification Problem Solving. In *Proceedings of the fourth national conference on artificial intelligence*, 45–54. Menlo Park, Calif.: AAAI Press.

———. 2021. The Engineering of Qualitative Models. Forthcoming.

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

———. 2022b. When is offline two-player zero-sum Markov game solvable? *arXiv preprint arXiv:2201.03522*.

Dalvi, Nilesh, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. 2004. Adversarial classification. In *Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining*, 99–108.

Dantzig, George. 1963. *Linear programming and extensions*. Princeton university press.

Dudek, Gregory, Michael RM Jenkin, Evangelos Milios, and David Wilkes. 1996. A taxonomy for multi-agent robotics. *Autonomous Robots* 3(4):375–397.

Engelmore, Robert, and Anthony Morgan, eds. 1986. *Blackboard systems*. Reading, Mass.: Addison-Wesley.

Fu, Wei, Chao Yu, Zelai Xu, Jiaqi Yang, and Yi Wu. 2022. Revisiting some common practices in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2206.07505*.

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

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

Good, RA. 1965. f-finger morra. *SIAM Review* 7(1):81–87.

Gu, Shixiang, Ethan Holly, Timothy Lillicrap, and Sergey Levine. 2017. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 ieee international conference on robotics and automation (icra)*, 3389–3396. IEEE.

Guan, Ziwei, Kaiyi Ji, Donald J Bucci Jr, Timothy Y Hu, Joseph Palombo, Michael Liston, and Yingbin Liang. 2020. Robust stochastic bandit algorithms under probabilistic unbounded adversarial attack. In *Proceedings of the aaai conference on artificial intelligence*, vol. 34, 4036–4043.

Guo, Wenbo, Xian Wu, Sui Huang, and Xinyu Xing. 2021. Adversarial policy learning in two-player competitive games. In *International conference on machine learning*, 3910–3919. PMLR.

Hart, Sergiu, and Andreu Mas-Colell. 2000. A simple adaptive procedure leading to correlated equilibrium. *Econometrica* 68(5):1127–1150.

Hasling, Diane Warner, William J. Clancey, and Glenn Rennels. 1984. Strategic explanations for a diagnostic consultation system. *International Journal of Man-Machine Studies* 20(1):3–19.

Hasling, Diane Warner, William J. Clancey, Glenn R. Rennels, and Thomas Test. 1983. Strategic Explanations in Consultation—Duplicate. *The International Journal of Man-Machine Studies* 20(1):3–19.

Heinrich, Johannes, Marc Lanctot, and David Silver. 2015. Fictitious self-play in extensive-form games. In *International conference on machine learning*, 805–813. PMLR.

Hernandez-Leal, Pablo, and Michael Kaisers. 2017. Towards a fast detection of opponents in repeated stochastic games. In *International conference on autonomous agents and multiagent systems*, 239–257. Springer.

Heuer, GA. 1979. Uniqueness of equilibrium points in bimatrix games. *International Journal of Game Theory* 8:13–25.

Hu, Junling, and Michael P Wellman. 2003. Nash q-learning for general-sum stochastic games. *Journal of machine learning research* 4(Nov):1039–1069.

Huang, Sandy, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. 2017a. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*.

———. 2017b. Adversarial attacks on neural network policies.

Huang, Yunhan, and Quanyan Zhu. 2019. Deceptive reinforcement learning under adversarial manipulations on cost signals. In *International conference on decision and game theory for security*, 217–237. Springer.

Jaderberg, Max, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. 2019. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science* 364(6443):859–865.

Jiang, Jiechuan, and Zongqing Lu. 2021. Offline decentralized multi-agent reinforcement learning. *arXiv preprint arXiv:2108.01832*.

Jin, Ying, Zhuoran Yang, and Zhaoran Wang. 2021a. Is pessimism provably efficient for offline rl? In *International conference on machine learning*, 5084–5096. PMLR.

———. 2021b. Is pessimism provably efficient for offline rl? In *International conference on machine learning*, 5084–5096. PMLR.

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

Kiourti, Panagiota, Kacper Wardega, Susmit Jha, and Wenchao Li. 2020. Trojdrl: evaluation of backdoor attacks on deep reinforcement learning. In *2020 57th acm/ieee design automation conference (dac)*, 1–6. IEEE.

