Inception: Efficiently Computable Misinformation Attacks on Markov Games

Jeremy McMahan
jmcmahan@wisc.edu
Dept. of Computer Sciences
University of Wisconsin-Madison

Young Wu
yw@cs.wisc.edu
Dept. of Computer Sciences
University of Wisconsin-Madison

Yudong Chen
yudong.chen@wisc.edu
Dept. of Computer Sciences
University of Wisconsin-Madison

Xiaojin Zhu
jerryzhu@cs.wisc.edu
Dept. of Computer Sciences
University of Wisconsin-Madison

Qiaomin Xie
qiaomin.xie@wisc.edu
Dept. of Industrial and Systems Engineering
University of Wisconsin-Madison

Abstract

We study security threats to Markov games due to information asymmetry and misinformation. We consider an attacker player who can spread misinformation about its reward function to influence the robust victim player's behavior. Given a fixed fake reward function, we derive the victim's policy under worst-case rationality and present polynomial-time algorithms to compute the attacker's optimal worst-case policy based on linear programming and backward induction. Then, we provide an efficient inception ("planting an idea in someone's mind") attack algorithm to find the optimal fake reward function within a restricted set of reward functions with dominant strategies. Importantly, our methods exploit the universal assumption of rationality to compute attacks efficiently. Thus, our work exposes a security vulnerability arising from standard game assumptions under misinformation.

1 Introduction

As multi-agent systems become increasingly decentralized and privacy-focused, games with incomplete information become inevitable. In many scenarios, a player only has partial information about the opponent's rewards and rationality, gleaned from external sources like the internet. However, misinformation spread by the opponent—possibly through fake news—can significantly impact the player's decision-making. For example, participants in first-price auctions may intentionally misrepresent their intended bids to manipulate other bids downward. To build robust multi-agent systems, it is crucial to understand the impact of misinformation on games.

We focus on two-player Markov Games (MG). We suppose that the second player, the attacker, knows both reward functions, (R_1, R_2). In contrast, the first player, the victim, only knows its reward function, R_1, and a misinformed attacker reward function, $R_2^\dagger$. A robust victim also constructs an uncertainty set $\Pi_2^b(R_2^\dagger)$ of possible attacker policies. Nevertheless, the attacker can choose $R_2^\dagger$ to manipulate the victim's behavior. We call these fake rewards *inception attacks*. The attacker's goal is to design an inception attack that optimizes its worst-case utility.

Although inception attacks can be devastating, computing optimal attacks is often challenging. Unlike standard reward poisoning (Wu et al., 2023b), an inception attack can not modify both players' rewards, which is necessary to illicit arbitrary victim behavior. Even if an oracle gave the attacker optimal fake rewards, computing a worst-case optimal attacker policy is a constrained optimization problem with nested maximins. Moreover, due to the information asymmetry, the attacker cannot utilize standard algorithms for computing robust optimization equilibrium (ROE) (Aghassi & Bertsimas, 2006) or Bayes-Nash equilibrium (BNE) (Harsanyi, 1967) to tackle this lower-level policy optimization problem.

Our Contributions. Although the computational complexity of inception might seem to limit its threat, we show that inception attacks can be efficiently computed by leveraging the universal rationality assumptions in multi-agent reinforcement learning (MARL). Specifically, for any rational or robust victim, we present an efficient algorithm for computing optimal dominant-policy inception attacks. The key insight is a rational victim always best-responds to a perceived attacker dominant strategy. Consequently, if the attacker focuses on fake reward functions admitting a dominant strategy, its complex optimization can be solved efficiently via backward induction. Our work exposes a security vulnerability arising from standard game assumptions under misinformation, motivating the need for novel approaches to building robust multi-agent systems.

To develop our inception algorithm, we first characterize outcomes in MGs with misinformation under worst-case rationality. Armed with these insights, we propose an efficient approach to compute the corresponding worst-case optimal policy for a given inception attack. Our method involves iteratively solving linear programs (LPs) based on worst-case Q functions. We derive these LPs by dualizing the best-response polytope, which transforms the maximin problems into maximization problems. Our approach accommodates any finitely generated victim uncertainty set, including completely naive and secure victims.

1.1 Related Work.

Information Asymmetry. Incomplete information games were first studied through the framework of Bayesian games (Harsanyi, 1967; 1968a;b) and with the solution concept being BNE. To address the high sensitivity of BNE to the player's beliefs (Rubinstein, 1989; Jehiel et al., 2006), the work (Holmström & Myerson, 1983) introduced a more robust equilibrium concept called ex-post equilibrium, which is a NE under all possible realizations of the uncertain parameters. Going beyond the need for belief distributions, (Aghassi & Bertsimas, 2006) introduced the notion of robust games with the solution concept being ROE. However, both the BNE and ROE approaches require non-trivial assumptions about the information structure, namely, an uncertainty parametrization or distributional assumption on the opponent's rewards. Thus, they do not apply to our setting where the victim knows nothing concrete about the attacker's true rewards.

Reward Poisoning Attacks. Most reward-poisoning attacks, for example, Ma et al. (2019); Rakhsha et al. (2020; 2021); Rangi et al. (2022); Zhang & Parkes (2008); Zhang et al. (2009) in the single-agent setting, and Wu et al. (2023d;c;a) in the multi-agent setting, focus on changing the victim's perceived rewards to induce negative behaviors rather than changing the victim's perception of the attacker's rewards. Unlike reward poisoning, which may not be possible in situations where the victim knows their preferences, inception attacks are more often possible since they fake the preferences of the attacker, which is usually not public information. Our setting also differs from past work by Gleave et al. (2019); Guo et al. (2021) on adversarial multi-agent reinforcement learning where an attacker is one of the agents (or controls one of the agents): they studied the problem in which an attacker modifies the action of an agent to influence the behavior of another agent (the victim).

1.2 Notations

We defer formal definitions of standard concepts in game theory to Appendix A.

Normal-form Games. Let $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{n\times m}$ denote the reward matrices for the victim and attacker, respectively. We represent a pure strategy by a one-hot vector, so $e_i \in \mathbb{R}^n$ corresponds to the victim's strategy i and $e_j \in \mathbb{R}^m$ the attacker's strategy j. Let $\Delta(k) := \left\{s \in [0,1]^k \mid \sum_{i=1}^k s_i = 1\right\}$ denote the set of mixed strategies, where $s \in \Delta(k)$ corresponds to playing e_i with probability s_i.

