
Stackelberg Learning with Outcome-based Payment

Tom Yan
Carnegie Mellon University
tyyan@cmu.edu

Chicheng Zhang
University of Arizona
chichengz@cs.arizona.edu

Abstract

With businesses starting to deploy agents to act on their behalf, an emerging challenge that businesses have to contend with is how to incentivize other agents with differing interests to work alongside its own agent. In present day commerce, payment is a common way that different parties use to *economically* align their interests. In this paper, we study how one could analogously learn such payment schemes for aligning agents in the decentralized multi-agent setting. We model this problem as a Stackelberg Markov game, in which the leader can commit to a policy and also designate a set of outcome-based payments. We are interested in answering the question: when do efficient learning algorithms exist? To this end, we characterize the computational and statistical complexity of planning and learning in general-sum and cooperative games. In general-sum games, we find that planning is computationally intractable. In cooperative games, we show that learning can be statistically hard without payment and efficient with payment, showing that payment is necessary for learning even with aligned rewards. Altogether, our work aims to consolidate our theoretical understanding of outcome-based payment algorithms that can economically align decentralized agents.

1 Introduction

Increasingly, we are seeing businesses deploying agents to carry out tasks on their behalf. In the coming agentic era, we will inevitably have multiple, decentralized agents interacting together. An emerging challenge that businesses may have to face is how to incentivize other agents to work alongside its agent. This challenge requires addressing a central difficulty in decentralized multi-agent systems, which is that of differing interests.

In present day commerce, payment is a standard way that two parties use to resolve this challenge and more closely align their business interests. This inspires us to study the overarching question in this paper: how can we analogously implement such payment schemes in the multi-agent setting and enable *economic alignment*? That is, if I am a business looking to use payment to incentivize another business (and/or its agent) to work with my agent, how can I learn a good policy for my agent along with a payment scheme to go with it?

On a technical level, this setting may be viewed as a Stackelberg Markov game. In this paper, we study the two-player Stackelberg game, where one player (leader) commits to a policy taking into account the best response to the policy by the other player (follower). We focus on Stackelberg Markov games in particular as agents will be interacting over multiple turns and potentially long horizons. Finally, to model the payment aspect, the leader is able to also increase the reward of the follower in the Markov game, which may be viewed as a form of reward shaping in line with the existing formulation in the literature [Ben-Porat et al., 2024, Bollini et al., 2024, Ivanov et al., 2024, Scheid et al., 2024, Wu et al., 2024].

In this work, we aim to consolidate the theoretical foundations of Stackelberg learning with payment, as complexity results have yet to be established for two-player Stackelberg Markov games. We focus

Planning, Learning	Without Payment	With Payment	
	DAG	Tree	DAG
Cooperative	✓, ✗ (Theorem 5.15.2)	✓, ✓	✓, ✓ (Theorem 6.46.5)
General Sum	✗, ✗ (Proposition 4.1)	✓, ✓ (Proposition 4.3)	✗, ✗ (Theorem 4.2)

Table 1: Planning & learning settings where computationally *and* statistically efficient algorithms exist.

on a fundamental question: is there an efficient algorithm that can provably compute or learn the optimal policy and payment? Indeed, this is an important question to address as businesses in the future would want payment schemes with *provable guarantees*, so as to ensure that their expenditure is optimal.

Contributions: We analyze the planning and learning setting through both the computational and statistical lens. Please see Table 1 for an overview of our results.

1. We begin by considering planning in general-sum games. Is there an efficient algorithm that can return the optimal policy and payment? We prove that such a computationally efficient algorithm cannot exist unless $\text{NP}=\text{P}$, and identify the structural property of the MDP that results in this hardness. To complement the negative results, we develop an efficient algorithm, applicable when this property is removed.
2. Next, we turn to Cooperative games, which is a broad subclass of Markov games useful for modeling e.g. the interaction between AI service-providers and their users. Moreover, planning is computationally efficient in this setting, making it plausible that efficient learning algorithms may be attainable. As the rewards are already aligned, we begin by considering learning in the Stackelberg game without payment. Surprisingly, however, we find that an efficient algorithm cannot exist, this time in the statistical sense. We identify structural properties of the MDP that result in statistical hardness, and develop an efficient algorithm for when such properties are removed to complement our negative results.
3. Finally, we study learning in Cooperative games with payment. Can payment be used to alleviate the statistical hardness of learning? We answer this in the affirmative by showing that we can adapt existing no-regret RL algorithms to enable sample-efficient learning. In closing, we also use this setting to contrast the two different payment settings we study. We derive matching upper and lower regret bounds for when the leader has to make payments upfront versus on-the-fly, allowing us to quantitatively assess the benefits of being able to make payments on-the-fly.

2 Formulation

2.1 Stackelberg Markov Game

We consider the standard two-player, episodic finite-horizon Markov game M parameterized by $\langle S, A, B, H, P, \rho, r^L, r^F \rangle$ with state space S , initial state distribution $s_0 \sim \rho$, transitions P and episode length H . The leader has action set A and reward $r^L \in [-1, 1]$, the follower action set B and reward $r^F \in [-1, 1]$. In the case that the game is cooperative, $r^L = r^F$.

In the problem of online learning for Stackelberg Markov games, the learner plays the role of the leader, where apriori the reward functions r^L, r^F and the transitions are unknown to the leader. At each episode $k \in [T]$, the leader commits first to a policy π_k . The follower best responds to π_k with $\mu(\pi_k) \in \arg\max_{\mu} V^{\pi_k, \mu}(s_0; r^F)$. One may view best response as the equilibrium behavior of the follower to the leader policy.

After the episode, the leader and the follower observe the resultant trajectory $\tau_k = \{(s_i, a_i, b_i, r^L(s_i, a_i, b_i), r^F(s_i, a_i, b_i))\}_{i=1}^H$ realized by the chosen policies in M : $a_i \sim \pi_k(s_i)$, $b_i \sim \mu(\pi_k)(s_i)$, $s_{i+1} \sim P_i(\cdot | s_i, a_i, b_i)$.

This trajectory is the outcome of the policies' interaction, which in turn determines the *outcome-based* payment the follower receives.

Leader Payment: Following existing formulations in prior literature, the leader can increase r^F by creating outcome-based payment $b_i^k(s_i, a_i, b_i)$, if state-actions s_i, a_i, b_i are realized during the episode, $s_i, a_i, b_i \in \tau_k$. This results in a modified Markov game where the leader is able to additionally assign payment, with the payment function having signature $b_i^k : S \times A \times B \rightarrow \mathbb{R}^+$.

We note that the outcome-based payment need not correspond to direct monetary transfer. For example, we may be interested in modeling the setting where the leader is an AI-service-provider and the follower is a customer user. The leader spends money to improve its agent, and this improved agent adds additional value (e.g. more saved time) for the user during its use. But during this interaction, there is no direct transfer of money from the company to the user.

Thus, to model indirect payments in addition to direct ones, we introduce a final piece of notation, multiplier $\kappa \in \mathbb{R}^+$. $\kappa \cdot b_i^k(s_i, a_i, b_i)$ corresponds to the proportional cost to the leader in creating payment (reward) $b_i^k(s_i, a_i, b_i)$ for the follower. We believe proportionality is a natural assumption to make, and verily $\kappa = 1$ corresponds to direct payment.

2.2 Payment Settings

To complete the formulation, we touch on the two types of payment settings considered in this paper.

Trajectory Payment: The first is the existing payment setting commonly studied in prior literature, which we term trajectory payment. Here, a payment is made by the leader for every state-action on the realized trajectory. This form of payment is considered in principal-agent contracting literature, where the trajectory informing how much the leader will be paying ex-post [Dutting et al., 2021].

Moreover, this form of payment corresponds to the trendy outcome-based pricing model, which is experiencing rapid adoption by several notable SaaS companies due to the rising usage of AI agents [Stripe, 2025, Intercom, 2025, Zendesk, 2025]. Indeed, this marks a fundamental paradigm shift in software pricing in industry, moving from seat-based subscriptions (traditional SaaS) and usage-based models (cloud infrastructure) to now outcome-based pricing in the agent era [Boston Consulting Group, 2025, Sequoia Capital, 2025]. This also makes it imperative then to bolster our theoretical understanding of outcome based pricing, which we study in this paper.

Upfront Payment: In this paper, we will also consider a setting that we term upfront payment. As the name suggests, the leader pays for every state-action in the MDP, regardless of the realized trajectory. Note that the follower is still paid based on the realized trajectory. This is more realistic in settings where the leader pays indirectly to the benefit of the follower, and is bound by temporal constraints such that the payment cannot be made on-the-fly.

For a motivating example, consider the AI-service provider setting discussed earlier. The company invests before deployment to improve the agent’s functionality, which means that the user (follower) gains added value (reward) on the trajectory realized during the agent’s use. However, the key temporal constraint is that the company cannot improve its agent on-the-fly, as the users are using it. Thus, this makes upfront payment a more realistic model of the leader’s expenditure. The leader had to invest upfront to improve the agent’s capabilities in all states, even though this includes off-trajectory states that are not visited during the interaction with the user. For instance, suppose the agent is a computer-using-agent [Anthropic, 2024]. The user may use it to handle emails, and the agent would act in states of the computer corresponding to the inbox. However, even though the company had also invested to improve the agent’s capabilities in coding, the user may not invoke the agent to do so (perhaps due to excessive risk). And so, the agent would not have acted in other states of the computer corresponding to the codebase.

More generally, there is sizable body of economics contracting literature studying settings where only ex-ante (upfront) payment is possible. Some reasons for this include non-enforceable contracts, where the principal can renege upon observing the outcome Hart and Moore [1988]. Another cause for this may be non-verifiable outcomes; that is, when outcomes cannot be verified, ex-post contracts become unenforceable as there is no way to condition legally binding payments Aghion and Holden [2011]. Finally, one other reason may simply be that the agent is risk-averse, thus preferring upfront payment in face of stochastic outcomes Laffont and Martimort [2002].

Leader Optimization: Putting it all together, we can now write down the resulting Stackelberg game under the two payment settings.

Definition 2.1. In Stackelberg Markov games with trajectory payment, the leader optimizes:

$$\begin{aligned} \max_{\pi, b \geq 0} \quad & V^{\pi, \mu(\pi)}(s_0; r^L - \kappa \cdot b) \\ \text{s.t.} \quad & \mu(\pi) \in \operatorname{argmax}_{\mu'} V^{\pi, \mu'}(s_0; r^F + b) \end{aligned} \quad (1)$$

In Stackelberg Markov games with upfront payment, the leader optimizes:

$$\begin{aligned} \max_{\pi, b \geq 0} \quad & \left(V^{\pi, \mu(\pi)}(s_0; r^L) - \kappa \cdot \sum_{s, a, b \in S \times A \times B} b(s, a, b) \right) \\ \text{s.t.} \quad & \mu(\pi) \in \operatorname{argmax}_{\mu'} V^{\pi, \mu'}(s_0; r^F + b) \end{aligned} \quad (2)$$

Before moving on, we highlight the generality of the class of games we are studying. The class of Stackelberg Markov games with payment generalizes Stackelberg Markov games. Indeed, constraining the leader to zero payment (i.e. $(\pi, b) = (\pi, 0)$) corresponds to the leader’s policy space in Stackelberg Markov Games. Analogously, for Cooperative Stackelberg Markov games with payment studied in the later sections, this class of games generalizes Cooperative Stackelberg Markov games.

3 Related Works

As we focus on Stackelberg Markov games with payment, our paper is most related to two lines of work. The first is the line of work studying the complexity of Stackelberg policy computation in Markov games. And the second is algorithms for computing optimal payment schemes in MDPs. We cover both lines of work below, and include a discussion on additional related works in Appendix E.

Stackelberg Optimal Policies in Markov Games without Payment: Due to the wide applicability of the Stackelberg Markov games, there has been a long line of work seeking to understand how to compute optimal leader policies with provable guarantees.

For planning, Conitzer and Sandholm [2006], Letchford et al. [2012], Letchford and Conitzer [2010] study the computational tractability of optimal Stackelberg policy computation in Markov games and subclasses thereof. For stochastic MDPs, they establish that computing the optimal Stackelberg policy is NP-Hard.

For learning, Zhao et al. [2023] studies the statistical complexity in cooperative bandit games. Bai et al. [2021] studies the statistical complexity in bandit-RL games, a particular subclass of Markov games. Our work differs from this line of work in focusing on Markov games, which are more general than bandit-RL games and have longer horizon than bandit settings. Moreover, the leader is allowed to use payments to shape the follower’s rewards. As we will see, this turns out to be crucial for improved exploration during learning in certain settings.

Learning the Optimal Payment Scheme in MDPs: Recently, there has been burgeoning interest in computing optimal payment schemes for contracting agents to act in MDP environments, wherein the leader may increase the follower’s rewards as a form of reward shaping to incentivize the follower to play policies desirable to the leader.

The single-agent MDP setting, where only the follower acts in the MDP and the leader incentivizes, is formulated by Ben-Porat et al. [2024], Chen et al. [2022]. This is followed by a series of interesting work by Bollini et al. [2024], Ivanov et al. [2024], Wu et al. [2024], studying learning under a variety of different payment functions taking as input the state, the state-action or the state-next-state. Our work adds to this line of work by focusing on two-player Markov games, which generalize the single-player setting. Furthermore, while previous works mostly focus on trajectory payment, we also consider upfront payment, applicable in settings where the leader cannot pay on the fly due to temporal constraints. We derive tight regret guarantees to contrast the two differing payment settings.

The paper closest in formulation to that of ours is that by Scheid et al. [2024], who considers the same state-action based payment function in the bandit setting. Our work differs in focusing on Markov games, with a longer horizon than that in bandit settings. This in turn introduces difficulty in terms of exploration, and requires a more nuanced optimal payment computation beyond the binary search approach used in [Scheid et al., 2024].

Finally, as payment may be viewed as strategic reward shaping, our analysis is also related to existing RL literature that seeks to theoretically quantify the benefits of reward shaping [Ng et al., 1999]. Gupta et al. [2022] quantifies how statistical sample complexity is improved by reward shaping in the single-agent setting. By contrast, in our work, we study improved sample complexity in two-player cooperative Stackelberg Markov games.

4 Planning in General-sum Games

In this section, we ask: is there an efficient algorithm that can compute the optimal policy and payment in general-sum games? We investigate the computational complexity of such an algorithm, starting with the planning setting, where the Markov game dynamics and reward functions are known.

Our main finding is that there is no such computationally efficient algorithm unless $\text{NP}=\text{P}$. Outcome-based payment does not alleviate the computational intractability of computing the optimal Stackelberg policy, even in planning [Conitzer and Sandholm, 2006]. We identify that when the MDP has DAG structure, this leads to computational intractability. Later in the section, we complement this negative result with a positive result for when the MDP has tree structure. All proofs in this section may be found in Appendix A.

4.1 Hardness Results

We first derive a result showing that it is NP-Hard to compute the optimal leader policy even in deterministic MDPs, without payment. Note that in [Conitzer and Sandholm, 2006], computational intractability is demonstrated in stochastic MDPs.