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

Kos, Jernej, and Dawn Song. 2017a. Delving into adversarial attacks on deep policies. *arXiv preprint arXiv:1705.06452*.

———. 2017b. Delving into adversarial attacks on deep policies.

Kutschinski, Erich, Thomas Uthmann, and Daniel Polani. 2003. Learning competitive pricing strategies by multi-agent reinforcement learning. *Journal of Economic Dynamics and Control* 27(11-12):2207–2218.

Lee, Jae Won, and Jangmin O. 2002. A multi-agent q-learning framework for optimizing stock trading systems. In *International conference on database and expert systems applications*, 153–162. Springer.

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

Leibo, Joel Z, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. 2017. Multi-agent reinforcement learning in sequential social dilemmas. *arXiv preprint arXiv:1702.03037*.

Lin, Xiaomin, Stephen C Adams, and Peter A Beling. 2019a. Multi-agent inverse reinforcement learning for certain general-sum stochastic games. *Journal of Artificial Intelligence Research* 66:473–502.

Lin, Xiaomin, Stephen C. Adams, and Peter A. Beling. 2019b. Multi-agent inverse reinforcement learning for certain general-sum stochastic games. *Journal of Artificial Intelligence Research* 66:473–502.

Lin, Xiaomin, Peter A Beling, and Randy Cogill. 2014. Multi-agent inverse reinforcement learning for zero-sum games. *arXiv preprint arXiv:1403.6508*.

———. 2017a. Multiagent inverse reinforcement learning for two-person zero-sum games. *IEEE Transactions on Games* 10(1):56–68.

Lin, Yen-Chen, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. 2017b. Tactics of adversarial attack on deep reinforcement learning agents. *arXiv preprint arXiv:1703.06748*.

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

Liu, Fang, and Ness Shroff. 2019. Data poisoning attacks on stochastic bandits. In *International conference on machine learning*, 4042–4050. PMLR.

Liu, Guanlin, and Lifeng Lai. 2020. Action-manipulation attacks on stochastic bandits. In *Icassp 2020-2020 ieee international conference on acoustics, speech and signal processing (icassp)*, 3112–3116. IEEE.

———. 2021. Provably efficient black-box action poisoning attacks against reinforcement learning. *Advances in Neural Information Processing Systems* 34.

Liu, Yong, Yujing Hu, Yang Gao, Yingfeng Chen, and Changjie Fan. 2019. Value function transfer for deep multi-agent reinforcement learning based on n-step returns. In *Ijcai*, 457–463.

Lu, Shiyin, Guanghui Wang, and Lijun Zhang. 2021. Stochastic graphical bandits with adversarial corruptions. In *Proceedings of the aaai conference on artificial intelligence*, vol. 35, 8749–8757.

Lu, Yunlong, and Kai Yan. 2020. Algorithms in multi-agent systems: a holistic perspective from reinforcement learning and game theory. *arXiv preprint arXiv:2001.06487*.

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

Ma, Yuzhe, Kwang-Sung Jun, Lihong Li, and Xiaojin Zhu. 2018. Data poisoning attacks in contextual bandits. In *International conference on decision and game theory for security*, 186–204. Springer.

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

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

MacDermed, Liam, Charles Isbell, and Lora Weiss. 2011. Markov games of incomplete information for multi-agent reinforcement learning. In *Workshops at the twenty-fifth aaai conference on artificial intelligence*.

Mangasarian, Olvi. 1978. Uniqueness of solution in linear programming. Tech. Rep., University of Wisconsin-Madison Department of Computer Sciences.

Mannion, Patrick, Karl Mason, Sam Devlin, Jim Duggan, and Enda Howley. 2016. Dynamic economic emissions dispatch optimisation using multi-agent reinforcement learning. In *Proceedings of the adaptive and learning agents workshop (at aamas 2016)*.

Maskin, Eric, and Jean Tirole. 2001. Markov perfect equilibrium: I. observable actions. *Journal of Economic Theory* 100(2):191–219.

Meng, Linghui, Muning Wen, Yaodong Yang, Chenyang Le, Xiyun Li, Weinan Zhang, Ying Wen, Haifeng Zhang, Jun Wang, and Bo Xu. 2021. Offline pre-trained multi-agent decision transformer: One big sequence model conquers all starcraftii tasks. *arXiv preprint arXiv:2112.02845*.