Markov Games. A finite-horizon *Markov game* (Shapley, 1953) is defined by a tuple $G = (S, \mathcal{A}, R, P, H, \mu)$ with state-space S, joint action space $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 = [n] \times [m]$ ($[i] := \{1, \dots, i\}$), joint reward function R, transition function P, horizon H, and initial state distribution μ. We denote by $\pi = \{\pi_{1,h}(s) \in \Delta(n) \times \Delta(m)\}_{h,s}$ a joint Markovian policy. Let Π_i denote the set of all Markovian policies for player $i \in \{1, 2\}$ (victim and attacker). The value received by player i under π is the expected total rewards over H steps: $V_i^\pi := \mathrm{E}_G^\pi \left[\sum_{h=1}^H \pi_{1,h}(s_h)^\top R_{i,h}(s_h)\pi_{2,h}(s_h)\right]$. Similarly we define the stage value, $V_{i,h}^\pi(s)$, for each $h \in [H]$ by summing rewards over steps h through H. Throughout the paper, we assume that players know the transition function P.

2 Inception

Reward Uncertainty. We formalize misinformation threats through Markov games with reward uncertainty. Suppose that the victim has learned an alleged $R_2^\dagger$ directly from the attacker or external sources. A robust victim is aware that $R_2^\dagger$ may be inaccurate, so it constructs an uncertainty set $\mathcal{U}(R_2^\dagger)$ that it believes contains the attacker's true rewards. Furthermore, the victim believes the attacker behaves as playing some policy $\pi_2 \in \Pi_2^b(\mathcal{U}(R_2^\dagger))$, which depends on the belief rewards. To simplify notation, we assume the victim's belief about the attacker takes the form $\Pi_2^b(R_2^\dagger) \subseteq \Pi_2$, with the understanding that the victim may be using robust reasoning inside the belief function.

Assumption 1 (Victim's Belief)**.** The victim knows some uncertain reward function $R_2^\dagger$ and believes the attacker's policy must lie in the set $\Pi_2^b(R_2^\dagger)$. Furthermore, this is common knowledge.

Example 1 (Naive Belief)**.** If the victim believes it knows exactly which policy $\pi_2^\dagger$ the attacker will play, then $\Pi_2^b(R_2^\dagger) = \{\pi_2^\dagger\}$.

Example 2 (Secure Belief)**.** If the victim believes it knows nothing about the attacker, it may assume any attacker policy is possible, $\Pi_2^b(R_2^\dagger) = \Pi_2$.

Example 3 (Rational Belief)**.** If the victim believes the standard assumption of common-knowledge rationality, which is the case if it uses any standard MARL algorithm, then it assumes the attacker is rational. Concretely, the victim might assume the attacker plays some solution to the perceived game, $\Pi_2^b(R_2^\dagger) = \{\pi_2 \in \Pi_2 \mid \exists \pi_1 \in \Pi_1, (\pi_1, \pi_2) \in Sol(R_1, R_2^\dagger)\}$, where Sol is any standard solution concept such as DSE, NE, and maximin equilibrium[1]. In this work, we focus on inception attacks that only require the most basic form of rationality: rational agents never play strictly dominated strategies (Wu et al., 2023b), which includes all the Sol options above.

2.1 Game Outcomes for Fixed $R_2^\dagger$

For any fixed $R_2^\dagger$, we can reason how both players will behave when the victim believes the attacker's policy is contained in the uncertainty set $\Pi_2^b(R_2^\dagger)$. To formally reason about the outcomes of such games, we turn to the standard notion of worst-case rationality (Aghassi & Bertsimas, 2006).

Assumption 2. (Worst-Case Rationality) Both players seek to optimize their worst-case value given their available information.

Victim Behavior. For the victim to be robust, it should optimize against the worst possible policy the attacker could play. By Assumption 1, it need only consider attacker policies in $\Pi_2^b(R_2^\dagger)$.

[1]The assumption also holds for CCE, where Sol corresponds to the marginal policy of the CCE for each player.

Observation 1 (Victim Behaviour)**.** *Under Assumption 1 and Assumption 2, the victim plays some policy* $\pi_1^* \in \Pi_1^*(R_2^\dagger)$ *and achieves the optimal worst-case value* $V_1^*(R_2^\dagger)$ *where,*

$$\Pi_1^*(R_2^\dagger) := \arg\max_{\pi_1 \in \Pi_1} \min_{\pi_2 \in \Pi_2^b(R_2^\dagger)} V_1^{\pi_1,\pi_2} \quad \textit{and} \quad V_1^*(R_2^\dagger) := \max_{\pi_1 \in \Pi_1} \min_{\pi_2 \in \Pi_2^b(R_2^\dagger)} V_1^{\pi_1,\pi_2}. \tag{VBR}$$

We observe that this behavior may be computationally intractable in general but is provably optimal under worst-case rationality. Also, this behavior can be viewed as a constrained security strategy that exploits the victim's beliefs to achieve better outcomes. This behavior directly generalizes security strategies, corresponding to the case when $\Pi_2^b(R_2^\dagger) = \Pi_2$.

Attacker Behavior. According to Assumption 1, the attacker knows $\Pi_2^b(R_2^\dagger)$. Thus, it can reason that the victim optimizes its worst-case value. Given this information, it can follow the same reasoning as the victim to predict how the victim behaves according to Observation 1. Specifically, the attacker should choose a policy that optimizes its value for the worst possible $\pi_1 \in \Pi_1^*(R_2^\dagger)$.

Observation 2. *Under Assumption 1 and Assumption 2, the attacker plays some* $\pi_2^* \in \Pi_2^*(R_2^\dagger)$ *and achieves the optimal worst-case value* $V_2^*(R_2^\dagger)$ *where,*

$$\Pi_2^*(R_2^\dagger) := \arg\max_{\pi_2 \in \Pi_2} \min_{\pi_1 \in \Pi_1^*(R_2^\dagger)} V_2^{\pi_1,\pi_2} \quad \textit{and} \quad V_2^*(R_2^\dagger) := \max_{\pi_2 \in \Pi_2} \min_{\pi_1 \in \Pi_1^*(R_2^\dagger)} V_2^{\pi_1,\pi_2}. \tag{ABR}$$

Importantly, the attacker exploits its information asymmetry to constrain the inner minimization. This allows the attacker to achieve a higher value than it would from a standard security strategy.

Overall, we can see exactly how the Markov game with reward uncertainty will play out.

Proposition 1 (Game Outcomes)**.** *For any fixed* $R_2^\dagger$*, under Assumption 1 and Assumption 2,* (π_1^*, π_2^*) *is a solution to the game if and only if* $(\pi_1^*, \pi_2^*) \in \Pi_1^*(R_2^\dagger) \times \Pi_2^*(R_2^\dagger)$.