Proposition 4.1. *Under Markov games that are deterministic DAGs, it is NP-Hard to compute the optimal policy:*

$$\begin{aligned} \max_{\pi} \quad & V^{\pi, \mu(\pi)}(s_0; r^L) \\ \text{s.t.} \quad & \mu(\pi) \in \operatorname{argmax}_{\mu'} V^{\pi, \mu'}(s_0; r^F) \end{aligned}$$

Helpfully, deterministic MDPs allow us to provide guarantees for both two payment settings. As we show in the proof, the optimal payment scheme pays zero in off-policy states, which can be readily characterized in deterministic MDPs. This result is intuitive as paying in off-policy states only incentivizes the follower to deviate off-policy, which is undesirable and increases leader total payment. With this result, we can derive that the optimal payment scheme is the same under trajectory and upfront payment. Thus, we use same construction, which provides a reduction to the PARTITION problem, to prove computational intractability under both payment settings.

Theorem 4.2. *Under Markov games that are deterministic DAGs, it is NP-Hard to compute the optimal policy and optimal trajectory payment:*

$$\begin{aligned} \max_{\pi, b \geq 0} \quad & V^{\pi, \mu(\pi)}(s_0; r^L - \kappa \cdot b) \\ \text{s.t.} \quad & \mu(\pi) \in \operatorname{argmax}_{\mu'} V^{\pi, \mu'}(s_0; r^F + b) \end{aligned}$$

and it is also NP-Hard to compute the optimal policy and optimal upfront payment:

$$\begin{aligned} \max_{\pi, b \geq 0} \quad & \left(V^{\pi, \mu(\pi)}(s_0; r^L) - \kappa \cdot \sum_{s, a, b \in S \times A \times B} b(s, a, b) \right) \\ \text{s.t.} \quad & \mu(\pi) \in \operatorname{argmax}_{\mu'} V^{\pi, \mu'}(s_0; r^F + b) \end{aligned}$$

In closing, we note that the optimal objective value of the subset of Markov games used to reduce to the PARTITION problem is an integral multiple of $1/2$. Due to this, we have that computational intractability in planning implies computational intractability in learning. In more detail, let M^* be the optimal objective value, which is an integer multiple of $1/2$. Suppose by contradiction that we had an algorithm with sublinear regret T^α ($\alpha < 1$). We can then set T large enough such that $T^\alpha/T < 1/2$. This allows us to infer M^* exactly by rounding to the nearest $1/2$, giving us a computationally efficient algorithm for answering the decision version of the PARTITION problem, which is a contradiction.

Algorithm 1 Planning Algorithm for MDP with Deterministic Tree Structure

Require: Pre-computed policy $\pi^- \in \operatorname{argmin}_{\pi} V^{\pi, \mu(\pi)}(s_0; r^F)$ (efficiently computed via Nash-VI)
for all root to leaf paths $\tau = s_1, a_1, b_1, s_2, a_2, b_2, \dots, s_H, a_H, b_H$ **do**
 Define $\pi(s_i) = a_i$ for $s_i, a_i \in \tau$. In every other state $s'_i \notin \tau$, let $\pi(s'_i) = \pi^-(s'_i)$.
 Compute $\mu(\pi)$ and compute follower Q-values, $Q^{\pi, \mu(\pi)}(\cdot, \cdot, \cdot)$.
 Solve for the minimal payment scheme using LP:

$$\begin{aligned} b^\tau(\pi) = \operatorname{argmin}_b \quad & \sum_{s_i, a_i, b_i \in \tau} b(s_i, a_i, b_i) \\ \text{s.t.} \quad & \sum_{i \geq h, s_i, a_i, b_i \in \tau} r^F(s_i, a_i, b_i) + b(s_i, a_i, b_i) \geq \max_{b'_h \neq b_h} Q^{\pi, \mu(\pi)}(s_h, a_h, b'_h; r^F) \end{aligned} \quad (3)$$

end for

Output the leader policy π and payment scheme of the path τ with maximal return $\sum_{s_i, a_i, b_i \in \tau} r^L(s_i, a_i, b_i) - \kappa \cdot b^\tau(\pi)$.

4.2 Positive Results

To complement our negative results, we show that positive results are attainable in MDPs without DAG structure. That is, in general-sum games where the MDP has tree structure, there is a polynomial-time algorithm for learning the optimal leader policy and payment. We describe our planning algorithm, Algorithm 1 that forms the crux of our approach to learning in this setting, and is applicable under both trajectory and upfront payment.

Proposition 4.3. *Under Markov games that are deterministic trees, there exists a polynomial-time planning algorithm that computes the optimal policy and payment.*

Remark 4.4. *To complete the result, we note in Appendix A that there is a simple exploration strategy using payment for general-sum, deterministic trees, as exploration needs to only recover rewards. This strategy allows us to reduce learning to planning, and then apply Algorithm 1.*

Before moving on, we note that in this general-sum game, the leader behaves in a zero-sum like manner in off-policy states in Algorithm 1. This incentivizes the follower to take the desired policy and allows the leader to minimize the total payment needed to incentivize such policy.

Finally, due to the intractability of computing a global Stackelberg optimum, it is natural to consider computing a local Stackelberg optimum instead, so that the policy and payment scheme does attain some guarantees. Building on existing results on first order methods in Stackelberg games [Shen et al., 2024], we derive a first order approach to this end. Note that while our paper is concerned with global Stackelberg optimality guarantees, we use this to illustrate that a more relaxed solution concept can be computed, if desired.

5 Learning in Cooperative Games without Payment

The computational intractability in the general-sum case prompts us to investigate whether efficient algorithms are attainable in significant subclasses of Markov games. Cooperative games are a broad subclass of Markov games useful for modeling e.g. the aforementioned AI-service based setting. Indeed, since the goal of the assistant agent is to aid the user, their rewards are aligned. And so, such settings correspond to a two-player cooperative game, making it an important subclass of Markov games to understand.

Moreover, on a technical level, it seems that there is hope for efficient algorithms as planning is efficient in cooperative games (e.g. via Nash-VI as in Bai and Jin [2020]). And so, in this section, we study the question: is there an efficient learning algorithm in cooperative games? We delve into this by first considering cooperative games without payment, which has yet to be addressed in the prior literature. Since the rewards are already aligned, we might expect that there are efficient learning algorithms. To our surprise, however, we find that learning in Cooperative Markov games

can be prohibitively hard, this time in the statistical sense. All proofs in this section may be found in Appendix B.

Structural properties of MDP: We identify the specific MDP properties under which exploration can be statistically intractable, along with complementary positive results. In a nutshell, we find that if the MDP has deterministic tree structure, then efficient algorithms are possible. However, allowing for stochastic or DAG transitions leads to statistical hardness.

Theorem 5.1. *There exists a turn-based Stochastic Tree Markov game such that: any (possibly randomized) algorithm that returns the optimal leader policy with probability at least $1/2$ requires at least $\Omega(2^{|S|})$ number of episodes.*

Theorem 5.2. *There exists a turn-based Deterministic DAG Markov game such that: any (possibly randomized) algorithm that returns the optimal leader policy with probability at least $1/2$ requires at least $\Omega(2^{|H|})$ number of episodes.*

Proposition 5.3. *Under Markov games that are deterministic trees, then there exists a polynomial-time algorithm that can learn a near-optimal leader policy.*

We remark that the statistical intractability results are based on a “needle-in-the-haystack” construction, where only a specific combination of leader actions is optimal. Structural properties of the MDP like stochastic or DAG transitions allow us to embed this construction in the MDP. Combined with the follower best responding instead of coordinating exploration with the leader, we can show that an exponential number of samples is needed by the leader to find the right combination, even if the rewards are already aligned.

Relaxing Follower Best Response behavior: As the statistical hardness is due to both the structural property of the MDP and the best response nature of the follower, a natural question one may ask is: can relaxing the latter alleviate statistical hardness and allow for efficient learning across all MDPs?

The natural way to relax best response is to consider best response under λ -entropy-regularization, which generalizes follower best response (corresponding to when $\lambda = \infty$). This behavior model is often used to model human behavior in human-AI interaction and behavioral economics literature [Ziebart et al., 2010, Reddy et al., 2018, McKelvey and Palfrey, 1995]. However, we again find that learning with this follower behavior does not allow for more sample efficient exploration:

Theorem 5.4. *There exists a turn-based Deterministic DAG Markov game such that: any (possibly randomized) algorithm that outputs the optimal policy given λ -Entropy-regularized best response with probability at least $1/2$ requires at least $\Omega(\exp(\lambda^2 H/8))$ episodes if $\lambda \leq 1$ and $\Omega(\exp(H/8))$ episodes if $\lambda > 1$.*

In closing, we offer a conceptual interpretation of the technical results in this section, using the example of the assistant agent and the user. Our results suggest that the service provider company can have difficulty exploring, due to the user’s best response. Indeed, users are simply looking to use the agent wherever it is at its best, and will not use the agent for the sake of its improvement. In particular, this means that users are not willing to use the agent in states that it currently does not currently excel in. Even though, these are precisely the states that the agent needs to obtain more training samples in. And so, this suggests that if the company wants to efficiently explore to learn an even better agent, incentivized exploration is needed.

6 Learning in Cooperative Games with Payment

In sum, we know from the previous section that in Stackelberg games, coordinated exploration is necessary for efficient learning. And so, in this section, we study how payment can be used to align the follower and enable efficient leader exploration. Our overall finding is that payment can lead to efficient exploration, and alleviate the statistical hardness in cooperative games without payment. All proofs in this section may be found in Appendix C.

6.1 Regret Guarantees in Cooperative Games

We study regret guarantees under the standard reinforcement learning setup with unknown transitions and unknown rewards, which can be stochastic.

Learning protocol: At each episode $k \in [T]$, the leader commits first to a policy π^k and a payment function b^k . The follower best responds to π^k with $\mu(\pi^k) \in \arg\max_{\mu} V^{\pi^k, \mu}(s_0; r^F + b^k)$. After the episode, the leader and the follower observe the resultant trajectory $\tau_k = \{(s_i, a_i, b_i, r^L(s_i, a_i, b_i))\}_{i=1}^H$ realized by the chosen policies in M (recall that $r^L = r^F$). The goal of the learner is to minimize its Stackelberg regret, defined as follows:

Definition 6.1. In Stackelberg games with trajectory payment, the Stackelberg regret is defined as:

$$\mathcal{R}(T) = \sum_{k=1}^T V^{\pi^*, \mu(\pi^*; r^F + b^*)}(s_0; r^L - \kappa \cdot b^*) - V^{\pi^k, \mu(\pi^k; r^F + b^k)}(s_0; r^L - \kappa \cdot b^k)$$

The regret under upfront payment may be defined analogously.

Towards analyzing Stackelberg regret, we characterize the optimal policy and trajectory payment when $r^L = r^F$; we can analogously show the same result under upfront payment.

Lemma 6.2. For any π^*, b^* such that:

$$\begin{aligned} \pi^*, b^* &= \arg\max_{\pi, b} V^{\pi, \mu(\pi; r^F + b)}(s_0; r^L - \kappa \cdot b) \\ \text{s.t. } \mu(\pi; r^F + b) &\in \arg\max_{\mu'} V^{\pi, \mu'}(s_0; r^F + b) \end{aligned} \quad (4)$$

If $r^L = r^F$, then we must have $\pi^*, \cdot = \arg\max_{\pi, \mu} V^{\pi, \mu}(s_0; r^L)$ and $b^* = 0$.

With this, we have that the optimal payment scheme in any cooperative game must be zero, as one would intuitively expect with already aligned rewards. This allows us to decompose Stackelberg regret into regret due to sub-optimality in policy and regret due to payment used during exploration, which will be responsible for the differing rates between trajectory and upfront payment.

Moreover, we note an interesting contrast due to this result. As we just saw, learning can be prohibitively hard in the absence of payment. Hence, we have that payment is not necessary in planning, but is crucial for learning (efficiently).

The crux of our positive results is that we can apply the canonical optimism under uncertainty principle to achieving sublinear Stackelberg regret. This follows from the observation that payment enables optimism in learning, which the leader can operationalize by setting payments according to its bonuses. This incentivizes the follower to also explore optimistically. A key lemma for bounding the policy regret portion of Stackelberg regret goes as follows.

Lemma 6.3. Suppose we can construct an optimistic MDP M_k of the true MDP M . Let the optimal leader policy under M_k be π_k , then:

$$\sum_{k=1}^T V_M^{\pi^*, \mu_M(\pi^*)}(s_0; r^L) - V_M^{\pi_k, \mu_{M_k}(\pi_k)}(s_0; r^L) \leq \sum_{k=1}^T V_{M_k}^{\pi_k, \mu_{M_k}(\pi_k)}(s_0; r^L) - V_M^{\pi_k, \mu_{M_k}(\pi_k)}(s_0; r^L)$$

Note that because the leader knows M_k , they know the policy $\mu_{M_k}(\pi_k)$ that they would like to incentivize the follower to play. Using this, we show that one can also bound the regret due to the cumulative payment, to obtain the following regret guarantees.

Theorem 6.4. UCB-VI-FP (Algorithm 2) incurs $O(T^{1/2})$ regret under trajectory payment. This is tight as there exists a subset of Markov games, where any learning algorithm must incur $\Omega(T^{1/2})$ regret.

Theorem 6.5. There exists an algorithm, leveraging UCB-VI-FP as subroutine, that incurs $O(T^{2/3})$ regret under upfront payment.

6.2 Contrasting Trajectory Payment with Upfront Payment

Finally, as positive results are attainable in Cooperative Markov games, we can analyze the difference in regret rates under the two different payment settings. What is the benefit afforded by settings where the leader can pay on-the-fly? Towards answering this question, we analyze the simple setup of unknown, deterministic rewards. Helpfully, this learning task already a sizable contrast in terms of regret between the two settings. We provide tight bounds on regret guarantees under both payment settings to contrast the two payment settings.

Algorithm 2 UCB-VI with Follower Payment (UCB-VI-FP)

Initialize $Q_h(s, a, b) = H$ for all $h \in [H], s, a, b \in S_h \times A \times B$.
for $k = 1, \dots, T$ **do**
 for $h = H, \dots, 1; s, a, b \in S_h \times A \times B$ **do** \triangleright construct M_k
 Compute estimated transitions from data in buffer: $\hat{P}_h(s'|s, a, b) = \frac{N_h^k(s, a, b, s')}{N_h^k(s, a, b)}$
 Compute optimistic rewards of M_k from reward samples in buffer: $\hat{r}_h^k(s, a, b) = \bar{r}_h^k(s, a, b) + c\sqrt{\frac{H^2}{N_h^k(s, a, b)}}$ \triangleright standard bonus for stochastic rewards
 $Q_h(s, a, b) = \min(H, \hat{r}_h^k(s, a, b) + \hat{P}_h^k V_{h+1}(s, a, b))$
 $V_h(s) = \max_{a, b} Q_h(s, a, b)$
 end for
 Leader commits to Stackelberg policy π^k : $\pi^k(s_h) = \operatorname{argmax}_a \max_b Q_h(s_h, a, b)$.
 Set outcome-based payment scheme: $\beta_h^k(s_h, a_h, b_h) = 2 \cdot c\sqrt{\frac{H^2 |S|}{N_h^k(s, a, b)}}$.
 for $h = 1, \dots, H$ **do**
 Leader plays $a_h^k \sim \pi^k(s_h^k)$, follower plays b_h^k via $\mu(\pi^k)$
 Transition to $s_{h+1}^k \sim P(\cdot | s_h^k, a_h^k, b_h^k)$ and save data $(s_h^k, a_h^k, b_h^k, s_{h+1}^k)$ in buffer
 end for
end for

Proposition 6.6. *UCB-VI-FP with indicator bonus incurs constant $O(|S||A||B|)$ regret under trajectory payment, where we designate reward under indicator bonus to be $\hat{r}(s, a, b) = 1\{\text{if } (s, a, b) \text{ is unvisited}\}$ and $r(s, a, b)$ o.w.*

As the regret bound is constant in T , we have that the bound must be tight. Next, we derive regret rates under upfront payment, whose regret lower bound requires a significantly nuanced probabilistic argument using Yao’s lemma.