Millham, CB. 1972. Constructing bimatrix games with special properties. *Naval Research Logistics Quarterly* 19(4):709–714.

Minagawa, Junichi. 2020. On the uniqueness of nash equilibrium in strategic-form games. *Journal of Dynamics & Games* 7(2):97.

Mohanty, Sharada, Erik Nygren, Florian Laurent, Manuel Schneider, Christian Scheller, Nilabha Bhattacharya, Jeremy Watson, Adrian Egli, Christian Eichenberger, Christian Baumberger, et al. 2020. Flatland-rl: Multi-agent reinforcement learning on trains. *arXiv preprint arXiv:2012.05893*.

Monderer, Dov, and Moshe Tennenholtz. 2003. k-implementation. In *Proceedings of the 4th acm conference on electronic commerce*, 19–28.

———. 2004. k-implementation. *Journal of Artificial Intelligence Research* 21:37–62.

Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 1765–1773.

NASA. 2015. Pluto: The 'other' red planet. Accessed: 2018-12-06.

Nash Jr, John F. 1950. Equilibrium points in n-person games. *Proceedings of the national academy of sciences* 36(1):48–49.

Ng, Andrew Y, Stuart Russell, et al. 2000. Algorithms for inverse reinforcement learning. In *Icml*, vol. 1, 2.

Ono, Norihiko, and Kenji Fukumoto. 1997. A modular approach to multi-agent reinforcement learning. In *Workshop on learning in distributed artificial intelligence systems, workshop on learning, interaction, and organization in multiagent environments*, 25–39. Springer, Berlin, Heidelberg.

Osborne, Martin J. 2004. *An introduction to game theory*, vol. 3. Oxford university press New York.

Pan, Ling, Longbo Huang, Tengyu Ma, and Huazhe Xu. 2022. Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification. In *International conference on machine learning*, 17221–17237. PMLR.

Pattanaik, Anay, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. 2017. Robust deep reinforcement learning with adversarial attacks. *arXiv preprint arXiv:1712.03632*.

Pavan, Alessandro, Ilya Segal, and Juuso Toikka. 2014. Dynamic mechanism design: A myersonian approach. *Econometrica* 82(2):601–653.

Prasad, HL, and Shalabh Bhatnagar. 2015. A study of gradient descent schemes for general-sum stochastic games. *arXiv preprint arXiv:1507.00093*.

Prasad, HL, Prashanth LA, and Shalabh Bhatnagar. 2015. Two-timescale algorithms for learning nash equilibria in general-sum stochastic games. In *Proceedings of the 2015 international conference on autonomous agents and multiagent systems*, 1371–1379.

Quintas, Luis G. 1988. *Uniqueness of Nash equilibrium points in bimatrix games*. Center for Mathematical Studies in Economics and Management Science.

Raghavan, TES. 1994. Zero-sum two-person games. *Handbook of game theory with economic applications* 2:735–768.

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

———. 2021a. Policy teaching in reinforcement learning via environment poisoning attacks. *Journal of Machine Learning Research* 22(210):1–45.

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

Ramachandran, Deepak, and Eyal Amir. 2007. Bayesian inverse reinforcement learning. In *Ijcai*, vol. 7, 2586–2591.

Rangi, Anshuka, Long Tran-Thanh, Haifeng Xu, and Massimo Franceschetti. 2022a. Saving stochastic bandits from poisoning attacks via limited data verification. In *Proceedings of the aaai conference on artificial intelligence*, vol. 36, 8054–8061.

Rangi, Anshuka, Haifeng Xu, Long Tran-Thanh, and Massimo Franceschetti. 2022b. Understanding the limits of poisoning attacks in episodic reinforcement learning. In *Proceedings of the thirty-first international joint conference on artificial intelligence, IJCAI-22*, ed. Lud De Raedt, 3394–3400. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Reddy, Tummalapalli Sudhamsh, Vamsikrishna Gopikrishna, Gergely Zaruba, and Manfred Huber. 2012. Inverse reinforcement learning for decentralized non-cooperative multiagent systems. In *2012 ieee international conference on systems, man, and cybernetics (smc)*, 1930–1935. IEEE.