2.2 Inception Attacks

The attacker can induce the fake reward $R_2^\dagger$ that the victim learns, possibly by spreading misinformation. For any induced $R_2^\dagger$, the attacker can achieve up to $V_2^*(R_2^\dagger)$ value in the worst-case according to Observation 2. Thus, the attacker should choose an inception attack, $R_2^\dagger$, that maximizes $V_2^*(R_2^\dagger)$.

Definition 1 (Inception)**.** An *optimal inception attack* is any $R_2^\dagger$ that achieves V_2^* where,

$$V_2^* := \max_{R_2^\dagger} V_2^*(R_2^\dagger). \tag{INC}$$

In general, (INC) is a complex, bi-level optimization problem. However, this does not mean the victim is safe from such attacks. We show in Section 3 that damaging inception attacks can be computed in polynomial time for many settings.

Example 4 (Inception Attack)**.** Consider the simple normal-form game (R_1, R_2) and its corresponding inception-attack-induced game $(R_1, R_2^\dagger)$ given in Figure 1. Also, suppose that the victim believes the attacker plays its part of an NE for the faked game, i.e., $\Pi_2^b(R_2^\dagger) = \{y \mid \exists x,\ (x, y) \in \mathrm{NE}(R_1, R_2^\dagger)\}$.

1. The original game in Figure 1a has a unique NE that is the pure strategy (D, L). Thus, $\Pi_2^b(R_2) = \{L\}$ and the victim plays its best-response D. This leads to the attacker always achieving a value of 0.
2. The fake game in Figure 1b has a unique NE which is the pure strategy (U, R). Thus, $\Pi_2^b(R_2^\dagger) = \{R\}$ and the victim plays its best-response U. This leads to the attacker always achieving its highest possible value of 5 for the true game.

Therefore, the attacker can simply fake that it prefers action R while it actually prefers action L to manipulate the victim into achieving its ideal value.

	L	R
U	0, 5	1, 0
D	1, 0	0, 0

(a) True Game

	L	R
U	0, 5	1, 5+ϵ
D	1, 0	0, ϵ

(b) Inception Attack

Figure 1: Inception Example

3 Efficient Inception Algorithms

In this section, we show that for certain families of victims, the optimal inception attacks can be computed efficiently. To start, we show for a fixed $R_2^\dagger$ how the attacker can efficiently compute some best response policy in $\Pi_2^*(R_2^\dagger)$, which is already a complex problem. Then, we move on to computing optimal inception attacks for restricted classes of reward functions.

3.1 Efficiently Exploiting $R_2^\dagger$

Suppose that $R_2^\dagger$ is fixed. We observe that computing some $\pi_2 \in \Pi_2^*(R_2^\dagger)$ is a complicated optimization problem with constraints and a nested maximin optimization. Specifically,

$$\begin{aligned} \Pi_2^*(R_2^\dagger) = &\arg\max_{\pi_2^* \in \Pi_2} \min_{\pi_1^* \in \Pi_1^*} V_2^{\pi_1^*, \pi_2^*} \\ &\text{s.t. } \Pi_1^* = \arg\max_{\pi_1 \in \Pi_1} \min_{\pi_2 \in \Pi_2^b(R_2^\dagger)} V_1^{\pi_1, \pi_2}. \end{aligned} \tag{1}$$

The optimization (1) can be arbitrarily complicated due to the arbitrary belief set $\Pi_2^b(R_2^\dagger)$. To have any hope of efficient solutions, we must restrict the belief set. Here, we consider any belief set that is a per-stage mixture of some finite set of base policies.

Assumption 3 (Finite Generation)**.** The victim's belief set is $\Pi_2^b(R_2^\dagger) = \Delta(\Pi)$, where $\Pi := \{\pi_2^1, \ldots, \pi_2^K\} \subseteq \Pi_2$ is a finite set of attacker policies and $\Delta(\Pi)$ is the simplex of per-stage mixings of Π, i.e.,

$$\Delta(\Pi) := \left\{ \pi \in \Pi_2 \mid \forall (h, s),\ \exists p \in \Delta(K) \ \text{ s.t. } \pi_{1,h}(s) = \sum_{k=1}^{K} p_k \pi_{2,h}^k(s) \right\}. \tag{2}$$

3.1.1 Normal-form Games

To see how Assumption Assumption 3 enables efficient computation, consider a normal-form game (A, B) and $\Pi = \{y_1, \ldots, y_K\} \subseteq \Delta(m)$.

Victim Best Response. It is well-known Dantzig (1951) that the victim can efficiently compute a maximin solution for A, i.e., $\max_{x \in \Delta(n)} \min_{y \in \Delta(m)} x^\top A y$, by solving the LP in Figure 2a. The inequalities $z \leq x^\top A e_j$ for all j ensure that x is the best response to any of the attacker's pure strategies, which then implies it is the best response to any mixture in $\Delta(m)$. In particular, x must be the best response to the worst possible mixed strategy in $\Delta(m)$.

The same reasoning applies if we replace each e_j with y_j. The inequalities $z \leq x^\top A y_j$ for all j then guarantee that x is a best response to the set $\Delta(\{y_1, \ldots, y_K\})$. Observe that we can equivalently formulate these inequalities by replacing A in Figure 2a with $A' := [Ay_1, \ldots, Ay_K] := A\Pi^\top$. Again, this implies x is the best response to the worst possible mixed strategy in $\Delta(\{y_1, \ldots, y_K\})$. Since $\Pi_1^*(R_2^\dagger)$ is the set of the victim's worst-case best responses to $\Pi_2^b(R_2^\dagger) = \Delta(\{y_1, \ldots, y_K\})$, we can compute some $x \in \Pi_1^*(R_2^\dagger)$ by solving LP Figure 2a with the modified reward matrix A'.