Proposition 6.7. *There exists an algorithm, leveraging UCB-VI-FP with indicator bonus as subroutine, that incurs $O(T^{1/2})$ regret under upfront payment.*

Proposition 6.8. *There exists a subset of Markov Game instances such that any learning algorithm has to incur $\Omega(T^{1/2})$ regret under upfront payment.*

The construction of the negative result reveals the key difference in two payment schemes. In a nutshell, upfront payment is affected by difficult-to-reach states (ϵ -significant states [Jin et al., 2020]). On the other hand, trajectory payment is unaffected as the payment is made only if the follower does reach such a state. That is, the leader’s payment for actions in that statement is weighted by the visitation probability.

And so, the key difficulty in exploration under upfront payment is that when payment is needed to incentivize the follower to reach insignificant states, a lot of the payment can be wasted even if the follower is aligned, due to the low visitation probability. This is directly responsible for the sizable change in the regret guarantee, going from $O(1)$ to $\Omega(T^{1/2})$. Overall, this suggests that if the leader cannot pay on-the-fly, the payment scheme should factor in the reachability of states.

7 Discussion

In this work, we study learning in Stackelberg Markov games with payment. To consolidate the theoretical foundations of this setting, we chart the computational and statistical complexity of both planning and learning.

Future Work: Due to the intractability of general-sum settings, we believe that there is much more work to be done in analyzing more specific subclasses of Markov games. Which other subclasses of Markov games are such that efficient algorithms are attainable?

Limitations: One key assumption underlying our paper is that the follower’s action can be observed by the leader. We believe that this can be realistic for modeling certain digital settings (such as computers), wherein the agent’s actions can be readily tracked (computer-using-agent’s actions can be logged and monitored) [Anthropic, 2024, Sumers et al., 2025]. With that said, handling the case

for when the follower’s action is not observable is very important, especially in physical environment where monitoring is not possible. And we believe that results from the setting we study can serve as a stepping stone towards results in partial information settings with unobserved actions.

Another key underlying assumption is that the leader can readily observe the follower’s reward, either directly or through the follower’s report. It is conceivable that in cases the leader cannot observe the reward directly, the follower may not report their reward truthfully. In such settings, we note two observations. Let $(\pi^*(r), b^*(r))$ denote an optimal policy under reported follower reward r . Let r^F denote the true reward and r'^F the reported reward.

First, if we are in the cooperative setting, we observe that there is no incentive for the follower to misreport. Because the leader payment is zero, truthful reporting yields the highest return: $V^{\pi^*(r^F), \mu(\pi^*(r^F))}(s_0; r^F) \geq V^{\pi^*(r'^F), \mu(\pi^*(r'^F))}(s_0; r^F)$.

Second, in the general-sum bandit setting with direct payment considered by Scheid et al. [2024], the payment can now be nonzero but the follower’s gain from misreporting is bounded.

Proposition 7.1. *Suppose the follower can misreport r^F up to Δ , $\|r'^F - r^F\|_1 \leq \Delta$. In the bandit setting, the follower’s return can change by at most:*

$$|V^{\pi^*(r^F), \mu(\pi^*(r^F))}(s_0; r^F + b^*(r^F)) - V^{\pi^*(r'^F), \mu(\pi^*(r'^F))}(s_0; r^F + b^*(r'^F))| \leq 2\Delta$$

and the leader’s return can change by at most:

$$|V^{\pi^*(r^F), \mu(\pi^*(r^F))}(s_0; r^L - b^*(r^F)) - V^{\pi^*(r'^F), \mu(\pi^*(r'^F))}(s_0; r^L - b^*(r'^F))| \leq 2\Delta$$

However, an open question is whether such a bound carries over to the Markov game case. How much could the follower gain from misreporting r^F up to Δ ? Are there algorithms that can induce truthfulness, while still attaining some optimality guarantees? We believe there is a fruitful line of work to be done to handle cases where the leader cannot directly observe and/or verify the follower rewards.

8 Acknowledgement

TY is grateful for the support of the NSF GRFP. CZ is supported by NSF Award IIS-2440266.

References

- Omer Ben-Porat, Yishay Mansour, Michal Moshkovitz, and Boaz Taitler. Principal-agent reward shaping in mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9502–9510, 2024.
- Matteo Bollini, Francesco Bacchiocchi, Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Contracting with a reinforcement learning agent by playing trick or treat. *arXiv preprint arXiv:2410.13520*, 2024.
- Dima Ivanov, Paul Dütting, Inbal Talgam-Cohen, Tonghan Wang, and David C Parkes. Principal-agent reinforcement learning: Orchestrating ai agents with contracts. *arXiv preprint arXiv:2407.18074*, 2024.
- Antoine Scheid, Daniil Tiapkin, Etienne Boursier, Aymeric Capitaine, El Mahdi El Mhamdi, Éric Moulines, Michael I Jordan, and Alain Durmus. Incentivized learning in principal-agent bandit games. *arXiv preprint arXiv:2403.03811*, 2024.
- Jibang Wu, Siyu Chen, Mengdi Wang, Huazheng Wang, and Haifeng Xu. Contractual reinforcement learning: Pulling arms with invisible hands. *arXiv preprint arXiv:2407.01458*, 2024.
- Paul Dütting, Tim Roughgarden, and Inbal Talgam-Cohen. The complexity of contracts. *SIAM Journal on Computing*, 50(1):211–254, 2021.
- Stripe. Outcome-based pricing: A guide for businesses. Stripe Resources, 2025. URL <https://stripe.com/en-br/resources/more/outcome-based-pricing>.

- Intercom. Pricing AI agents: What does value-based pricing really mean for AI?, May 2025. URL <https://www.intercom.com/blog/pricing-ai-agents/>.
- Zendesk. Zendesk first in CX industry to offer outcome-based pricing for AI agents. Zendesk Newsroom, 2025. URL <https://www.zendesk.com/newsroom/articles/zendesk-outcome-based-pricing/>.
- Boston Consulting Group. Rethinking B2B software pricing in the agentic AI era. BCG Publications, 2025. URL <https://www.bcg.com/publications/2025/rethinking-b2b-software-pricing-in-the-era-of-ai>.
- Sequoia Capital. Pricing in the AI era: From inputs to outcomes, with Paid CEO Manny Medina. Sequoia Capital Podcast, 2025. URL <https://sequoiacap.com/podcast/pricing-in-the-ai-era-from-inputs-to-outcomes-with-paid-ceo-manny-medina/>.
- Anthropic. Developing a computer use model, 2024. URL <https://www.anthropic.com/news/developing-computer-use>.
- Oliver Hart and John Moore. Incomplete contracts and renegotiation. *Econometrica: Journal of the Econometric Society*, pages 755–785, 1988.
- Philippe Aghion and Richard Holden. Incomplete contracts and the theory of the firm: What have we learned over the past 25 years? *Journal of Economic Perspectives*, 25(2):181–197, 2011.
- Jean-Jacques Laffont and David Martimort. *The theory of incentives: the principal-agent model*. Princeton university press, 2002.
- Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 82–90, 2006.
- Joshua Letchford, Liam MacDermid, Vincent Conitzer, Ronald Parr, and Charles Isbell. Computing optimal strategies to commit to in stochastic games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 1380–1386, 2012.
- Joshua Letchford and Vincent Conitzer. Computing optimal strategies to commit to in extensive-form games. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 83–92, 2010.
- Geng Zhao, Banghua Zhu, Jiantao Jiao, and Michael Jordan. Online learning in stackelberg games with an omniscient follower. In *International Conference on Machine Learning*, pages 42304–42316. PMLR, 2023.
- Yu Bai, Chi Jin, Huan Wang, and Caiming Xiong. Sample-efficient learning of stackelberg equilibria in general-sum games. *Advances in Neural Information Processing Systems*, 34:25799–25811, 2021.
- Siyu Chen, Donglin Yang, Jiayang Li, Senmiao Wang, Zhuoran Yang, and Zhaoran Wang. Adaptive model design for markov decision process. In *International Conference on Machine Learning*, pages 3679–3700. PMLR, 2022.
- Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287. Citeseer, 1999.
- Abhishek Gupta, Aldo Pacchiano, Yuexiang Zhai, Sham Kakade, and Sergey Levine. Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity. *Advances in Neural Information Processing Systems*, 35:15281–15295, 2022.
- Han Shen, Zhuoran Yang, and Tianyi Chen. Principled penalty-based methods for bilevel reinforcement learning and rlhf. *arXiv preprint arXiv:2402.06886*, 2024.
- Yu Bai and Chi Jin. Provable self-play algorithms for competitive reinforcement learning. In *International conference on machine learning*, pages 551–560. PMLR, 2020.
- Brian D Ziebart, J Andrew Bagnell, and Anind K Dey. Modeling interaction via the principle of maximum causal entropy. 2010.

- Sid Reddy, Anca Dragan, and Sergey Levine. Where do you think you’re going?: Inferring beliefs about dynamics from behavior. *Advances in Neural Information Processing Systems*, 31, 2018.
- Richard D McKelvey and Thomas R Palfrey. Quantal response equilibria for normal form games. *Games and economic behavior*, 10(1):6–38, 1995.
- Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pages 4870–4879. PMLR, 2020.
- Theodore Sumers, Raj Agarwal, Nathan Bailey, Tim Belonax, Brian Clarke, Jasmine Deng, Evan Frondorf, Kyla Guru, Keegan Hanks, Jacob Klein, Lynx Lean, Kevin Lin, Linda Petrini, Madeleine Tucker, Ethan Perez, Mrinank Sharma, and Nikhil Saxena. Monitoring computer use via hierarchical summarization, 2025. URL <https://alignment.anthropic.com/2025/summarization-for-monitoring>.
- Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International conference on machine learning*, pages 263–272. PMLR, 2017.
- Han Zhong, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Can reinforcement learning find stackelberg-nash equilibria in general-sum markov games with myopically rational followers? *Journal of Machine Learning Research*, 24(35):1–52, 2023.
- Siyu Chen, Mengdi Wang, and Zhuoran Yang. Actions speak what you want: Provably sample-efficient reinforcement learning of the quantal stackelberg equilibrium from strategic feedbacks. *arXiv preprint arXiv:2307.14085*, 2023.
- Guru Guruganesh, Yoav Kolumbus, Jon Schneider, Inbal Talgam-Cohen, Emmanouil-Vasileios Vlatakis-Gkaragkounis, Joshua Wang, and S Weinberg. Contracting with a learning agent. *Advances in Neural Information Processing Systems*, 37:77366–77408, 2024.
- Hsu Kao, Chen-Yu Wei, and Vijay Subramanian. Decentralized cooperative reinforcement learning with hierarchical information structure. In *International Conference on Algorithmic Learning Theory*, pages 573–605. PMLR, 2022.
- Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. Commitment without regrets: Online learning in stackelberg security games. In *Proceedings of the sixteenth ACM conference on economics and computation*, pages 61–78, 2015.
- Matthias Gerstgrasser and David C Parkes. Oracles & followers: Stackelberg equilibria in deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 11213–11236. PMLR, 2023.
- Stephan Dempe and Alain B Zemkoho. On the karush–kuhn–tucker reformulation of the bilevel optimization problem. *Nonlinear Analysis: Theory, Methods & Applications*, 75(3):1202–1218, 2012.
- Roi Naveiro and David Ríos Insua. Gradient methods for solving stackelberg games. In *International conference on algorithmic decision theory*, pages 126–140. Springer, 2019.
- Vinzenz Thoma, Barna Pásztor, Andreas Krause, Giorgia Ramponi, and Yifan Hu. Contextual bilevel reinforcement learning for incentive alignment. *Advances in Neural Information Processing Systems*, 37:127369–127435, 2024.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: The abstract and introduction state the claims made, including the contributions made in the paper and important assumptions and limitations.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We discuss this in the Discussions section.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: We have written down all the proofs for the results in the paper, which are in the appendix due to the page limit of the main paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.

- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [NA]

Justification: This is a theoretical paper that does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: The paper does not include experiments requiring code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.

- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research conducted in the paper conform in every respect with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: This is a theoretical papers focused on foundational research. We do not anticipate this paper will have any immediate societal impact, especially negative ones.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We believe the paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: The paper does not use existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Proofs for Planning Results in General-sum Games

In [Conitzer and Sandholm, 2006], it is demonstrated that it is NP-Hard to compute the optimal Stackelberg policy, in absence of payment. But with the ability to pay, we are interested in answering the question: is there a general, efficient algorithm that can compute the optimal policy and payment?

A.1 Hardness Results

Proposition A.1. *Under Markov games that are deterministic DAGs, it is NP-Hard to compute the optimal policy:*

$$\begin{aligned} \max_{\pi} \quad & V^{\pi, \mu(\pi)}(s_0; r^L) \\ \text{s.t.} \quad & \mu(\pi) \in \operatorname{argmax}_{\mu'} V^{\pi, \mu'}(s_0; r^F) \end{aligned}$$

Proof. We show that one can reduce optimal policy computation to the DECISION Knapsack problem.

For a given knapsack instance $\langle \{v_i\}_i, \{w_i\}_i, W \rangle$. Construct the following MDP:

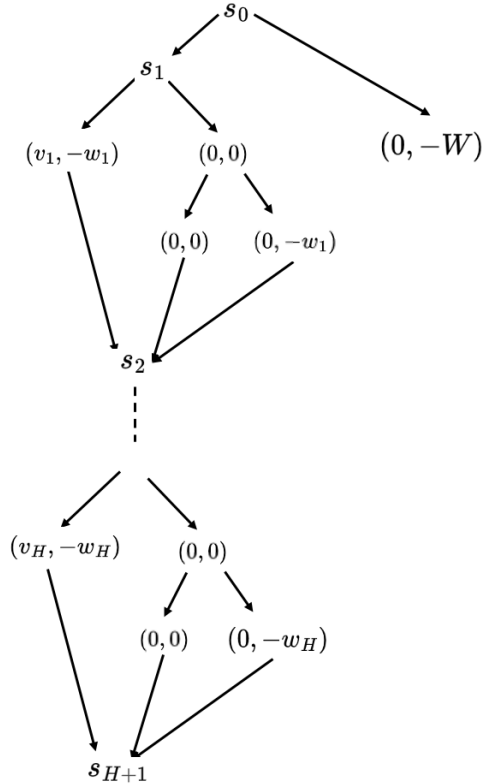


Figure 1

At the i th time step, the follower will select one of the two actions with rewards $(v_i, -w_i)$ or $(0, 0)$. The leader influences this through the probability $\pi(s_i) = \pi_i$ of playing $(0, -w_i)$. Here, we assume that the follower plays in favor of the leader in the event of a tie-break. That is, the follower plays the left action with reward $(v_i, -w_i)$ iff the leader plays $(0, -w_i)$ w.p. $\pi_i = 1$.