Rice, James. 1986. Poligon: A System for Parallel Problem Solving. Technical Report KSL-86-19, Dept. of Computer Science, Stanford Univ.

Riedmiller, Martin, Thomas Gabel, Roland Hafner, and Sascha Lange. 2009. Reinforcement learning for robot soccer. *Autonomous Robots* 27: 55–73.

Robinson, Arthur L. 1980a. New ways to make microcircuits smaller. *Science* 208(4447):1019–1022. `https://science.sciencemag.org/content/208/4447/1019.full.pdf`.

———. 1980b. New Ways to Make Microcircuits Smaller—Duplicate Entry. *Science* 208:1019–1026.

Shalev-Shwartz, Shai, Shaked Shammah, and Amnon Shashua. 2016. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*.

Shapley, Lloyd S. 1953. Stochastic games. *Proceedings of the national academy of sciences* 39(10):1095–1100.

Silver, David, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou,

Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. *nature* 529(7587):484–489.

Silver, David, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, and Adrian Bolton. 2017. Mastering the game of go without human knowledge. *nature* 550(7676):354–359.

Slivkins, Aleksandrs. 2019. Introduction to multi-armed bandits. *arXiv preprint arXiv:1904.07272*.

Suematsu, Nobuo, and Akira Hayashi. 2002. A multiagent reinforcement learning algorithm using extended optimal response. In *Proceedings of the first international joint conference on autonomous agents and multiagent systems: Part 1*, 370–377.

Sun, Jianwen, Tianwei Zhang, Xiaofei Xie, Lei Ma, Yan Zheng, Kangjie Chen, and Yang Liu. 2020a. Stealthy and efficient adversarial attacks against deep reinforcement learning. In *Proceedings of the aaai conference on artificial intelligence*, vol. 34, 5883–5891.

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

Sun, Yanchao, Ruijie Zheng, Yongyuan Liang, and Furong Huang. 2021. Who is the strongest enemy? towards optimal and efficient evasion attacks in deep rl. *arXiv preprint arXiv:2106.05087*.

Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Szilágyi, Peter. 2006. On the uniqueness of the optimal solution in linear programming. *Revue d'analyse numérique et de théorie de l'approximation* 35(2):225–244.

Tagiew, Rustam. 2009. Hypotheses about typical general human strategic behavior in a concrete case. In *Congress of the italian association for artificial intelligence*, 476–485. Springer.

Tan, Ming. 1993. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, 330–337.

Terry, J, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, et al. 2021. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems* 34:15032–15043.

Tewolde, Emanuel. 2023. Game transformations that preserve Nash equilibria or best response sets. *arxiv preprint arXiv:2111.00076*.

Tramèr, Florian, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*.

Vinyals, Oriol, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, and Petko Georgiev. 2019. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature* 575(7782):350–354.

Vorotnikov, Sergey, Konstantin Ermishin, Anaid Nazarova, and Arkady Yuschenko. 2018. Multi-agent robotic systems in collaborative robotics. In *International conference on interactive collaborative robotics*, 270–279. Springer.

Wang, Lun, Zaynah Javed, Xian Wu, Wenbo Guo, Xinyu Xing, and Dawn Song. 2021. Backdoorl: Backdoor attack against competitive reinforcement learning. *arXiv preprint arXiv:2105.00579*.

Wang, Xianmin, Jing Li, Xiaohui Kuang, Yu-an Tan, and Jin Li. 2019a. The security of machine learning in an adversarial setting: A survey. *Journal of Parallel and Distributed Computing* 130:12–23.

Wang, Xingyu, and Diego Klabjan. 2018. Competitive multi-agent inverse reinforcement learning with sub-optimal demonstrations. In *International conference on machine learning*, 5143–5151. PMLR.

Wang, Yining, Ruosong Wang, Simon S Du, and Akshay Krishnamurthy. 2019b. Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136*.

Wei, Chen-Yu, Christoph Dann, and Julian Zimmert. 2022. A model selection approach for corruption robust reinforcement learning. In *International conference on algorithmic learning theory*, 1043–1096. PMLR.