Lemma 1. *If (x^*, z^*) is a solution to LP 2a for input $A' := [Ay_1, \ldots, Ay_K]$, then $V_1^*(R_2^\dagger) = z^*$ and $x^* \in \Pi_1^*(R_2^\dagger)$. Furthermore, $\Pi_1^*(R_2^\dagger) = \{x \in \Delta(n) \mid \forall j \in [K],\ x^\top A' e_j \geq z^*\}$ is a non-empty polytope.*

$$\begin{aligned} \max_{x\in\mathbb{R}^n, z\in\mathbb{R}} \quad & z \\ \text{s.t.} \quad & z \le x^\top A e_j, \quad \forall j \in [m] \\ & 1^\top x = 1, \quad x \ge 0. \end{aligned}$$

(a) Victim's BR LP

$$\begin{aligned} \max_{y\in\mathbb{R}^m, w\in\mathbb{R}^K, \alpha\in\mathbb{R}} \quad & z^* 1^\top w - \alpha \\ \text{s.t.} \quad & \alpha + e_i^\top B y - e_i^\top A' w \ge 0 \quad \forall i \in [n] \\ & 1^\top y = 1, \quad y \ge 0 \quad w \ge 0. \end{aligned}$$

(b) Attacker's BR LP

Figure 2: Best-response LPs

Algorithm 1 Normal-Form Game Attacker Best Response

Require: Π, A, and B

1: $A' \leftarrow A\Pi^T$
2: $(x^*, z^*) \leftarrow Sol(LP\ 2a(A'))$
3: $(y^*, w^*, \alpha^*) \leftarrow Sol(LP\ 2b(z^*, A', B))$
4: **return** $(y^*, z^*, z^* 1^\top w^* - \alpha^*)$

Attacker Best Response. Now that we have understood the victim's best response $\Pi_1^*(R_2^\dagger)$ polytope, the attacker can exploit this structure to compute some $y \in \Pi_2^*(R_2^\dagger)$. Recall the attacker's true reward matrix is B. For any fixed y, note that the attacker's inner minimization in (1) can be written as the following LP and its dual in Figure 3.

$$\begin{aligned} \min_{x\in\mathbb{R}^n_{\ge 0}} \quad & x^\top B y \\ \text{s.t.} \quad & z^* - x^\top A' e_j \le 0, \quad \forall j \in [K], \\ & 1^\top x - 1 = 0. \end{aligned}$$

(a) Primal

$$\begin{aligned} \max_{w\in\mathbb{R}^K_{\ge 0}, \alpha\in\mathbb{R}} \quad & z^* 1^\top w - \alpha \\ \text{s.t.} \quad & \alpha + e_i^\top B y - e_i^\top A' w \ge 0, \quad \forall i \in [n]. \end{aligned}$$

(b) Dual

Figure 3: Attacker's Inner Minimization

Applying $\max_{y\in\Delta(m)}$ on top of (3b) yields the LP in Figure 2b, which computes a $y \in \Pi_2^*(R_2^\dagger)$. We give the full derivation in the Appendix.

Lemma 2. *If (y^*, w^*, α^*) is a solution to LP 2b, then $V_2^*(R_2^\dagger) = z^* 1^\top w^* - \alpha^*$ and $y^* \in \Pi_2^*(R_2^\dagger)$. Furthermore, $\Pi_2^*(R_2^\dagger)$ is a non-empty polytope.*

Therefore, the attacker can compute a $y \in \Pi_2^*(R_2^\dagger)$ by first computing a solution (x^*, z^*) to LP 2a and then using z^* to formulate and solve LP 2b. Importantly, the attacker can solve LP 2a due to the information asymmetry: it knows the victim's A. The computation is summarized in Algorithm 1.

Theorem 1. *If $K \le poly(m)$, then under Assumption 3 the attacker can compute some $y \in \Pi_2^*(R_2^\dagger)$ for a normal-form game in polynomial time by using Algorithm 1.*

3.1.2 Markov Games

To extend our results to full Markov games, we solve our LPs on each stage game via backward induction. To formalize this approach, we study the worst-case stage value and its corresponding worst-case Q functions:

$$V_{1,h}^*(s) := \max_{\pi_1\in\Pi_1} \min_{\pi_2\in\Pi_2^b(R_2^\dagger)} V_{1,h}^{\pi_1,\pi_2}(s) \text{ and } V_{2,h}^*(s) := \max_{\pi_2\in\Pi_2} \min_{\pi_1\in\Pi_1^*(R_2^\dagger)} V_{2,h}^{\pi_1,\pi_2}(s), \tag{3}$$

$$Q_{i,h}^*(s)[a_1, a_2] = R_{i,h}(s, a_1, a_2) + \sum_{s'} P_h(s' \mid s, a_1, a_2) V_{i,h+1}^*(s'). \tag{4}$$

Algorithm 2 Markov Game Attacker Best Response

Require: Π and G
1: $V^*_{i,H+1}(s) = 0$ for all $s \in S$.
2: **for** $h = H$ down to 1 **do**
3: **for** $s \in S$ **do**
4: $Q^*_{1,h}(s), Q^*_{2,h}(s) \leftarrow$ Equation (4)
5: $\pi^*_{2,h}(s), V^*_{1,h}(s), V^*_{2,h}(s) \leftarrow$ *Algorithm* 1$(\pi_{1,h}(s), Q^*_{1,h}(s), Q^*_{2,h}(s))$
6: **end for**
7: **end for**
8: **return** $\pi^*_2 := \{\pi^*_{2,h}(s)\}_{h,s}$

In particular, for each $h \in [H], s \in S$, the worst-case stage-value functions $V^*_{i,h}(s)$ can be computed from the worst-case Q functions $Q^*_{i,h}(s)$, using Algorithm 1 with $(Q^*_{1,h}(s), Q^*_{2,h}(s))$ as the norm-form game reward matrix. We let $\pi_{1,h}(s) := \{\pi^1_{2,h}(s), \ldots, \pi^K_{2,h}(s)\}$.

Lemma 3. *For all h, s, we have $(*, V^*_{1,h}(s), V^*_{2,h}(s)) =$ Algorithm 1$(\pi_{1,h}(s), Q^*_{1,h}(s), Q^*_{2,h}(s))$.*

Since the worst-case value is uniquely defined, we can use backward induction to compute a solution for the whole Markov game in Algorithm 2.

Theorem 2. *If $K \leq poly(m)$, then under Assumption 3 the attacker can compute some $\pi_2 \in \Pi^*_2(R^\dagger_2)$ for a Markov game in polynomial time using Algorithm 2.*

Remark 1 (Secure Victims). If the victim does not trust $R^\dagger_2$ as in Example 2 and simply ignores the information by computing a maximin strategy, $\max_{\pi_1 \in \Pi_1} \min_{\pi_2 \in \Pi_2} V_1^{\pi_1,\pi_2}$, the attacker can still exploit its information asymmetry. In particular, it can compute its best response in polynomial time using Algorithm 2 on $\Pi = \{\pi^j_2\}_{j=1}^m$ where $\pi^j_{2,h}(s) := e_j$. This leads to $\Delta(\Pi) = \Pi_2$.

3.2 Efficiently Optimizing $R^\dagger_2$

In the previous section, we saw how to compute best-response policies for a class of beliefs of the victim. However, to compute an optimal inception attack, we require additional structure on how the victim maps rewards to belief sets.