Therefore, we have that an optimal leader policy π^* must maximize the following objective:

$$\begin{aligned}
& \max_{\pi_1, \dots, \pi_H} \sum_{i=1}^H v_i \mathbb{1} \{ \pi_i = 1 \} \\
& \text{s.t.} \quad \sum_{i=1}^H w_i \pi_i \leq W \\
& \quad 0 \leq \pi_i \leq 1
\end{aligned} \tag{5}$$

since the leader wishes to incentivize the follower to play in the left branch (holds iff $\sum_{i=1}^H \pi_i (-w_i) \geq -W$), while maximizing the return in the left branch.

We will show that introducing the constraint $\pi_i \in \{0, 1\}$ is without loss of optimality. Consider some optimal policy π^* . For any i such that $\pi_i^* \in (0, 1)$, let $\pi_i' = 0$ and let $\pi_i' = \pi_i^*$ otherwise. It follows that π_i' is still feasible, while retaining the same objective value.

Hence, given some π^* optimal leader policy, its return matches the optimal objective value of the program:

$$\begin{aligned}
\sum_{i=1}^H v_i \mathbb{1} \{ \pi_i^* = 1 \} &= \max_{\pi_1, \dots, \pi_H} \sum_{i=1}^H v_i \mathbb{1} \{ \pi_i = 1 \} \\
&\text{s.t.} \quad \sum_{i=1}^H w_i \pi_i \leq W \\
&\quad \pi_i \in \{0, 1\}
\end{aligned} \tag{6}$$

Hence, the return of the optimal leader policy can be used to answer the Knapsack Decision problem, making optimal Stackelberg leader policy computation at least as hard as the Knapsack Decision problem. \square

We show that even with payment, it is still NP-Hard to compute the optimal policy in both trajectory and upfront payment settings. Note that in the proof that follows, we can scale the rewards such that $r(\cdot) \in [-1, 1]$ is satisfied, as the optimal policies remain unchanged under scaling.

Theorem A.2. *Under Markov games that are deterministic DAGs, it is NP-Hard to compute the optimal policy and optimal trajectory payment:*

$$\begin{aligned}
& \max_{\pi, b \geq 0} V^{\pi, \mu(\pi)}(s_0; r^L - \kappa \cdot b) \\
& \text{s.t.} \quad \mu(\pi) \in \operatorname{argmax}_{\mu'} V^{\pi, \mu'}(s_0; r^F + b)
\end{aligned}$$

and it is also NP-Hard to compute the optimal policy and optimal upfront payment:

$$\begin{aligned}
& \max_{\pi, b \geq 0} \left(V^{\pi, \mu(\pi)}(s_0; r^L) - \kappa \cdot \sum_{s, a, b \in S \times A \times B} b(s, a, b) \right) \\
& \text{s.t.} \quad \mu(\pi) \in \operatorname{argmax}_{\mu'} V^{\pi, \mu'}(s_0; r^F + b)
\end{aligned}$$

Proof. We will consider the same construction visualized in Figure 2 and for ease of presentation, we will show that the problem is NP-Hard under $\kappa = 1$ (e.g. the direct-payment settings).

Notation-wise, at the i th time step, let b_{i1}, b_{i4} be the leader payment in the left/right branch, b_{i2} be the total payment for the left leader path in the right branch and b_{i3} be the total payment on the right leader path in the right branch. Let $b_0 = b(s_0, a_L)$. And note that clearly any optimal payment is such that $b^*(s_0, a_R) = 0$, since the leader wishes to incentivize the follower to play a_L to obtain a positive return.

With this, we can write down the optimization program under both payment settings.

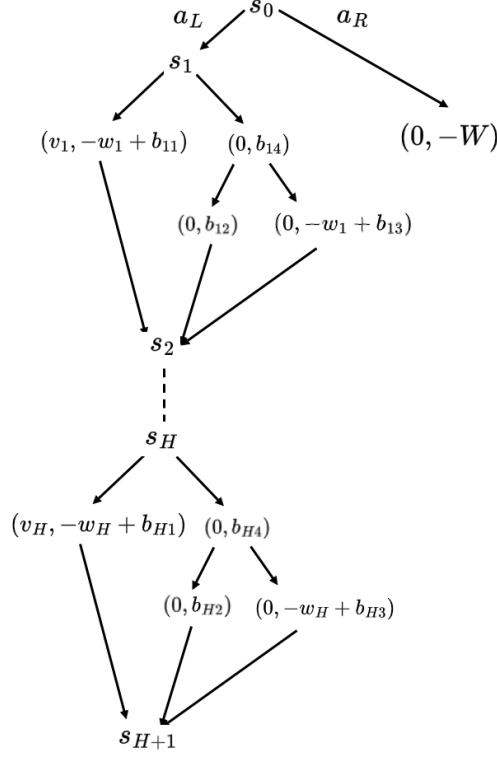


Figure 2: Same deterministic MDP with payment

- Under trajectory payment:

$$\begin{aligned}
 M_{trajectory}^* &= \max_{\pi_1, \dots, \pi_H, b} \sum_{i=1}^H v_i x_i - \sum_{i=1}^H b_{i1} x_i - \sum_{i=1}^H ((1 - \pi_i) b_{i2} + \pi_i (b_{i3} + b_{i4}) y_i - b_0 \\
 \text{s.t.} \quad &\sum_{i=1}^H (-w_i + b_{i1}) x_i + \sum_{i=1}^H ((1 - \pi_i) b_{i2} + \pi_i (-w_i + b_{i3}) + b_{i4}) y_i + b_0 \geq -W \\
 &x_i = \mathbb{1} \{ -w_i + b_{i1} \geq (1 - \pi_i) b_{i2} + \pi_i (-w_i + b_{i3}) + b_{i4} \}, y_i = 1 - x_i \\
 &b_{i1}, b_{i2}, b_{i3}, b_{i4}, b_0 \geq 0
 \end{aligned} \tag{7}$$

- Under upfront payment:

$$\begin{aligned}
 M_{upfront}^* &= \max_{\pi_1, \dots, \pi_H, b} \sum_{i=1}^H v_i x_i - \sum_{i=1}^H (b_{i1} + b_{i2} + b_{i3} + b_{i4}) x_i - \sum_{i=1}^H (b_{i1} + b_{i2} + b_{i3} + b_{i4}) y_i - b_0 \\
 \text{s.t.} \quad &\sum_{i=1}^H (-w_i + b_{i1}) x_i + \sum_{i=1}^H ((1 - \pi_i) b_{i2} + \pi_i (-w_i + b_{i3}) + b_{i4}) y_i + b_0 \geq -W \\
 &x_i = \mathbb{1} \{ -w_i + b_{i1} \geq (1 - \pi_i) b_{i2} + \pi_i (-w_i + b_{i3}) + b_{i4} \}, y_i = 1 - x_i \\
 &b_{i1}, b_{i2}, b_{i3}, b_{i4}, b_0 \geq 0
 \end{aligned} \tag{8}$$

Now, consider any optimal policy π^*, b^* . Let $x_i^*(\pi^*, b^*), y_i^*(\pi^*, b^*)$ denote the follower BR to π^*, b^* . Note that $x_i^*(\pi^*, b^*), y_i^*(\pi^*, b^*) = x_i^*(\pi_i^*, b_i^*), y_i^*(\pi_i^*, b_i^*)$.

Claim 1: We will show that there exists an optimal solution π', b' of the form:

1. If $y_i^*(\pi^*, b^*) = 1$, then $\pi'_i = 0, b'_{i1} = b'_{i3} = b'_{i4} = 0$.
2. If $x_i^*(\pi^*, b^*) = 1$, then $\pi'_i = 1, b'_{i2} = b'_{i3} = b'_{i4} = 0$.

We will construct a π', b' based on π^*, b^* that satisfy the desired two properties. The construction is as follows:

1. If $y_i^*(\pi^*, b^*) = 1$, we have two cases:

Case 1: If $(1 - \pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^* \geq 0$, set $\pi'_i = 0, b'_{i2} = (1 - \pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^*, b'_{i1} = b'_{i3} = b'_{i4} = 0$.

Firstly, $y'_i(\pi', b') = 1$ still since,

$$\begin{aligned} & -w_i + b'_{i1} \\ & \leq -w_i + b_{i1}^* \\ & < (1 - \pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^* \\ & = b'_{i2} = (1 - \pi'_i)b'_{i2} \end{aligned}$$

Thus, feasibly still holds since,

$$((1 - \pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^*)y_i = ((1 - \pi'_i)b'_{i2} + \pi'_i(-w_i + b'_{i3}) + b'_{i4})y_i$$

Finally, the trajectory payment objective value increases as total payment for i th step decreases from $(1 - \pi_i^*)b_{i2}^* + \pi_i^*b_{i3}^* + b_{i4}^*$ to $(1 - \pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^*$.

The upfront payment objective value increases as total payment for i th step decreases from $b_{i1}^* + b_{i2}^* + b_{i3}^* + b_{i4}^*$ to $(1 - \pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^*$.

Case 2: If $(1 - \pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^* < 0$, set $\pi'_i = 0$ and $b'_{i1} = b'_{i2} = b'_{i3} = b'_{i4} = 0$. $y'_i(\pi', b') = 1$ still since,

$$\begin{aligned} & -w_i + b'_{i1} \\ & \leq -w_i + b_{i1}^* \\ & < (1 - \pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^* \\ & < 0 = (1 - \pi'_i)b'_{i2} \end{aligned}$$

Thus, feasibly still holds since,

$$((1 - \pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^*)y_i < 0 \cdot 1 = ((1 - \pi'_i)b'_{i2} + \pi'_i(-w_i + b'_{i3}) + b'_{i4})y_i$$

Finally, the trajectory payment objective value increases as total payment for i th step decreases from $(1 - \pi_i^*)b_{i2}^* + \pi_i^*b_{i3}^* + b_{i4}^*$ to 0.

The upfront payment objective value increases as total payment for i th step decreases from $b_{i1}^* + b_{i2}^* + b_{i3}^* + b_{i4}^*$ to 0.

2. If $x_i^*(\pi^*, b^*) = 1$, set $\pi'_i = 1, b'_{i1} = b_{i1}^*$ and $b'_{i2} = b'_{i3} = b'_{i4} = 0$.

$x_i(\pi', b') = 1$ still because,

$$\begin{aligned}
& -w_i + b'_{i1} \\
& = -w_i + b^*_{i1} \\
& \geq (1 - \pi_i^*)b^*_{i2} + \pi_i^*(-w_i + b^*_{i3}) + b^*_{i4} \\
& \geq \pi_i^* \cdot -w_i \\
& \geq \pi_i' \cdot -w_i
\end{aligned}$$

Feasibility still holds since $(-w_i + b^*_{i1})x_i = (-w_i + b'_{i1})x_i$.

Finally, the trajectory payment objective value is unchanged as total payment for i th step is still b^*_{i1} .

The upfront payment objective value increases as total payment for i th step decreases from $b^*_{i1} + b^*_{i2} + b^*_{i3} + b^*_{i4}$ to b^*_{i1} .

Simplification of optimization programs:

From the above, we can introduce the constraints that define π', b' in the optimization programs without loss of optimality.

- The trajectory payment program with the constraints as in Claim 1 simplifies to:

$$\begin{aligned}
M^*_{trajectory} = \max_{\pi_1, \dots, \pi_H, b} \quad & \sum_{i=1}^H v_i x_i - \sum_{i=1}^H b_{i1} x_i - \sum_{i=1}^H b_{i2} y_i - b_0 \\
\text{s.t.} \quad & \sum_{i=1}^H (-w_i + b_{i1}) x_i + \sum_{i=1}^H b_{i2} y_i + b_0 \geq -W \\
& x_i = \mathbb{1} \{ \pi_i = 1 \}, y_i = \mathbb{1} \{ \pi_i = 0 \} \\
& b_{i1}, b_{i2}, b_0 \geq 0 \\
& \pi_i \in \{0, 1\}
\end{aligned} \tag{9}$$

- The upfront payment program with the constraints as in Claim 1 simplifies to:

$$\begin{aligned}
M^*_{upfront} = \max_{\pi_1, \dots, \pi_H, b} \quad & \sum_{i=1}^H v_i x_i - \sum_{i=1}^H b_{i1} x_i - \sum_{i=1}^H b_{i2} y_i - b_0 \\
\text{s.t.} \quad & \sum_{i=1}^H (-w_i + b_{i1}) x_i + \sum_{i=1}^H b_{i2} y_i + b_0 \geq -W \\
& x_i = \mathbb{1} \{ \pi_i = 1 \}, y_i = \mathbb{1} \{ \pi_i = 0 \} \\
& b_{i1}, b_{i2}, b_0 \geq 0 \\
& \pi_i \in \{0, 1\}
\end{aligned} \tag{10}$$

Reduction to PARTITION:

Since both programs are now the same under the constraints, we will call the optimal objective value $M^* = M^*_{trajectory} = M^*_{upfront}$.

$$\begin{aligned}
M^* = \max_{\pi_1, \dots, \pi_H, b} \quad & \sum_{i=1}^H v_i x_i - \sum_{i=1}^H b_{i1} x_i - \sum_{i=1}^H b_{i2} y_i - b_0 \\
\text{s.t.} \quad & \sum_{i=1}^H (-w_i + b_{i1}) x_i + \sum_{i=1}^H b_{i2} y_i + b_0 \geq -W \\
& x_i = \mathbb{1} \{ \pi_i = 1 \}, y_i = \mathbb{1} \{ \pi_i = 0 \} \\
& b_{i1}, b_{i2}, b_0 \geq 0 \\
& \pi_i \in \{0, 1\}
\end{aligned} \tag{11}$$

Write $B = \sum_{i=1}^H b_{i1} x_i + \sum_{i=1}^H b_{i2} y_i + b_0$ to simplify the program to:

$$\begin{aligned}
M^* = \max_{x_1, \dots, x_H, B} \quad & \sum_{i=1}^H v_i x_i - B \\
\text{s.t.} \quad & \sum_{i=1}^H w_i x_i \leq W + B \\
& x_i \in \{0, 1\} \\
& B \geq 0
\end{aligned} \tag{12}$$

Now consider Markov game instances where $v_i \in \mathbb{Z}^+$, $w_i = 2v_i$, $W = \sum_{i=1}^H v_i$.

$$\begin{aligned}
M^* = \max_{x_1, \dots, x_H, B} \quad & \sum_{i=1}^H v_i x_i - B \\
\text{s.t.} \quad & 2 \sum_{i=1}^H v_i x_i \leq W + B \\
& x_i \in \{0, 1\} \\
& B \geq 0
\end{aligned} \tag{13}$$

Claim 2: We can use this subset of the instances to answer the PARTITION decision problem for any PARTITION instance. We output YES if the objective value M^* of the optimal leader solution computed is exactly $M^* = W/2$ and NO otherwise.

To see this, we will show that $M^* = \frac{W}{2}$ if and only if there is a balanced partition.