Wikipedia contributors. 2021. Volunteer's dilemma — Wikipedia, the free encyclopedia. [Online; accessed 16-September-2021].

Wittel, Gregory L, and Shyhtsun Felix Wu. 2004. On attacking statistical spam filters. In *Ceas*.

Wright, Stephen J. 2006. *Numerical optimization*. New York, NY: Wiley.

Wu, Fan, Linyi Li, Zijian Huang, Yevgeniy Vorobeychik, Ding Zhao, and Bo Li. 2021. Crop: Certifying robust policies for reinforcement learning through functional smoothing. *arXiv preprint arXiv:2106.09292*.

Wu, Fan, Linyi Li, Chejian Xu, Huan Zhang, Bhavya Kailkhura, Krishnaram Kenthapadi, Ding Zhao, and Bo Li. 2022. Copa: Certifying robust

policies for offline reinforcement learning against poisoning attacks. *arXiv preprint arXiv:2203.08398*.

Wu, Young, Jeremy McMahan, Xiaojin Zhu, and Qiaomin Xie. 2023a. On faking a Nash equilibrium. *arXiv preprint arXiv:2306.08041*.

———. 2023b. Reward poisoning attacks on offline multi-agent reinforcement learning. In *The thirty-seventh aaai conference on artificial intelligence (aaai)*.

———. 2023c. Reward poisoning attacks on offline multi-agent reinforcement learning. In *Proceedings of the aaai conference on artificial intelligence*, vol. 37, 10426–10434.

Xie, Qiaomin, Yudong Chen, Zhaoran Wang, and Zhuoran Yang. 2020. Learning zero-sum simultaneous-move markov games using function approximation and correlated equilibrium. In *Conference on learning theory*, 3674–3682. PMLR.

Yang, Lin, Mohammad Hajiesmaili, Mohammad Sadegh Talebi, John Lui, and Wing Shing Wong. 2021. Adversarial bandits with corruptions: Regret lower bound and no-regret algorithm. In *Advances in neural information processing systems (neurips)*.

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

Yu, Lantao, Jiaming Song, and Stefano Ermon. 2019. Multi-agent adversarial inverse reinforcement learning. In *Proceedings of the 36th international conference on machine learning*, ed. Kamalika Chaudhuri and Ruslan Salakhutdinov, vol. 97 of *Proceedings of Machine Learning Research*, 7194–7201. PMLR.

Zhang, Haoqi, and David Parkes. 2008a. Value-based policy teaching with active indirect elicitation. In *Proceedings of the 23rd national conference on artificial intelligence - volume 1*, 208–214. AAAI'08, AAAI Press.

Zhang, Haoqi, and David C Parkes. 2008b. Value-based policy teaching with active indirect elicitation. In *Aaai*, vol. 8, 208–214.

Zhang, Haoqi, David C Parkes, and Yiling Chen. 2009. Policy teaching through reward function learning. In *Proceedings of the 10th acm conference on electronic commerce*, 295–304.

Zhang, Huan, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. 2020a. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in Neural Information Processing Systems* 33:21024–21037.

Zhang, Kaiqing, Zhuoran Yang, and Tamer Başar. 2021a. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control* 321–384.

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

Zhang, Xuezhou, Yiding Chen, Xiaojin Zhu, and Wen Sun. 2021c. Robust policy gradient against strong data corruption. In *International conference on machine learning*, 12391–12401. PMLR.

Zhang, Xuezhou, Yuzhe Ma, Adish Singla, and Xiaojin Zhu. 2020b. Adaptive reward-poisoning attacks against reinforcement learning. In *International conference on machine learning*, 11225–11234. PMLR.

Zhao, Wentao, Jun Long, Jianping Yin, Zhiping Cai, and Geming Xia. 2012. Sampling attack against active learning in adversarial environment. In

Modeling decisions for artificial intelligence: 9th international conference, mdai 2012, girona, catalonia, spain, november 21-23, 2012. proceedings 9, 222–233. Springer.

Zheng, Stephan, Alexander Trott, Sunil Srinivasa, Nikhil Naik, Melvin Gruesbeck, David C Parkes, and Richard Socher. 2020. The ai economist: Improving equality and productivity with ai-driven tax policies. *arXiv preprint arXiv:2004.13332*.

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

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