Assumption 4 (Common Rationality)**.** If $\pi^\dagger_2$ is an ι-strictly dominant Markov-perfect strategy for $R^\dagger_2$, then $\Pi^b_2(R^\dagger_2) = \{\pi^\dagger_2\}$.

Remark 2. (Rationality) Note that Assumption 4 holds whenever the victim believes common knowledge rationality as in Example 3. We again emphasize this assumption is made by all standard MARL algorithms as rationality is the basis of these game-theoretic approaches.

Policy Reduction. Observe that if $\Pi^b_2(R^\dagger_2) = \Pi^b_2(R^{\dagger\dagger}_2)$, then $V^*_2(R^\dagger_2) = V^*_2(R^{\dagger\dagger}_2)$. Consequently, whenever $\Pi^b_2(R^\dagger_2) = \{\pi^\dagger_2\}$, we see that $V^*_2(R^\dagger_2)$ is completely determined by $\pi^\dagger_2$ and not the specific structure of $R^\dagger_2$. Thus, with a slight abuse of notation, we can view V^*_2 as a function of $\pi^\dagger_2$ by defining $V^*_2(\pi^\dagger_2) := V^*_2(R^\dagger_2)$ where $R^\dagger_2$ is any reward functions satisfying $\Pi^b_2(R^\dagger_2) = \{\pi^\dagger_2\}$. Overall, we can reduce the problem of finding fake rewards to the problem of finding a fake policy.

If $\Pi^b_2(R^\dagger_2) = \{\pi^\dagger_2\}$, then by definition $\Pi^*_1(R^\dagger_2) = \arg\max_{\pi_1 \in \Pi_1} V_1^{\pi_1,\pi^\dagger_2} =: BR(\pi^\dagger_2)$ is just the victim's traditional best response to $\pi^\dagger_2$. In addition, $V^*_2(\pi^\dagger_2) = \max_{\pi_2 \in \Pi_2} \min_{\pi_1 \in BR(\pi^\dagger_2)} V_2^{\pi_1,\pi_2}$ can be efficiently computed using Algorithm 2. As only deterministic policies can be dominant, this simplifies the attacker's search to a finite set. Thus, the policy version of the problem is simpler to tackle. The attacker can then do inverse reward engineering to find a reward function for which $\pi^\dagger_2$ is a dominant strategy, which is possible even for robust victims Wu et al. (2023b).

Lemma 4 (Reward-Policy Reduction)**.** *Under Assumption 4,*

$$\max_{R^\dagger_2 \in \mathcal{D}} V^*_2(R^\dagger_2) = \max_{\pi^\dagger_2 \in \Pi^D_2} V^*_2(\pi^\dagger_2), \tag{5}$$

where $\mathcal{D}$ is the set reward functions with an ι-strictly dominant Markov-perfect strategy, and Π_2^D is the set of deterministic attacker policies. We let $\hat{V}_2 := \max_{\pi_2^\dagger \in \Pi_2^D} V_2^*(\pi_2^\dagger)$ denote the optimal value.

Lemma 4 states that if the misinformation-induced reward function $R_2^\dagger$ is restricted to the set admitting strictly dominant strategies, one can solve the optimal inception attack problem by solving the pure strategy optimization problem. We note this restricted set is infinite and captures many interesting reward functions.

Remark 3 (Reward Design). We note the choice of $R_{2,h}^\dagger(s,a) = \iota\frac{(H-h+1)(H-h+2)}{2}\mathbb{I}[a_2 = \pi_{2,h}^\dagger(s)]$ suffices to ensure $\pi_2^\dagger$ is the dominant strategy in any stage game and can be computed in polynomial time. If there are other constraints on the reward function, other reward poisoning frameworks can be used black box to compute optimal attacks.

Algorithmic Approach. For the normal-form game (A, B), it is easy to see that for any pure strategy $j \in [m]$ that $V_2^*(j) = \max_{y\in\Delta(m)} \min_{x\in BR(j)} x^\top By$ can be computed using *Algorithm* 1$(\{j\}, A, B)$ in polynomial time. The maximal pure strategy can then be found efficiently by iterating over all $j \in [m]$: $\hat{V}_2 = \max_j V_2^*(j)$. Thus, we can solve the policy problem for a normal-form game efficiently by repeatedly applying Algorithm 1.

This line of argument can be extended to Markov games by replacing (A, B) with the Q-function matrices and using backward induction. Suppose the attacker has already constructed a partial policy $\pi_2^\dagger$ for times $h+1, \ldots, H$. At time h and state s, the attacker can tentatively define $\pi_{2,h}^\dagger(s) = j$. For this choice, the attacker can reason about the victim's best-response set and value $\hat{V}_{1,h}(s,j)$, which is also constructed via backward induction. The attacker can then just choose the optimal j that leads to its highest worst-case stage value, $\hat{V}_{2,h}(s,j)$. Formally, we define,

$$\hat{V}_{2,h}(s) = \max_{\pi_2^\dagger \in \Pi_2^D} \min_{\pi_1 \in BR(\pi_2^\dagger)} V_{2,h}^{\pi_1,\pi_2}(s) \text{ and } \hat{V}_{1,h}(s) = \max_{\pi_1\in\Pi_1} V_{1,h}^{\pi_1,\pi_2^\dagger}(s), \tag{6}$$

to be the value of the best inception policy for the attacker at the current stage and the victim's best response value to a fixed inception policy $\pi_2^\dagger$, respectively. We can similarly define the corresponding $\hat{Q}$ function through (4) by replacing V^* with $\hat{V}$. Then, for any fixed $j \in [m]$, we define,

$$\hat{V}_{2,h}(s,j) = \max_{y\in\Delta(m)} \min_{x\in BR(j)} x^\top \hat{Q}_{2,h}(s)y \quad \text{and} \quad \hat{V}_{1,h}(s,j) = \max_{x\in\Delta(n)} x^\top \hat{Q}_{1,h}(s)e_j, \tag{7}$$

as the value when the attacker chooses $\pi_{2,h}^\dagger(s) = j$ at step h, and applies the optimal inception policy for times $h+1, \ldots, H$.

Lemma 5. *For all h, s, j, we have $(*, \hat{V}_{1,h}(s,j), \hat{V}_{2,h}(s,j)) =$ Algorithm 1$(\{j\}, \hat{Q}_{1,h}(s), \hat{Q}_{2,h}(s))$. Furthermore, if $j^* \in \arg\max_{j\in[m]} \hat{V}_{2,h}(s,j)$, then $\hat{V}_{i,h}(s) = \hat{V}_{i,h}(s,j^*)$ for each $i \in \{1,2\}$.*

In the same spirit as Algorithm 2, we can compute an optimal $\pi_2^\dagger$ using Algorithm 3.