(\Rightarrow) : Suppose x_1, \dots, x_H, B achieves M^* , we have $\sum_{i=1}^H v_i x_i - B \leq \frac{W}{2} - \frac{B}{2}$ from the feasibility condition. Therefore, for this to be $\frac{W}{2}$, it must be the case that $B = 0$, which implies that $\sum_{i=1}^H v_i x_i = \frac{W}{2}$. And $S = \{i : x_i = 1\}$ gives the balanced partition.

(\Leftarrow) : Given some balanced partition S , define $x_i = \mathbb{1} \{i \in S\}$ and $B = 0$. It is feasible because $2 \sum_{i=1}^H v_i x_i = 2 \cdot W/2 \leq W + 0$. And it achieves the optimal objective value $\frac{W}{2}$ because $M^* \leq W/2$: from the feasibility constraint, for all x 's and B , $\sum_{i=1}^H v_i x_i - B \leq \frac{W}{2} - \frac{B}{2}$.

□

Remark A.3. Scaling all parameters $\langle v_i, w_i, W \rangle$ to $\langle v_i/\kappa, w_i/\kappa, W/\kappa \rangle$ in the reduction to PARTITION allows us to show hardness for all $\kappa > 0$.

Corollary A.4. There exists no computationally efficient, no-regret learning algorithm.

Proof. Note that here M^* is an integer multiple of $1/2$. Suppose by contradiction that we had such an algorithm with regret T^α ($\alpha < 1$). We can then set T large enough such that $T^\alpha/T < 1/2$. This allows us to infer M^* exactly, thus giving us a computationally efficient algorithm for answering the PARTITION decision problem.

□

A.2 Positive Results

In terms of positive results, we show that there is a polynomial time algorithm for learning the optimal leader policy and payment, even in general-sum games. This holds when the MDP has tree structure.

Proposition A.5. *Under Markov games that are deterministic trees, there exists a polynomial-time planning algorithm that computes the optimal policy and payment.*

Proof. We know that the optimal leader policy and payment induce some root to leaf path τ^* . And so, it is sufficient to examine all possible root to leaf paths τ , which is efficient as there are $|S_H|$ root to leaf paths.

Characterizing policies with payment: For each path τ , let the set of leader policies, payment that realize τ be $\Gamma(\tau)$. We have the following characterization.

$\pi, b \in \Gamma(\tau)$ iff:

1. $\pi(s_i) = a_i \quad \forall s_i, a_i \in \tau$
2. $Q^{\pi, BR(\pi)}(s_j, a_j, b_j; r^F + b) \geq \max_{b'_j \neq b_j} Q^{\pi, BR(\pi)}(s_j, a_j, b'_j; r^F + b) \quad \forall s_j, a_j, b_j \in \tau$

The (\Leftarrow) direction is clear. And the (\Rightarrow) direction can be shown by proving the contrapositive. Indeed, if the first condition is not satisfied and $\pi(s_i) \neq a_i$, then $\pi, b \notin \Gamma(\tau)$. Or, if the second condition is not satisfied and $Q^{\pi, BR(\pi)}(s_j, a_j, b_j; r^F + b) < \max_{b'_j \neq b_j} Q^{\pi, BR(\pi)}(s_j, a_j, b'_j; r^F + b)$, then the follower would play $b'_j \neq b_j$ at s_j , which implies $\pi, b \notin \Gamma(\tau)$.

We are interested in the pair of policy, payment $\pi^*(\tau), b^*(\tau) \in \Gamma(\tau)$ that are the optimum of the following optimization program. Note that because $\pi^*(\tau), b^*(\tau) \in \Gamma(\tau)$, they realize τ , thus fixing the leader's return to $r^L(\tau)$. And so, $\pi^*(\tau), b^*(\tau)$ are such that they minimize the total payment needed to realize τ :

$$\begin{aligned}
 \pi^*, b^* = \operatorname{argmin}_{\pi, b \geq 0} \quad & \sum_{s_i \in S_i, a_i \in A, b_i \in B} b(s_i, a_i, b_i) \\
 \text{s.t.} \quad & Q^{\pi, BR(\pi)}(s_j, a_j, b_j; r^F + b) \geq \max_{b'_j \neq b_j} Q^{\pi, BR(\pi)}(s_j, a_j, b'_j; r^F + b) \quad \forall s_j, a_j, b_j \in \tau \\
 & \pi(s_i) = a_i \quad \forall s_i, a_i \in \tau
 \end{aligned} \tag{14}$$

Simplifying Optimization Program: Next, we make two observations that simplify the optimization program:

1. Since $\pi, b \in \Pi(\tau)$ and they realize τ , we have that:

$$Q^{\pi, BR(\pi)}(s_j, a_j, b_j; r^F + b) = r^F(\tau[j :]) + \sum_{i \geq j, s_i, a_i, b_i \in \tau} b(s_i, a_i, b_i)$$

where as the follower is only rewarded payment on the trajectory.

2. Next, we observe that $b^*(s_i, a_i, b_i) = 0$ for $s_i, b_i \notin \tau$.

If not, setting $b(s_i, a_i, b_i) = 0$ maintains feasibility, since it can only reduce the RHS in the constraints, while reducing the objective.

Thus, we can simplify the optimization program to:

$$\begin{aligned}
\pi^*, b^* = \operatorname{argmin}_{\pi, b \geq 0} \quad & \sum_{s_i \in S_i, a_i \in A, b_i \in B} b(s_i, a_i, b_i) \\
\text{s.t.} \quad & r^F(\tau[h :]) + \sum_{i \geq h, s_i, a_i, b_i \in \tau} b(s_i, a_i, b_i) \geq \max_{b'_h \neq b_h} Q^{\pi, BR(\pi)}(s_h, a_h, b'_h; r^F) \\
& \pi(s_i) = a_i \quad \forall s_i, a_i \in \tau
\end{aligned} \tag{15}$$

Let $\pi^- \in \operatorname{argmin}_{\pi} V^{\pi, BR(\pi)}(\cdot; r^F)$, which may be computed by Nash-VI in polynomial time. This minimax policy is such that $\pi^- \in \operatorname{argmin}_{\pi} V^{\pi, BR(\pi)}(s; r^F)$ for any state $s \in S$.

We claim that without loss of optimality, we can set $\pi^*(s'_i) = \pi^-(s'_i)$ for all states $s'_i \notin \tau$.

Given a pair of optimal solution (π^*, b^*) , let π' be such modification of a π^* .

We observe that (π', b^*) achieves the same objective value, while still being feasible. The former holds by construction as the payment remains unchanged.

$\pi'(s_i) = a_i \quad \forall s_i, a_i \in \tau$ holds still by construction. Now, feasibility holds because at any s_i , for any $b'_i \neq b_i$:

$$\begin{aligned}
& Q^{\pi^*, BR(\pi^*)}(s_i, a_i, b'_i; r^F) \\
& = V^{\pi^*, BR(\pi^*)}(s'_{i+1}; r^F) \quad (\text{state } s'_{i+1} = P(s_i, a_i, b'_i) \text{ deterministically}) \\
& \geq V^{\pi^-, BR(\pi^-)}(s'_{i+1}; r^F) \quad (\text{definition of } \pi^-) \\
& = V^{\pi', BR(\pi')}(s'_{i+1}; r^F) \\
& = Q^{\pi', BR(\pi')}(s_i, a_i, b'_i; r^F)
\end{aligned} \tag{*}$$

(*) : Due to the tree structure of the MDP, the set of successor states of s'_{i+1} does not contain any leader states in τ , as they belong to a different branch than the one τ and thus s_{i+1} are in, with the root at state s_i . And so, π' 's actions starting at state s'_{i+1} are exactly the same as that of π^- .

And so, π' is feasible because for every s_h ,

$$r^F(\tau[h :]) + \sum_{i \geq h, s_i, a_i, b_i \in \tau} b(s_i, a_i, b_i) \geq \max_{b'_h \neq b_h} Q^{\pi^*, BR(\pi^*)}(s_h, a_h, b'_h; r^F) \geq \max_{b'_h \neq b_h} Q^{\pi', BR(\pi')}(s_h, a_h, b'_h; r^F)$$

Minimal Payment LP: Since we have fully determined an optimal policy π^* , the optimal payment may be found by solving the following LP:

$$\begin{aligned}
b^* = \operatorname{argmin}_{b \geq 0} \quad & \sum_{s_i, a_i, b_i \in \tau} b(s_i, a_i, b_i) \\
\text{s.t.} \quad & r^F(\tau[h :]) + \sum_{i \geq h, s_i, a_i, b_i \in \tau} b(s_i, a_i, b_i) \geq \max_{b'_h \neq b_h} Q^{\pi, BR(\pi)}(s_h, a_h, b'_h; r^F) \\
& \pi(s_i) = a_i \quad \forall s_i, a_i \in \tau \\
& \pi(s'_i) = \pi^-(s'_i) \quad \forall s'_i \notin \tau
\end{aligned} \tag{16}$$

□

A.2.1 General sum learning in deterministic tree

Learning Setting: As a quick recap of the learning setting in the general-sum case, the only unknown is the follower rewards r^F . The MDP is deterministic and so the transitions are known. Also, the leader knows his own reward r^L .

Remark A.6. To complete the result, there is a simple exploration strategy using payment for general-sum, deterministic trees, as exploration needs to only recover rewards. This strategy allows us to reduce learning to planning, and then apply Algorithm 1.

Explore: To see this, for each root-to-leaf path $\tau = \{(s_i, a_i, b_i)\}_{i=1}^H$, set $b(s_i, a_i, b_i) = H$ for $s_i, a_i, b_i \in \tau$. Then, setting the leader policy to match τ will realize τ , as the follower is incentivized to play actions that realize this path. With this exploration strategy, we obtain estimates of the stochastic reward at every node of the tree \hat{r}^F to precision δ w.h.p. (better than $1 - 1/T$) after $m = \tilde{O}(|S| \cdot 1/\delta^2)$ number of episodes.

Exploit: Now, we bound the instantaneous regret when we plan using \hat{r}^F . This regret arises to due to the difference in payment computed by the LP, since r^L is known exactly. The LP admits closed form solution: $b^*(s_h, a_h, b_h) = \max(\max_{b'_h \neq b_h} Q^{\pi^-, \mu(\pi^-)}(s_h, a_h, b'_h; r^F) - \sum_{i \geq h, s_i, a_i, b_i \in \tau} r^F(s_i, a_i, b_i) - \sum_{i \geq h+1, s_i, a_i, b_i \in \tau} b^*(s_i, a_i, b_i), 0)$. Define $\hat{b}(s_h, a_h, b_h)$ analogously under \hat{r}^F and $\hat{\pi}^-$ (computed using \hat{r}^F).

Since the function $\max(x, 0)$ is 1-Lipschitz, we will bound the argument of the function. First, $|\sum_{i \geq h, s_i, a_i, b_i \in \tau} \hat{r}^F(s_i, a_i, b_i) - r^F(s_i, a_i, b_i)| \leq (H - h)\delta$.

Second, for any action b'_h , let $P(s'_{h+1} | s_h, a_h, b'_h) = 1$. We have that:

$$\begin{aligned} & Q^{\pi^-, \mu(\pi^-)}(s_h, a_h, b'_h; r^F) - Q^{\hat{\pi}^-, \mu(\hat{\pi}^-)}(s_h, a_h, b'_h; \hat{r}^F) \\ &= V^{\pi^-, \mu(\pi^-)}(s'_{h+1}; r^F) - V^{\hat{\pi}^-, \mu(\hat{\pi}^-)}(s'_{h+1}; \hat{r}^F) \\ &= \max_{\mu} V^{\pi^-, \mu}(s'_{h+1}; r^F) - \max_{\mu} V^{\pi^-, \mu}(s'_{h+1}; \hat{r}^F) + \max_{\mu} V^{\pi^-, \mu}(s'_{h+1}; \hat{r}^F) - \max_{\mu} V^{\hat{\pi}^-, \mu}(s'_{h+1}; \hat{r}^F) \\ &\geq \max_{\mu} V^{\pi^-, \mu}(s'_{h+1}; r^F) - \max_{\mu} V^{\pi^-, \mu}(s'_{h+1}; \hat{r}^F) + 0 \quad (\hat{\pi}^- \in \operatorname{argmin}_{\pi} \max_{\mu} V^{\pi, \mu}(s'_{h+1}; \hat{r}^F)) \\ &= \max_{\mu} V^{\mu}(s'_{h+1}; \pi^-, r^F) - \max_{\mu} V^{\mu}(s'_{h+1}; \pi^-, \hat{r}^F) \\ &\geq -\delta \\ &\text{(due to same visitation probability, any policy } \mu \text{'s return under } \pi^-, r^F \text{ vs } \pi^-, \hat{r}^F \text{ differs by } \leq \delta) \end{aligned}$$

The other direction follows analogously to get that: $|Q^{\pi^-, \mu(\pi^-)}(s_h, a_h, b'_h; r^F) - Q^{\hat{\pi}^-, \mu(\hat{\pi}^-)}(s_h, a_h, b'_h; \hat{r}^F)| \leq \delta$.

Therefore, we have that $|\hat{b}(s_h, a_h, b_h) - b^*(s_h, a_h, b_h)| \leq O((H - h)^2\delta)$. This means that the instantaneous regret due to $\sum_{s_i, a_i, b_i \in \tau} \hat{b}(s_i, a_i, b_i) - \sum_{s_i, a_i, b_i \in \tau} b^*(s_i, a_i, b_i) \leq O(H^3\delta)$. Choosing $\delta = T^{-1/3}$, we have that the cumulative regret is $O(\delta^{-2}) + O(TH^3\delta)$ respectively from the explore and exploit phase for a total of $O(T^{2/3})$ regret.

A.2.2 Computing Local Stackelberg Optimum

The computational hardness result does not preclude algorithms for other solution concepts. Due to the intractability of computing a global Stackelberg optimum, we may be interested in computing instead a local Stackelberg optimum. To this end, we derive a first order approach to this end, illustrating that this looser solution concept can be computed. We focus on trajectory-based payment below as the upfront payment gradient w.r.t. b for f is straightforward.

Value-based Penalty: We can use the existing idea of encoding the BR as Langrangian with value-based penalty [Shen et al., 2024]. To recap the notation for the policy-based method, leader policy has policy parameters denoted by x and payment b . Follower has policy with parameter y .

$$\min_{x, y, b} \underbrace{-V^{\pi_x, \pi_y}(s_0; r^L - b)}_{f(x, y, b)} + \lambda \underbrace{(-V^{\pi_x, \pi_y}(s_0; r^F + b) + \max_{y'} V^{\pi_x, \pi_{y'}}(s_0; r^F + b))}_{p(x, y, b)}$$

From [Shen et al., 2024], we have that,

$$\nabla_{x,b} p(x, b, y) = \nabla_{x,b} V_{M(x)}^{\pi_y}(\rho) + \nabla_{x,b} V_{M(x)}^{\pi_y}(\rho) \Big|_{\pi=\pi^*(x,b)}$$

where $\pi^*(x, b)$ is the optimal BR to leader policy x and b , and $M(x)$ is the single-agent MDP w.r.t. follower parameterized by leader policy π_x .

We will now describe the gradient component by component, as the overall gradient is the sum of Stackelberg game and reward shaping gradients:

1.

$$\nabla_x f(x, y, b) = -\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t Q_{\pi_y}^{\pi_x}(s_t, a_t; r^L - b) \nabla \log \pi_x(a_t | s_t) \mid s_0 \sim \rho, \pi_y, \pi_x\right]$$

2.

$$\nabla_y f(x, y, b) = -\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t Q_{\pi_x}^{\pi_y}(s_t, b_t; r^L - b) \nabla \log \pi_y(b_t | s_t) \mid s_0 \sim \rho, \pi_x, \pi_y\right]$$

3.

$$\nabla_b f(x, y, b) = -\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \nabla_b r(s_t, a_t, b_t; r^L - b) \mid s_0 \sim \rho, \pi_x, \pi_y\right]$$

4.

$$\begin{aligned} \nabla_x p(x, y, b) &= \nabla_x V_{\pi_x}^{\pi_y}(\rho) + \nabla_x V_{\pi_x}^{\pi_y}(\rho) \Big|_{\pi=\pi^*(x,b)} \\ &= -\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t (Q^{\pi_x, \pi_y}(s_t, a_t, b_t; r^f + b) - \tau h(\pi_x)) \nabla \log \pi_x(a_t | s_t) \mid s_0 \sim \rho, \pi_x, \pi_y\right] \\ &\quad + -\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t (Q^{\pi_x, \pi_y^*(x,b)}(s_t, a_t, b_t; r^f + b) - \tau h(\pi_x)) \nabla \log \pi_x(a_t | s_t) \mid s_0 \sim \rho, \pi_x, \pi_y^*(x, b)\right] \end{aligned}$$

5.

$$\nabla_y p(x, y, b) = \nabla_y V_{\pi_x}^{\pi_y}(\rho) = -\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t Q_{\pi_x}^{\pi_y}(s_t, b_t; r^F + b) \nabla \log \pi_y(b_t | s_t) \mid s_0 \sim \rho, \pi_x, \pi_y\right]$$

6.

$$\begin{aligned} \nabla_b p(x, y, b) &= \nabla_b V_{\pi_x}^{\pi_y}(\rho) + \nabla_{x,b} V_{\pi_x}^{\pi_y}(\rho) \Big|_{\pi=\pi^*(x,b)} \\ &= -\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \nabla_b r(s_t, a_t, b_t; r^F + b) \mid s_0 \sim \rho, \pi_x, \pi_y\right] \\ &\quad + -\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \nabla_b r(s_t, a_t, b_t; r^F + b) \mid s_0 \sim \rho, \pi_x, \pi_y^*(x, b)\right] \end{aligned}$$

B Proofs for Learning Results in Cooperative Games without Payment

From the hardness of computing the optimal policy in the general sum setting, we know that this translates to the computational hardness of learning. On the other hand, we know that planning in cooperative games is efficient. This begs the questions, is learning in cooperative games efficient?

We delve into this question by first considering cooperative games without subsidy, as this has been previously unaccounted for in prior literature. Surprisingly, we find that learning in Cooperative Markov games can be prohibitively hard, statistically.

Lemma B.1. *Suppose an algorithm A tries to identify $a^* \in \{a_1, \dots, a_n\}$. Each step, it can make a query a_i and receive signal $\mathbb{1}\{a_i = a^*\}$. Then, for any possibly randomized algorithm A with query budget m . If $m \leq n/4$, then:*

$$\Pr_{a^* \in \text{unif}([n]), A} (a^* \notin \{a_1, \dots, a_m\}) \geq \frac{1}{2}$$

Proof. Let a^* be drawn uniformly from $[n]$. Let a_1, \dots, a_m be queries made by algorithm. And let $Y_i = \mathbb{1}\{a_i = a^*\}$. We will show by induction that for $j \leq m$,

$$\Pr(Y_{:j} \neq 0) = \Pr(a^* \in \{a_1, \dots, a_j\}) \leq \frac{2j}{n}$$

Base Case: Since a_1 is independent of a^* , $a^*|a_1$ is uniform over $[n]$. Thus,

$$\Pr(a^* = a_1) = \frac{1}{n} \leq \frac{2}{n}$$

Induction Step: Suppose $\Pr(a^* \in \{a_1, \dots, a_{j-1}\}) \leq \frac{2(j-1)}{n}$, we have:

$$\begin{aligned} & \Pr(a^* \in \{a_1, \dots, a_j\}) \\ &= \Pr(a^* \in \{a_1, \dots, a_{j-1}\}) + P(a^* = a_j, a^* \notin \{a_1, \dots, a_{j-1}\}) \\ &\leq \frac{2(j-1)}{n} + P(a^* = a_j, a^* \notin \{a_1, \dots, a_{j-1}\}) \\ &\leq \frac{2(j-1)}{n} + P(a^* = a_j | a^* \notin \{a_1, \dots, a_{j-1}\}) \\ &= \frac{2(j-1)}{n} + \mathbb{E}[P(a^* = a_j | a^* \notin \{a_1, \dots, a_{j-1}\}) | a_1, \dots, a_{j-1}] \\ &= \frac{2(j-1)}{n} + \frac{1}{n - (j-1)} \tag{★} \\ &\leq \frac{2(j-1)}{n} + \frac{2}{n} \tag{j \leq n/4} \end{aligned}$$

(★) : for any fixed a_1, \dots, a_{j-1} , conditioned on $a^* \notin \{a_1, \dots, a_{j-1}\}$, a^* is uniform over $[n] \setminus \{a_1, \dots, a_{j-1}\}$. Thus,

$$P(a^* = a_j | a^* \notin \{a_1, \dots, a_{j-1}\}) \leq \frac{1}{n - (j-1)}$$

In conclusion,

$$P(a^* \in \{a_1, \dots, a_m\}) \leq 2j/n \leq 1/2 \Rightarrow P(a^* \notin \{a_1, \dots, a_m\}) \geq 1/2$$

where we use that $j \leq m/4 \Rightarrow 2j/n \leq 1/2$.

□

Theorem B.2. *There exists a turn-based Stochastic Tree Markov game such that: any (possibly randomized) algorithm that returns the optimal leader policy with probability at least $1/2$ requires at least $\Omega(2^{|S|})$ number of episodes.*

Proof. Setup: Consider a two-branch MDP, where the follower chooses first action a_L or a_R , which deterministically transitions to the left and right branch. The leader has return $1 - 1/2|S|$ for the left branch. In the right branch, this transitions with uniform probability $1/|S|$ to one of $|S|$ possible states, each state has two possible actions with reward 0 and 1.

Then, we see that each time step, the leader will choose right branch policy $(\pi_t(s_1), \dots, \pi_t(s_m))$, and receive feedback $\mathbb{1}\{\pi_t = \pi^*\} = \mathbb{1}\{\mu(\pi_t) = a_R\}$, as $\mu(\pi_t) = a_R \Leftrightarrow r(\pi_t) \geq 1 - 1/2|S| \Leftrightarrow r(\pi_t) = 1 \Leftrightarrow \pi_t = \pi^*$.

To finish, we may use Lemma B.1 to get that any algorithm with budget at most $T \leq |\Pi|/4$ will be s.t. $P_A(\pi^* \in \{\pi_1, \dots, \pi_T\}) \leq 1/2$. Hence, using the contrapositive, any algorithm such that $P_A(\pi^* \in \{\pi_1, \dots, \pi_T\}) > 1/2$ must have budget $T \geq |\Pi|/4 + 1 = 2^{|S|}/4 + 1 = \Omega(2^{|S|})$. \square

Theorem B.3. *There exists a turn-based Deterministic DAG Markov game such that: any (possibly randomized) algorithm that returns the optimal leader policy with probability at least $1/2$ requires at least $\Omega(2^{|H|})$ number of episodes.*

Proof. Setup: Consider a two-branch MDP, where the follower chooses first action a_L or a_R , which deterministically transitions to the left and right branch. The leader has return $H - 1/2$ for the left branch. In the right branch, there are two actions at each of the H time steps with reward 0 and 1.

Then, we see that each time step, the leader will choose right branch policy $(\pi_t(s_1), \dots, \pi_t(s_n))$, and receive feedback $\mathbb{1}\{\pi_t = \pi^*\} = \mathbb{1}\{\mu(\pi_t) = a_R\}$, as $\mu(\pi_t) = a_R \Leftrightarrow r(\pi_t) \geq H - 1/2 \Leftrightarrow r(\pi_t) = H \Leftrightarrow \pi_t = \pi^*$.

To finish, we may use the lemma above to get that any algorithm with budget at most $T \leq |\Pi|/4$ will be s.t. $P_A(\pi^* \in \{\pi_1, \dots, \pi_T\}) \leq 1/2$. Hence, using the contrapositive, any algorithm such that $P_A(\pi^* \in \{\pi_1, \dots, \pi_T\}) > 1/2$ must have budget $T \geq |\Pi|/4 + 1 = 2^H/4 + 1 = \Omega(2^H)$. \square

Algorithm 3 Learning Algorithm for Deterministic Tree Markov Game without Payment

for all root to leaf paths $\tau = s_1, a_1, b_1, \dots, s_H, a_H, b_H$ **do**
 Define $\pi(s_i) = a_i$ for $s_i, a_i \in \tau$, set $\pi(s)$ to any arbitrary action in states not on τ
 Commit to π and observe if τ is realized by $\mu(\pi)$
 If τ is realized, apply π m times and record estimated return $\hat{r}^L(\tau)$
end for
Return the leader policy that has realized τ and has the maximal $\hat{r}^L(\tau)$

Learning Setting: As a quick recap of the learning setting in the cooperative case, the only unknown is the leader rewards r^L (the same as that of the follower) The MDP is deterministic and so the transitions are known.

Proposition B.4. *Under Markov games that are deterministic trees, then there exists a polynomial-time algorithm that can learn a near-optimal leader policy.*

Proof. We know that there is some root to leaf path τ^* such that $\tau^* \in \operatorname{argmax}_{\tau} r^L(\tau)$. Our goal is to search for a leader policy that realizes τ^* . Note that it is sufficient to simply find τ^* , as any leader policy that matches τ^* will induce τ^* as the follower's reward is the same as that of the leader.

To find τ^* , it is sufficient to iterate through all possible root to leaf paths τ . There are $|S_H|$ many policies, and so this can be done in polynomial time. For each path τ , choose any leader policy π with $\pi(s_i) = a_i$ for $s_i, a_i \in \tau$. If τ is not realized by $\pi, \mu(\pi)$, then this means $r^L(\tau)$ must be dominated by a different path's $r^L(\tau')$. And so, $\tau \neq \tau^*$. Thus, after iterating through all possible τ 's, we must have observed τ^* among the paths realized.

We can then identify a near-optimal policy by choosing the policy, whose path $\hat{\tau}$ has the maximal $\hat{r}^L(\cdot)$. This gives us a near-optimal policy with high probability (e.g. better than $1 - 1/T$), as after $m = \tilde{O}(1/\delta^2)$ number of episodes, we have estimated every potential optimal path return $\hat{r}^L(\tau)$ to precision δ w.h.p. And so, our returned policy's return (i.e. $\hat{r}^L(\tau)$) must be sub-optimal by at most δ . Please see Algorithm 3 for the algorithm. \square

Finally, we illustrate that this statistical hardness is surprisingly difficult to overcome, even when we relax the BR nature of the follower.

Theorem B.5. *There exists a turn-based Deterministic DAG Markov game such that: any (possibly randomized) algorithm that outputs the optimal policy given λ -Entropy-regularized BR with probability at least $1/2$ requires at least $\Omega(\exp(\lambda^2 H/8))$ episodes if $\lambda \leq 1$ and $\Omega(\exp(H/8))$ episodes if $\lambda > 1$.*

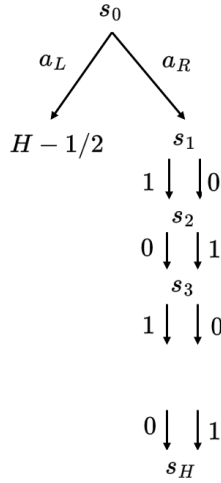


Figure 3: Right branch requires getting the right set of H binary actions at states s_1, \dots, s_H to exceed the return in the left branch.

Proof. To show this, we will use Yao's lemma. Let \mathcal{D} be the uniform distribution over all MDP instances \mathcal{X} , where the distribution is uniform over which of the two actions at each time step achieves reward 1 (and the other 0). Define $\lambda' = \min(\lambda, 1/\lambda)$ and let:

$$m = \min(\exp(\lambda \lambda' H/8), \exp(H/8))/16$$

Define $\text{cost}(A, t, x) = \mathbb{1}(a^* \notin \{a_1, \dots, a_t\})$, which is the probability of A not outputting the optimal policy a^* after t episodes.

We will show that for any deterministic algorithm A , if $t \leq m$, with probability at least $\frac{1}{2}$ over the choice of $x \sim \mathcal{D}$, $\text{cost}(A, t, x) = 1$. That is, $\min_A \mathbb{E}_{x \sim \mathcal{D}}[\text{cost}(A, t, x)] \geq \frac{1}{2}$, which by Yao's lemma means that any randomized algorithms R has $\min_R \max_{x \in \mathcal{X}} \mathbb{E}[\text{cost}(R, t, x)] \geq \frac{1}{2}$.

For some optimal code a^* , let the set $S_{H/4}(a^*)$ be all length- H binary combinations with Hamming distance at most $H/4$ or at least $3H/4$ from a^* . Let $S'_{H/4}(a^*)$ be its complement i.e. the set of codes with Hamming distance between $H/4$ and $3H/4$ with respect to a^* .

First, for any a^* , we bound the cardinality of the set $S_{H/4}(a^*)$: $|S_{H/4}(a^*)| = 2(\binom{H}{H} + \dots + \binom{H}{3H/4}) = 2(\binom{H}{0} + \dots + \binom{H}{H/4})$. And $2(\sum_{i=0}^{1/4 \cdot H} \binom{H}{i}) \leq 2 \cdot 2^H \exp(-2H(1/2 - 1/4)^2) \Rightarrow \frac{|S_{H/4}(a^*)|}{2^H} \leq 2 \exp(-H/8)$.

In round t , let the leader's chosen binary code be a_t , and we observe one realization of the random variable $b_t \sim \sigma(\lambda \cdot (d_L(a_t, a^*) - 1/2))$, which denotes whether the follower chooses to go left in the BR (1 indicates left).

Lemma B.6. For each $t \leq m$, we have that for any chosen code a_t :

$$\Pr(a_t \notin S_{H/4}(a^*) | b_{:t-1} = 1) \geq 1 - \frac{6}{\exp(H/8)}$$

Proof. Let us define $M = \frac{\lambda' H}{8}$ and $T = 1 + \exp(\lambda M)$. We will prove the result holds for any $t \leq T$. This is sufficient as $T \geq \exp(\lambda \lambda' H/8) \geq m$.