Theorem 3. *Under Assumption 4, Algorithm 3 computes a fake policy achieving value $\hat{V}_2$ in polynomial time.*

Remark 4 (Dominant Mixtures). The algorithm can be extended to allow a mixture of a set of policies by changing $\{j\}$ to a subset of actions. This captures reward matrices with several equally dominant columns.

4 Conclusions

In this work, we studied misinformation attacks on two-player MGs. When the victim player only knows a false attacker reward function, we showed how the game plays out under worst-case rationality. Then, we showed how the attacker can compute its worst-case optimal policy in polynomial time. Using this method as a subroutine, the attacker can exploit the universal assumption of rationality in MARL to compute an optimal dominant-policy inception attack in polynomial time. Our

Algorithm 3 Policy Inception

Require: Π and G

$\hat{V}_{i,H+1}(s) = 0$ for all $s \in S$.
for $h = H$ down to 1 **do**
  for $s \in S$ **do**
    $\hat{Q}_{1,h}(s), \hat{Q}_{2,h}(s) \leftarrow$ Equation (4)
    for $j \in [m]$ **do**
      $\pi^*_{2,h}(s), \hat{V}_{1,h}(s,j), \hat{V}_{2,h}(s,j) \leftarrow$ *Algorithm* 1$(\{j\}, \hat{Q}_{1,h}(s), \hat{Q}_{2,h}(s))$
    end for
    $\pi^\dagger_{2,h}(s) \leftarrow \arg\max_{j\in[m]} \hat{V}_{2,h}(s,j)$
    $\hat{V}_{i,h}(s) \leftarrow \hat{V}_{i,h}(s, \pi^\dagger_{2,h}(s))$ for $i \in [2]$
  end for
end for
return $\pi^\dagger_2 := \{\pi^\dagger_{2,h}(s)\}_{h,s}$

work highlights that the standard rationality notions produce vulnerabilities when misinformation is present. Thus, new approaches are needed to build multi-agent systems that are robust against misinformation.

Broader Impact Statement

This paper presents work whose goal is to advance the field of MARL. Our work is largely theoretical, so we do not see any immediate negative societal impacts.

Acknowledgments

Xie was supported in part by National Science Foundation Awards CNS-1955997 and EPCN-2339794. Zhu was supported in part by NSF grants 1836978, 2023239, 2202457, 2331669, ARO MURI W911NF2110317, and AF CoE FA9550-18-1-0166. Chen is partially supported in part by NSF grant CCF-2233152.

References

Michele Aghassi and Dimitris Bertsimas. Robust game theory. *Mathematical Programming*, 107(1):231–273, 2006. doi: 10.1007/s10107-005-0686-0. URL https://doi.org/10.1007/s10107-005-0686-0.

George B Dantzig. A proof of the equivalence of the programming problem and the game problem. *Activity analysis of production and allocation*, 13, 1951.

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

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

John C. Harsanyi. Games with incomplete information played by "bayesian" players, i–iii part i. the basic model. *Management Science*, 14(3):159–182, 1967. doi: 10.1287/mnsc.14.3.159. URL https://doi.org/10.1287/mnsc.14.3.159.

John C. Harsanyi. Games with incomplete information played by "bayesian" players part ii. bayesian equilibrium points. *Management Science*, 14(5):320–334, 1968a. doi: 10.1287/mnsc.14.5.320. URL https://doi.org/10.1287/mnsc.14.5.320.

John C. Harsanyi. Games with incomplete information played by 'bayesian' players, part iii. the basic probability distribution of the game. *Management Science*, 14(7):486–502, 1968b. doi: 10.1287/mnsc.14.7.486. URL https://doi.org/10.1287/mnsc.14.7.486.

Bengt Holmström and Roger B. Myerson. Efficient and durable decision rules with incomplete information. *Econometrica*, 51(6):1799–1819, 1983. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1912117.

Philippe Jehiel, Moritz Meyer-ter Vehn, Benny Moldovanu, and William R. Zame. The limits of ex post implementation. *Econometrica*, 74(3):585–610, 2006. doi: https://doi.org/10.1111/j.1468-0262.2006.00675.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-0262.2006.00675.x.

Michael Kearns, Yishay Mansour, and Satinder Singh. Fast planning in stochastic games. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, UAI'00, pp. 309–316, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607099.

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

John Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951. ISSN 0003486X. URL http://www.jstor.org/stable/1969529.

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

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

Anshuka Rangi, Haifeng Xu, Long Tran-Thanh, and Massimo Franceschetti. Understanding the limits of poisoning attacks in episodic reinforcement learning. In Lud De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 3394–3400. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/471. URL https://doi.org/10.24963/ijcai.2022/471. Main Track.

Ariel Rubinstein. The electronic mail game: Strategic behavior under "almost common knowledge". *The American Economic Review*, 79(3):385–391, 1989. ISSN 00028282. URL http://www.jstor.org/stable/1806851.

L. S. Shapley. Stochastic games*. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953. doi: 10.1073/pnas.39.10.1095. URL https://www.pnas.org/doi/abs/10.1073/pnas.39.10.1095.

John von Neumann, Oskar Morgenstern, and Ariel Rubinstein. *Theory of Games and Economic Behavior (60th Anniversary Commemorative Edition)*. Princeton University Press, 1944. ISBN 9780691130613. URL http://www.jstor.org/stable/j.ctt1r2gkx.

Young Wu, Jeremy McMahan, Yiding Chen, Yudong Chen, Xiaojin Zhu, and Qiaomin Xie. Minimally modifying a markov game to achieve any nash equilibrium and value. *arXiv preprint arXiv:2311.00582*, 2023a.

Young Wu, Jeremy McMahan, Xiaojin Zhu, and Qiaomin Xie. Reward-poisoning attacks on offline multi-agent reinforcement learning. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023b. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i9.26240. URL https://doi.org/10.1609/aaai.v37i9.26240.

Young Wu, Jeremy McMahan, Xiaojin Zhu, and Qiaomin Xie. On faking a Nash equilibrium. *arXiv preprint arXiv:2306.08041*, 2023c.

Young Wu, Jeremy McMahan, Xiaojin Zhu, and Qiaomin Xie. Reward poisoning attacks on offline multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10426–10434, 2023d.