We will show a stronger result that for any fixed sequence $a_{:t-1}$, we have that:

$$\Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1, a_{:t-1}) > 1 - \frac{6}{\exp(H/8)}$$

And so,

$$\Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1) = \mathbb{E}[\Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1, a_{:t-1}) | a_{:t-1}] > 1 - \frac{6}{\exp(H/8)}$$

Fix any sequence of $a_{:t-1}$ and observed $b_{:t-1} = 1$, first we note that the posterior is of the form:

$$\begin{aligned} P(a^* = a | b_{:t-1} = 1, a_{:t-1}) \\ &\propto P(a^* = a) \prod_{i=1}^{t-1} P(b_i = 1 | a^* = a, b_{:i-1} = 1, a_{:t-1}) \\ &= P(a^* = a) \prod_{i=1}^{t-1} \sigma(\lambda(d_L(a, a_i) - 1/2)) \quad (b_i \text{ is a function of only } a^*, a_i) \end{aligned}$$

Next, since the algorithm is deterministic, a_t is deterministic with given $a_{:t-1}$ and $b_{:t-1}$:

$$\begin{aligned} &\Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1, a_{:t-1}) \\ &= \sum_a \Pr(a_t \in S'_{H/4}(a^*) | a^* = a, b_{:t-1} = 1, a_{:t-1}) \Pr(a^* = a | b_{:t-1} = 1, a_{:t-1}) \\ &= \sum_a \mathbb{1} \left\{ a \in S'_{H/4}(a_t) \right\} \Pr(a^* = a | b_{:t-1} = 1, a_{:t-1}) \\ &= \sum_{a \in S'_{H/4}(a_t)} \Pr(a^* = a | b_{:t-1} = 1, a_{:t-1}) \end{aligned}$$

It suffices to lower bound the posterior probability mass over $S'_{H/4}(a_t)$: $\sum_{a \in S'_{H/4}(a_t)} P(a^* = a | b_{:t-1} = 1, a_{:t-1})$.

For any chosen code a_i for $i \in [t-1]$, consider all codes up to M hamming distance away from a_i . There are $\sum_{i=0}^M \binom{H}{i}$ such codes. This means there are at most $(t-1) \sum_{i=0}^M \binom{H}{i}$ codes with at least one posterior update factor less than $\sigma(\lambda(M-1/2))$.

This implies the remaining set of codes a in $S'_{H/4}(a_t)$ has likelihood factor of at least:

$$\begin{aligned}
& \prod_{i=1}^{t-1} \sigma(\lambda(d_L(a, a_i) - 1/2)) \\
& \geq \prod_{i=1}^{t-1} \sigma(\lambda(M - 1/2)) \\
& \geq (1 - \frac{1}{1 + \exp(\lambda M)})^{t-1} \\
& \geq (1 - \frac{1}{1 + \exp(\lambda M)})^T \geq 1/e \quad (T = 1 + \exp(\lambda M))
\end{aligned}$$

Let N_t be the normalizing factor at episode t . Then, the posterior is such that:

$$\begin{aligned}
& \sum_{a \in S'_{H/4}(a_t)} \Pr(a^* = a | b_{:t-1} = 1, a_{:t-1}) \\
& \geq \frac{1}{N_t} (2^H - (|S_{H/4}(a_t)| + (t-1) \cdot \sum_{i=1}^M \binom{H}{i})) \cdot 1/e \\
& \geq \frac{1}{N_t} (2^H - (|S_{H/4}(a_t)| + T \cdot \sum_{i=1}^M \binom{H}{i})) \cdot 1/e \\
& \geq \frac{1}{N_t} (2^H (1 - 2 \exp(-H/8) - (1 + \exp(\lambda M)) \exp(-2H(1/2 - M/H)^2)) \cdot 1/e \\
& \geq \frac{1}{N_t} \cdot 2^H (1 - 2 \exp(-H/8) - 2 \exp(\lambda M - 2H(1/2 - 1/8)^2)) \cdot 1/e \quad (M/H \leq 1/8) \\
& \geq \frac{1}{N_t} \cdot 2^H (1 - 2 \exp(-H/8) - 2 \exp(-10H/64)) \cdot 1/e \quad (\lambda M \leq H/8)
\end{aligned}$$

On the other hand, we also have that:

$$\begin{aligned}
& \Pr(a_t \in S_{H/4}(a^*) | b_{:t-1} = 1, a_{:t-1}) \\
& = \sum_{a \in S_{H/4}(a_t)} \Pr(a^* = a | b_{:t-1} = 1, a_{:t-1}) \\
& \leq \frac{1}{N_t} \cdot (2^H 2 \exp(-H/8)) \cdot 1
\end{aligned}$$

Therefore, for any $t \leq T$, the probability of selecting a code with return between $H/4$ and $3H/4$ is at least:

$$\begin{aligned}
& \Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1, a_{:t-1}) \\
& = 1 - \frac{1}{1 + \Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1, a_{:t-1}) / \Pr(a_t \in S_{H/4}(a^*) | b_{:t-1} = 1, a_{:t-1})} \\
& \geq 1 - \frac{1}{1 + \frac{1 - 2 \exp(-H/8) - 2 \exp(-10H/64)}{e[2 \exp(-H/8)]}} \\
& \geq 1 - \frac{1}{1 + \exp(H/8)/2e - 2/e} \\
& > 1 - \frac{6}{\exp(H/8)}
\end{aligned}$$

□

Then, we have that for any $t \leq m$:

$$\begin{aligned}
& \Pr(b_t = 1 | b_{:t-1} = 1) \\
& \geq \Pr(b_t = 1 | a_t \in S'_{H/4}(a^*), b_{:t-1} = 1) \Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1) \\
& \geq \frac{1}{1 + \exp(-\lambda[(H - 1/2) - (3H/4 - 1)])} \Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1) \\
& \geq (1 - \frac{1}{1 + \exp(\lambda H/4)}) \Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1) \\
& > 1 - \frac{6}{\exp(H/8)} - \frac{1}{1 + \exp(\lambda H/4)} \quad (\text{using the Lemma above})
\end{aligned}$$

With this, we have that for any algorithm A with output a_i at episode i :

$$\begin{aligned}
& \mathbb{E}_{x \sim \mathcal{D}}[\text{cost}(A, t, x)] \\
& = P(a^* \notin \{a_1, \dots, a_t\}) \\
& \geq P(\bigcup_{i \in [t]} a_i \in S'_{H/4}(a^*)) \\
& \geq P(\bigcup_{i \in [t]} a_i \in S'_{H/4}(a^*), b_{:t} = 1) \\
& = P(b_t = 1 | \bigcup_{i=1}^t a_i \in S'_{H/4}(a^*), b_{:t-1} = 1) P(a_t \in S'_{H/4}(a^*) | \bigcup_{i=1}^{t-1} a_i \in S'_{H/4}(a^*), b_{:t-1} = 1) \\
& \quad P(\bigcup_{i=1}^{t-1} a_i \in S'_{H/4}(a^*), b_{:t-1} = 1) \\
& \geq \frac{1}{1 + \exp(-\lambda[(H - 1/2) - (3H/4 - 1)])} P(a_t \in S'_{H/4}(a^*) | \bigcup_{i=1}^{t-1} a_i \in S'_{H/4}(a^*), b_{:t-1} = 1) \\
& \quad P(\bigcup_{i=1}^{t-1} a_i \in S'_{H/4}(a^*), b_{:t-1} = 1) \quad (\text{using that } a_t \in S'_{H/4}(a^*) \text{ and } b_t \text{ is a function of only } a_t) \\
& \geq (1 - \frac{1}{1 + \exp(\lambda H/4)}) (1 - \frac{6}{\exp(H/8)}) P(\bigcup_{i=1}^{t-1} a_i \in S'_{H/4}(a^*), b_{:t-1} = 1) \\
& \quad (\Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1, a_{:t-1}) \geq 1 - \frac{6}{\exp(H/8)} \text{ for any fixed sequence } a_{:t-1}) \\
& \geq (1 - \frac{1}{1 + \exp(\lambda H/4)} - \frac{6}{\exp(H/8)})^t \quad (\text{unrolling}) \\
& \geq 1 - \frac{8t}{\min(\exp(H/8), \exp(\lambda H/4))} \\
& \geq 1/2 \quad (t \leq m)
\end{aligned}$$

□

C Proofs for Learning Results in Cooperative Games with Payment

Due to the statistical hardness of learning without payment, we now consider whether payment can alleviate this hardness. Note that unlike prior works in bandits [Scheid et al., 2024], we can no longer exhaustively enumerate all possible leader policies, which is feasible only in the bandit setting. We main approach is to adopt the natural idea of setting the outcome-based payments be the bonuses, in order to align incentives during exploration.

C.1 General regret guarantees

We first prove two results that are used in all the following regret proofs. The first informs us what is the optimal policy and payment to compete against in the cooperative case.

Lemma C.1. *For any π^*, b^* such that:*

$$\begin{aligned} \pi^*, b^* &= \operatorname{argmax}_{\pi, b} V^{\pi, \mu(\pi; r^F + b)}(s_0; r^L - \kappa \cdot b) \\ \text{s.t. } \mu(\pi; r^F + b) &\in \operatorname{argmax}_{\mu'} V^{\pi, \mu'}(s_0; r^F + b) \end{aligned} \quad (17)$$

If $r^L = r^F$, then we must have $\pi^, \cdot = \operatorname{argmax}_{\pi, \mu} V^{\pi, \mu}(s_0; r^L)$ and $b^* = 0$.*

Proof. We show that for π^* such that it is part of a globally optimal pair $\pi^*, \mu^* = \operatorname{argmax}_{\pi, \mu} V^{\pi, \mu}(s_0; r^L)$.

We claim that $(\pi^*, 0)$ dominates every pair (π, b) :

$$\begin{aligned} &V^{\pi^*, \mu(\pi^*; r^F)}(s_0; r^L - 0) \\ &= V^{\pi^*, \mu^*}(s_0; r^L - 0) \quad (r^F = r^L \Rightarrow \mu^* \text{ is a BR to } \pi^*) \\ &\geq V^{\pi, \mu(\pi; r^F + b)}(s_0; r^L) \\ &\text{(joint optimality } \pi^*, \mu^* = \operatorname{argmax}_{\pi, \mu} V^{\pi, \mu}(s_0; r^L) \text{ implies } \pi^*, \mu^* \text{ dominates } \pi, \mu(\pi; r^F + b)) \\ &\geq V^{\pi, \mu(\pi; r^F + b)}(s_0; r^L - \kappa \cdot b) \end{aligned}$$

Similarly for upfront payment:

$$\begin{aligned} &V^{\pi^*, \mu(\pi^*; r^F)}(s_0; r^L) \\ &= V^{\pi^*, \mu^*}(s_0; r^L) \quad (r^F = r^L \Rightarrow \mu^* \text{ is a BR to } \pi^*) \\ &\geq V^{\pi, \mu(\pi; r^F + b)}(s_0; r^L) \\ &\text{(joint optimality } \pi^*, \mu^* = \operatorname{argmax}_{\pi, \mu} V^{\pi, \mu}(s_0; r^L) \text{ implies } \pi^*, \mu^* \text{ dominates } \pi, \mu(\pi; r^F + b)) \\ &\geq V^{\pi, \mu(\pi; r^F + b)}(s_0; r^L) - \kappa \cdot \sum_{s, a, b} b(s, a, b) \end{aligned}$$

□

Lemma C.2. *Suppose we can construct an optimistic MDP M_k of the true MDP M . Let the optimal leader policy under M_k be π_k , then:*

$$\sum_{k=1}^T V_M^{\pi^*, \mu_M(\pi^*)}(s_0; r^L) - V_M^{\pi_k, \mu_M(\pi_k)}(s_0; r^L) \leq \sum_{k=1}^T V_{M_k}^{\pi_k, \mu_{M_k}(\pi_k)}(s_0; r^L) - V_M^{\pi_k, \mu_{M_k}(\pi_k)}(s_0; r^L)$$

Proof. By optimality of π_k in M_k and optimism, we have that,

$$V_{M_k}^{\pi_k, \mu_{M_k}(\pi_k)} \geq V_{M_k}^{\pi^*, \mu_M(\pi^*)} \geq V_M^{\pi^*, \mu_M(\pi^*)}$$

Therefore, we may bound the instantaneous regret,

$$\begin{aligned}
& \sum_{k=1}^T V_M^{\pi^*, \mu_M(\pi^*)} - V_M^{\pi^k, \mu_M(\pi^k)} \\
& \leq \sum_{k=1}^T V_{M_k}^{\pi^k, \mu_{M_k}(\pi^k)} - V_M^{\pi^k, \mu_M(\pi^k)} \quad (\text{Optimism}) \\
& \leq \sum_{k=1}^T V_{M_k}^{\pi^k, \mu_{M_k}(\pi^k)} - V_M^{\pi^k, \mu_{M_k}(\pi^k)} \quad (\text{BR means } V_M^{\pi^k, \mu_M(\pi^k)} \geq V_M^{\pi^k, \mu_{M_k}(\pi^k)})
\end{aligned}$$

□

In the proofs that follow, we make use of the classic UCB-VI algorithm as the no-regret RL algorithm [Azar et al., 2017]. Note that although UCB-VI is applicable when the rewards are in $[0, 1]$, with translation and scaling, regret guarantees of the same order still hold when rewards are in $[-1, 1]$.

Theorem C.3. *UCB-VI-FP (Algorithm 2) incurs $O(T^{1/2})$ regret under trajectory payment. This is tight as there exists a subset of Markov games, where any learning algorithm must incur $\Omega(T^{1/2})$ regret.*

Proof. At episode k , we have three MDPs:

1. The ground truth MDP $M = (P, r)$
2. An empirical MDP $\hat{M}^k = (\hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k)$ with $\beta_h^k = 2c\sqrt{\frac{SH^2}{N_h^k(s, a, b)}}$.
3. The subsidy MDP $\tilde{M}^k = (P, r + \beta^k)$

Optimism: We want to show that:

1. \hat{M}^k is optimistic wrt M : for all π, μ ,

$$V^{\pi, \mu}(s_0; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k) \geq V^{\pi, \mu}(s_0; P, r)$$

2. \tilde{M}^k is optimistic wrt \hat{M}^k : for all π, μ ,

$$V^{\pi, \mu}(s_0; P, r + \beta^k) \geq V^{\pi, \mu}(s_0; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k)$$

To do this, fix some (s, a, b) , step h and policies π, μ . Then, we have from total-variation concentration bound for multinomial distribution:

$$\|\hat{P}^k(\cdot | s, a, b) - P(\cdot | s, a, b)\|_1 \leq c' \sqrt{\frac{|S|}{N_h^k(s, a, b)}}$$

Next, we have

$$(\hat{P}_h^k - P_h)V_{h+1}^{\pi, \mu}(s, a, b; \cdot) = \sum_{s'} (\hat{P}^k(s' | s, a, b) - P(s' | s, a, b))V_{h+1}^{\pi, \mu}(s'; \cdot).$$

Applying Hölder's inequality (with $p = 1, q = \infty$):

$$|(\hat{P}_h^k - P_h)V_{h+1}^{\pi, \mu}(s, a, b; \cdot)| \leq \|\hat{P}^k(\cdot | s, a, b) - P(\cdot | s, a, b)\|_1 \|V_{h+1}^{\pi, \mu}\|_\infty \leq c' \sqrt{\frac{|S|}{N_h^k(s, a, b)}} \cdot H.$$

Then by Hoeffding, this means there exists c such that:

$$|(\hat{P}_h^k - P_h)V_{h+1}^{\pi, \mu}(s, a, b; \cdot)| + |\bar{r}_h(s, a, b) - r_h(s, a, b)| \leq c \sqrt{\frac{|S|}{N_h^k(s, a, b)}} \cdot H.$$

1. Optimism of \hat{M}^k :

Using our preceding bound, we have that for all h, s, a, b :

$$\bar{r}_h^k(s, a, b) + \frac{1}{2}\beta_h^k(s, a, b) \geq r_h(s, a, b) + |(\hat{P}_h^k - P_h)V_{h+1}^{\pi, \mu}(s, a, b; \cdot)|.$$

We will use this in backward induction on $h \in [H]$ to show optimism:

Base Case $h = H + 1$: $V_h^{\pi, \mu}(s; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k) = 0 = V_h^{\pi, \mu}(s; P, r)$.