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

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

A Extended Preliminaries

Normal-form Games. In a (finite) normal-form game, two players compete simultaneously to maximize their reward. Suppose the first player, the victim, has n pure strategies and the second player, the attacker, has m pure strategies. Let $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{n\times m}$ denote the reward matrices for the victim and attacker, respectively. We may represent a pure strategy by a one-hot vector, so $e_i \in \mathbb{R}^n$ corresponds to the victim's strategy i and $e_j \in \mathbb{R}^m$ the attacker's strategy j. Let $\Delta(k) := \left\{s \in [0,1]^k \mid \sum_{i=1}^k s_i = 1\right\}$ denote the set of mixed strategies, where choosing $s \in \Delta(k)$ corresponds to playing e_i with probability s_i. For a pair of mixed strategies $x \in \Delta(n)$ and $y \in \Delta(m)$, the expected rewards to the victim and attacker are $x^\top Ay$ and $x^\top By$, respectively.

Nash Equilibrium. Solutions to games manifest as equilibrium concepts, among which the most famous is the *Nash Equilibrium* (NE) (Nash, 1951). An NE of a bimatrix game is a pair of strategies $(x^*, y^*) \in \Delta(n) \times \Delta(m)$ satisfying,

$$x^* \in \operatorname*{arg\,max}_{x\in\Delta(n)} x^\top Ay^* \quad \text{and} \quad y^* \in \operatorname*{arg\,max}_{y\in\Delta(m)} x^{*\top} By.$$

In words, x^* and y^* are mutual best-responses to each other. We let $NE(A, B)$ denote the set of all NEs for the game (A, B).

Security Strategies. Another solution concept is a *maximin strategy* or *security strategy*, which is a pair (x^*, y^*) given by,

$$x^* \in \operatorname*{arg\,max}_{x\in\Delta(n)} \min_{y\in\Delta(m)} x^\top Ay \quad \text{and} \quad y^* \in \operatorname*{arg\,max}_{y\in\Delta(m)} \min_{x\in\Delta(n)} x^\top By. \tag{8}$$

In a *zero-sum* game ($B = -A$), the Minimax Theorem (von Neumann et al., 1944) implies (x^*, y^*) is a NE if and only if it is a maximin strategy pair. Note that a game may have multiple NEs and maximin strategies. However, in zero-sum games, each player receives the same expected reward in every NE, which we denote by p_v^{NE} and p_e^{NE} respectively.

Markov Game Solutions. Equilibrium concepts can be defined for a Markov Game by viewing it as a (very large) bimatrix game with reward matrices $(V_1^{\pi_1,\pi_2})_{\pi_1,\pi_2}$ and $(V_2^{\pi_1,\pi_2})_{\pi_1,\pi_2}$. To avoid this complexity blowup, many works focus on *Markov Perfect Equilibrum* (MPE), which requires the stricter property that a policy pair is an equilibrium at *every* stage game, not just at stage $h = 1$. Formally, (π_1^*, π_2^*) is a MPE if, for all $(h, s) \in [H] \times S$,

$$V_{1,h}^{\pi_1^*,\pi_2^*}(s) = \max_{\pi_1\in\Pi_1} V_{1,h}^{\pi_1,\pi_2^*}(s) \quad \text{and} \quad V_{2,h}^{\pi_1^*,\pi_2^*}(s) = \max_{\pi_2\in\Pi_2} V_{2,h}^{\pi_1^*,\pi_2}(s).$$

B Proofs for Section 2

All the proofs from section 2 are immediate from the arguments given in the main text.

C Proofs for Section 3.1

As mentioned in the main text, the proof of Lemma 1 is immediate from standard bimatrix game theory (Dantzig, 1951).

C.1 Proof of Lemma 1

The proof is immediate from the argument given in the main text.

C.2 Proof of Lemma 2

To construct the dual in Figure 3, we introduce a dual vector $w \in \mathbb{R}^K_{\geq 0}$ corresponding to the inequality constraints and a dual variable $v \in \mathbb{R}$ corresponding to the equality constraint. We multiply these dual variables by their respective constraints and add them to the objective to get the equivalent optimization:

$$\max_{w \geq 0, v} \min_{x \geq 0} x^\top B y + (z^* 1^\top - x^\top A')w + (x^\top 1 - 1)v$$

By rearranging the objective to be in terms of x, we get:

$$\max_{w \geq 0, v} \min_{x \geq 0} x^\top (By - A'w + 1v) + z^* 1^\top w - v$$

Moving the terms involving x into the constraints then gives the Dual:

$$\begin{aligned} \max_{w \geq 0, \alpha} \quad & z^* 1^\top w - \alpha \\ \text{s.t.} \quad & \alpha + e_i^\top B y - e_i^\top A' w \geq 0 \quad \forall i \in [n], \end{aligned}$$

Applying $\max_{y \in \Delta(m)}$ outside of the Dual, yields the attacker's LP 2b:

$$\begin{aligned} \max_{y, w \in \mathbb{R}^K, \alpha \in \mathbb{R}} \quad & z^* 1^\top w - \alpha \\ \text{s.t.} \quad & \alpha + e_i^\top B y - e_i^\top A' w \geq 0 \quad \forall i \in [n] \\ & 1^\top y = 1, \quad y \geq 0 \quad w \geq 0. \end{aligned}$$

The fact that there exist optimal solutions, i.e., $\Pi_2^*(R_2^\dagger) \neq \varnothing$, follows from LP 2b being feasible and bounded. Specifically, it is easily seen that choosing $y = e_1$, $w = 0$, and $\alpha = \max_{i \in [n]} |e_i^\top B e_1$ gives a feasible solution to LP 2b. Boundedness follows from the fact that by LP duality, LP 2b is value equivalent to the original problem $\max_{y \in \Delta(m)} \min_{x \in \Pi_1^*(R_2^\dagger)} x^\top B y$, which is bounded being that (A, B) is a finite normal-form game. This completes the proof.

C.3 Proof of Theorem 1

The proof is immediate from Lemma 2.

C.4 Proof of Lemma 3

From Theorem 1 and the definition of Q^*, it suffices to show that V^* satisfies the following optimality equations:

$$V^*_{1,h}(s) = \max_{\pi_{1,h}(s) \in \Delta(n)} \min_{\pi_{2,h}(s) \in \pi_{1,h}(s)} \mathop{\mathrm{E}}_{a \sim \pi_{1,h}(s)} \left[R_{1,h}(s,a) + \sum_{s'} P_h(s' \mid s,a) V^*_{1,h+1}(s') \right], \tag{9}$$

and,

$$V^*_{2,h}(s) = \max_{\pi_{2,h}(s) \in \Delta(m)} \min_{\pi_{1,h}(s) \in \Pi^*_{1,h}(s)} \mathop{\mathrm{E}}_{a \sim \pi_{1,h}(s)} \left[R_{2,h}(s,a) + \sum_{s'} P_h(s' \mid s,a) V^*_{2,h+1}(s') \right], \tag{10}$$

where $\Pi^*_{1,h}(s)$ is the set of maximizers to (9). This follows from similar arguments to the proof of the NashVI algorithm Kearns et al. (2000) but with an added constraint set. For completeness, we give a full proof.

Proof. We show (10). The proof of (9) follows even easier as the constraint set is fixed in advance, independent of the attacker's actions. We proceed by induction on h. For the base case, consider the final time step $h = H + 1$. The claim is trivial as both values are 0. For the inductive step, consider any time step $h < H$ and fix any $s \in S$. Applying the bellman-consistency equations to the definition of $V^*_{2,h}(s)$ yields:

$$V^*_{2,h}(s) = \max_{\pi_2 \in \Pi_2} \min_{\pi_1 \in \Pi^*_1(R^\dagger_2)} \mathop{\mathrm{E}}_{a \sim \pi_{1,h}(s)} \left[R_{2,h}(s,a) + \sum_{s'} P_h(s' \mid s,a) V^\pi_{2,h}(s') \right].$$

Observe that the expression decomposes: the expectation only considers the policies at the current state and time, $(\pi_{1,h}(s), \pi_{2,h}(s))$, and the summation only considers the policies at future time steps. Consequently, we can break down the $\max_{\pi_2 \in \Pi_2}$ into the separate optimizations: $\max_{\pi_{2,h}(s) \in \Delta(m)}$ and $\max_{\pi_2 \in \Pi_{2,h+1}(s')}$ for each $s' \in S$, where $\Pi_{2,h+1}(s')$ is the set of partial policies for the attacker from time $h+1$ onwards starting at state s'.

Similarly, we can break down the $\min_{\pi_1 \in \Pi^*_1(R^\dagger_2)}$ into the separate optimizations: $\min_{\pi_{1,h}(s) \in \Pi^*_{1,h}(s)}$ and $\min_{\pi_1 \in \Pi^*_{1,h}(s')}$ for each $s' \in S$. This yields the equivalent optimization:

$$\max_{\pi_{2,h}(s) \in \Delta(m)} \max_{\pi_2 \in \times_{s'} \Pi_{2,h+1}(s')} \min_{\pi_{1,h}(s) \in \hat{\Pi}^*_{1,h}(s)} \min_{\pi \in \times_{s'} \Pi^*_{1,h}(s')} \mathop{\mathrm{E}}_{\pi_{1,h}(s), \pi_{2,h}(s)} [\ldots].$$

Now, consider the summation term inside of the optimization:

$$\mathop{\mathrm{E}}_{\pi_{1,h}(s), \pi_{2,h}(s)} \left[\sum_{s'} P_h(s' \mid s,a) V^\pi_{2,h+1}(s') \right].$$

We can apply linearity of expectation to get the equivalent term:

$$\sum_{s'} \mathop{\mathrm{E}}_{\pi_{1,h}(s), \pi_{2,h}(s)} \left[P_h(s' \mid s,a) V^\pi_{2,h+1}(s') \right].$$

Also, since $V^\pi_{2,h+1}(s')$ depends only on the partial policies at future steps, $V^\pi_{2,h+1}(s')$ is constant with respect to $(\pi_{1,h}(s), \pi_{2,h}(s))$ so can be pulled out of the summation to get the equivalent term:

$$\sum_{s'} V^\pi_{2,h+1}(s') \mathop{\mathrm{E}}_{\pi_{1,h}(s), \pi_{2,h}(s)} \left[P_h(s' \mid s,a) \right].$$

Now, by the induction hypothesis, we know for any s' at time $h+1$,

$$V^*_{2,h+1}(s') = \max_{\pi_{2,h+1}(s') \in \Pi_{2,h+1}(s')} \min_{\pi_{1,h+1}(s') \in \Pi^*_{1,h+1}(s')} \mathop{\mathrm{E}}_{\pi_{1,h+1}(s'), \pi_{2,h+1}(s')} \left[R_{2,h+1}(s',a) + \sum_{s'} P_{h+1}(s'' \mid s',a) V^*_{2,h+2}(s'') \right].$$

Since the term $V^*_{2,h+2}(s'')$ is fixed and shared amongst all s' at time $h+1$, we see the only variation in the stage value $V^*_{2,h+1}(s')$ comes from the choice of $(\pi_{1,h+1}(s'), \pi_{2,h+1}(s'))$ (i.e. varying the future partial policy cannot increase the objective value). These can be independently chosen for all s' at time $h+1$. Thus, the optimization problems $\max_{\pi_2 \in \Pi_{2,h+1}(s')} \min_{\pi_1 \in \Pi^*_{1,h+1}(s')} V^\pi_{2,h+1}(s') = V^*_{2,h+1}(s')$ are separable over s'. Thus, we can bring the maximin over partial policies into the summation to get the term:

$$\sum_{s'} \max_{\nu \in \Pi_{2,h+1}(s')} \min_{\pi \in \Pi^*_{1,h+1}(s')} V^\pi_{2,h+1}(s') \mathop{\mathrm{E}}_{\pi_{1,h}(s), \pi_{2,h}(s)} \left[P_h(s' \mid s,a) \right].$$

Since $V^*_{2,h+1}(s') = \max_{\pi_2 \in \Pi_{2,h+1}(s')} \min_{\pi_1 \in \Pi^*_{1,h+1}(s')} V^*_{2,h+1}(s')$, the expression becomes:

$$\sum_{s'} V^*_{2,h+1}(s') \mathop{\mathrm{E}}_{\pi_{1,h}(s), \pi_{2,h}(s)} \left[P_h(s' \mid s, a) \right].$$

As $V^*_{2,h+1}(s')$ is still constant with respect to $(\pi_{1,h}(s), \pi_{2,h}(s))$, we can reverse the previous steps of pulling out this term and applying linearity of expectation to get the final expression:

$$V^*_{2,h}(s) = \max_{\pi_{2,h}(s) \in \Delta(m)} \min_{\pi_{1,h}(s) \in \hat{\Pi}^*_{1,h}(s)} \mathop{\mathrm{E}}_{\pi_{1,h}(s), \pi_{2,h}(s)} \left[R_{2,h}(s,a) + \sum_{s'} P_h(s' \mid s, a) V^*_{2,h+1}(s') \right].$$

□

C.5 Proof of Theorem 2

The proof is immediate from Lemma 3.

D Proofs for Section 3.2

D.1 Proof of Lemma 4

The proof is immediate from the argument given in the main text.

D.2 Proof of Lemma 5

The proof follows similarly to the proof of Lemma 3 and the arguments from the main text.

D.3 Proof of Theorem 3

The proof is immediate from Lemma 5.