Induction Step: suppose we have that for all state s :

$$V_{h+1}^{\pi, \mu}(s; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k) \geq V_{h+1}^{\pi, \mu}(s; P, r)$$

It is sufficient to show that for every s, a, b :

$$Q_h^{\pi, \mu}(s, a, b; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k) \geq Q_h^{\pi, \mu}(s, a, b; P, r).$$

and the result follows from the inequality with $\max_{a,b}$ applied on both sides.

Using induction hypothesis:

$$\begin{aligned} Q_h^{\pi, \mu}(s, a, b; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k) &= \bar{r}_h^k(s, a, b) + \frac{1}{2}\beta_h^k(s, a, b) + \sum_{s'} \hat{P}_h^k(s'|s, a, b) V_{h+1}^{\pi, \mu}(s'; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k) \\ &\geq \bar{r}_h^k(s, a, b) + \frac{1}{2}\beta_h^k(s, a, b) + \sum_{s'} \hat{P}_h^k(s'|s, a, b) V_{h+1}^{\pi, \mu}(s'; P, r) \\ &\geq r_h(s, a, b) + \sum_{s'} P_h(s'|s, a, b) V_{h+1}^{\pi, \mu}(s'; P, r) \\ &= Q_h^{\pi, \mu}(s, a, b; P, r). \end{aligned}$$

2. Optimism of \tilde{M}^k w.r.t. \hat{M}^k :

Using our preceding bound, we have that for all h, s, a, b :

$$\bar{r}_h^k(s, a, b) + \frac{1}{2}\beta_h^k(s, a, b) \geq r_h(s, a, b) + |(\hat{P}_h^k - P_h)V_{h+1}^{\pi, \mu}(s, a, b; \cdot)|.$$

We will use this in backward induction on $h \in [H]$ to show optimism:

Base Case $h = H + 1$: $V_h^{\pi, \mu}(s; P, r + \beta^k) = 0 = V_h^{\pi, \mu}(s; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k)$.

Induction Step: suppose we have that for all state s :

$$V_{h+1}^{\pi, \mu}(s; P, r + \beta^k) \geq V_{h+1}^{\pi, \mu}(s; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k)$$

It is sufficient to show that for every s, a, b :

$$Q_h^{\pi, \mu}(s, a, b; P, r + \beta^k) \geq Q_h^{\pi, \mu}(s, a, b; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k)$$

Using induction hypothesis:

$$\begin{aligned}
Q_h^{\pi, \mu}(s, a, b; P, r + \beta^k) &= r_h(s, a, b) + \beta_h^k(s, a, b) + \sum_{s'} P_h(s'|s, a, b) V_{h+1}^{\pi, \mu}(s'; P, r + \beta^k) \\
&\geq r_h(s, a, b) + \beta_h^k(s, a, b) + \sum_{s'} P_h(s'|s, a, b) V_{h+1}^{\pi, \mu}(s'; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k) \\
&\geq \bar{r}_h^k(s, a, b) + \frac{1}{2}\beta_h^k(s, a, b) + \sum_{s'} \hat{P}_h^k(s'|s, a, b) V_{h+1}^{\pi, \mu}(s'; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k) \\
&= Q_h^{\pi, \mu}(s, a, b; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k),
\end{aligned}$$

Bounding Regret: With these, we have

$$\begin{aligned}
&\max_{\pi} \max_{\mu} V^{\pi, \mu}(s_0; P, r) \tag{18} \\
&\leq \max_{\pi} \max_{\mu} V^{\pi, \mu}(s_0; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k) \tag{Optimism property 1} \\
&\leq \max_{\mu} V^{\pi^k, \mu}(s_0; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k) \tag{Definition of \(\pi^k\)} \\
&\leq \max_{\mu} V^{\pi^k, \mu}(s_0; P, r + \beta^k) \tag{Optimism property 2} \\
&\leq V^{\pi^k, \mu^k}(s_0; P, r + \beta^k) \tag{Follower best responding in subsidized MDP}
\end{aligned}$$

where in the last line we use the shorthand that $\mu^k := \mu(\pi^k, r + \beta^k)$

This means that the instantaneous regret is bounded by:

$$\begin{aligned}
\text{reg}_k &= \max_{\pi} \max_{\mu} V^{\pi, \mu}(s_0; P, r) - V^{\pi^k, \mu^k}(s_0; P, r - \kappa\beta^k) \\
&\leq V^{\pi^k, \mu^k}(s_0; P, r + \beta^k) - V^{\pi^k, \mu^k}(s_0; P, r - \kappa\beta^k) \\
&= V^{\pi^k, \mu^k}(s_0; P, (1 + \kappa)\beta^k) \tag{Linearity of return}
\end{aligned}$$

In closing, we have the cumulative regret:

$$\begin{aligned}
\sum_{k=1}^T \text{reg}_k &\leq \sum_{k=1}^T V^{\pi^k, \mu^k}(s_0; P, (1 + \kappa)\beta^k) \\
&= \sum_{k=1}^T \sum_{i=1}^H (1 + \kappa) b(s_i^k, a_i^k, b_i^k) + O(H\sqrt{T}) \tag{Azuma's inequality} \\
&\leq O(\sqrt{H^3|S|^2|A||B|T})
\end{aligned}$$

where the last step holds as for any realization of $\{s_i^k, a_i^k, b_i^k\}_{k \in [T], i \in [H]}$, the sum is upper bounded by $O(\sqrt{H^2|S|} \cdot \sqrt{H|S||A||B|T})$.

□

Theorem C.4. *There exists an algorithm, leveraging UCB-VI-FP as subroutine, that incurs $O(T^{2/3})$ regret under upfront payment.*

Proof. By Lemma C.1, the total regret incurred by a learning algorithm under upfront payment is:

$$\mathcal{R}(T) = \sum_{i=1}^T V^{\pi^*, \mu^*} + 0 - \left(\sum_{i=1}^T \mathbb{E}[V^{\pi_i, \mu(\pi_i)}] - \kappa \cdot \sum_{s, a, b \in S \times A \times B} b^i(s, a, b) \right)$$

We can establish the $O(T^{2/3})$ with the following algorithm. First, run UCB-VI-FP for $m = T^{2/3}$ iterations. Then, using online to batch conversion, repeat this sequence of policies π_1, \dots, π_m for the remaining $T - m$ steps, with all payments set to zero.

As shown in Theorem 6.4, the policy regret from the explore phase is $\sum_{i=1}^m V^{\pi^*, \mu^*} - \mathbb{E}[V^{\pi_i, \mu(\pi_i)}] \leq O(T^{1/3})$. The payment from the explore phase is $\sum_{i=1}^m \kappa \sum_{s,a,b \in S \times A \times B} b^i(s, a, b) \leq m \cdot O(1) = O(T^{2/3})$. The policy regret from the exploit phase is $\sum_{i=m+1}^T V^{\pi^*, \mu^*} - \mathbb{E}[V^{\pi_i, \mu(\pi_i)}] \leq O((T - T^{2/3}) \frac{\sqrt{T^{2/3}}}{m}) = O(T^{2/3})$. And so, the total regret is $\mathcal{R}(T) = O(T^{2/3})$. \square

C.2 Contrasting Trajectory Payment with Upfront Payment

Proposition C.5. *UCB-VI-FP with indicator bonus incurs constant $O(|S||A||B|)$ regret under trajectory payment.*

Proof. Using Lemma C.2, we can reduce Stackelberg learning with payment to single-agent no-regret learning with joint policy class $\pi^{joint} : S \rightarrow A \times B$.

For the no-regret learning algorithm, we will again use the classical UCB-VI algorithm [Azar et al., 2017]. To obtain a tighter bound of the algorithm's expected regret, we construct an optimistic MDP M_k with reward $\hat{r}_h^k(s, (a, b)) := 1$ if (s, a, b) unvisited before step $k - 1$ at step h and $r(s, a, b)$ o.w. Since $r \in [-1, 1]$, $\hat{r}_h^k(s, a, b) \geq r(s, a, b)$ for all h, k .

Following the UCB-VI proof outline:

$$\begin{aligned} & \mathbb{E}[V_h^k(s_h^k) - V_h^{\pi_k}(s_h^k)] \\ &= \mathbb{E}[(Q_h^k - Q_h^{\pi_k})(s_h^k, (a_h^k, b_h^k))] \\ &= \mathbb{E}[P_h V_{h+1}^k(s_h^k, (a_h^k, b_h^k)) - P_h V_{h+1}^{\pi_k}(s_h^k, (a_h^k, b_h^k)) + (\hat{r}_h^k - r_h^k)(s_h^k, (a_h^k, b_h^k))] \\ &= \mathbb{E}[V_{h+1}^k(s_{h+1}^k) - V_{h+1}^{\pi_k}(s_{h+1}^k) + b_h^k] \quad (\text{let } b_h^k = \mathbb{E}[(\hat{r}_h^k - r_h^k)(s_h^k, a_h^k, b_h^k)]) \end{aligned}$$

Unrolling, we have that:

$$\sum_{k=1}^T \mathbb{E}[(V_1^* - V_1^{\pi_k})(s_1)] \leq \sum_{k=1}^T \mathbb{E}[(V_1^k - V_1^{\pi_k})(s_1)] \leq \mathbb{E}[\sum_{k=1}^T \sum_{h=1}^H b_h^k]$$

To finish, we observe that for every roll-out, we have that:

$$\sum_{k=1}^T \sum_{h=1}^H b_h^k = \sum_{s,a,b \in S \times A \times B} \mathbb{1} \left\{ N^T(s, a, b) \geq 1 \right\} (1 - r(s, a, b)) \geq 2|S||A||B|$$

\square

Proposition C.6. *There exists an algorithm, leveraging UCB-VI-FP with indicator bonus as subroutine, that incurs $O(T^{1/2})$ regret under upfront payment.*

Proof. We can establish the $O(T^{1/2})$ bound with the following algorithm. First, run UCB-VI-FP with indicator bonus for $m = T^{1/2}$ iterations. Then, using online to batch conversion, repeat this sequence of policies π_1, \dots, π_m for the remaining $T - m$ steps, with all payments set to zero.

As shown in Proposition 6.7, the policy regret from the explore phase is $\sum_{i=1}^m V^{\pi^*, \mu^*} - \mathbb{E}[V^{\pi_i, \mu(\pi_i)}] \leq O(1)$. The payment from the explore phase is $\sum_{i=1}^m \kappa \sum_{s,a,b \in S \times A \times B} b^i(s, a, b) \leq m \cdot O(1) = O(T^{1/2})$. The policy regret from the exploit phase is $\sum_{i=m+1}^T V^{\pi^*, \mu^*} - \mathbb{E}[V^{\pi_i, \mu(\pi_i)}] \leq O((T - T^{1/2}) \frac{1}{m}) = O(T^{1/2})$. And so, the total regret is $\mathcal{R}(T) = O(T^{1/2})$. \square

Proposition C.7. *There exists a subset of Markov Game instances such that any learning algorithm has to incur $\Omega(T^{1/2})$ regret under upfront payment.*

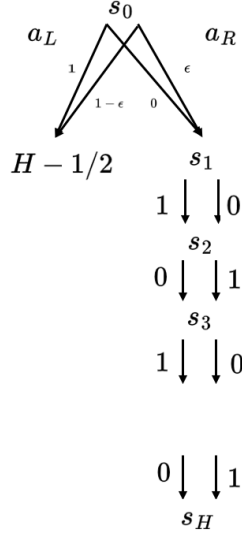


Figure 4: Right branch requires getting the right set of H binary actions at states s_1, \dots, s_H to exceed the return in the left branch.

Proof. Setup: We will use Yao's Lemma to show our result. Define cost of algorithm A at the i th episode in instance X as $cost(A, i, x) = \mathbb{E}[\mathbb{1}\{\pi_i \neq \pi^*\}(r(\pi^*) - r(\pi_i)) + S_i]$. Let \mathcal{D} be the uniform distribution over all MDP instances \mathcal{X} , where the distribution is uniform over which of the two actions at each time step achieves reward 1 (and the other 0). Our aim is to show $\min_A \mathbb{E}_{x \sim \mathcal{D}}[\sum_{i=1}^T cost(A, i, x)] \geq \Omega(\sqrt{T})$. This implies that any randomized algorithms R has $\min_R \max_{x \in \mathcal{X}} \mathbb{E}[\sum_{i=1}^T cost(R, i, x)] \geq \Omega(\sqrt{T})$.

Notation: Let the history up until time t be H_t :

$$H_t = (\pi_1, S_1, \tau_1 = (s_1^1, a_1^1, b_1^1, r(s_1^1, a_1^1, b_1^1)), \dots, s_H^1, a_H^1, b_H^1, r(s_H^1, a_H^1, b_H^1)), A_1, W'_1, \pi_2, S_2, \tau_2, \dots, \pi_t, S_t, \tau_t, A_t, W'_t)$$

- π_i is the right branch policy at time i
- τ_i is the trajectory in the Markov game at time i .
- S_i is the payment on the right branch at round i , and S'_i the payment on the left branch.
- Define right branch subsidy being at least $\frac{1}{2}$ as $X_i = \mathbb{1}(S_i \geq \frac{1}{2})$.
- $Y_i = \mathbb{1}\{\pi_i = \pi^*\}$.
- A_i denotes if the follower's BR is the right action.

$$A_i = 1$$

$$\begin{aligned} \Leftrightarrow r^F(s_0, a_2) &= (1 - \epsilon)(H - 1/2 + S'_i) + \epsilon(S_i + r(\pi_i)) \geq H - 1/2 + S'_i = r^F(s_0, a_1) \\ \Leftrightarrow \epsilon(S_i + r(\pi_i)) &\geq \epsilon(H - 1/2 + S'_i) \end{aligned}$$

And so,

$$A_i = \mathbb{1}(S_i + r(\pi_i) \geq S'_i + H - 1/2)$$

- Let Z_i denote Bernoulli(ϵ) random variable corresponding to the stochastic transition, when the follower chooses action a_2 .

$Z_i = 1$ if the follower transitions to the right branch provided $A_i = 1$.

- Let W'_i denotes if the follower goes into the right branch. We have that:

$$W'_i = A_i \wedge Z_i$$

We are interested in the event $W_t = Y_t \vee X_t Z_t$. It has the useful property that $W_t = 0 \Rightarrow Y_t = 0, X_t Z_t = 0$. Moreover, we claim that $W_t = 0 \Rightarrow W'_t = 0$.

$$\begin{aligned}
& Y_t = 0, X_t Z_t = 0 \\
& \Rightarrow r(\pi_t) \leq H - 1, X_t Z_t = 0 \\
& \Leftrightarrow H - 1/2 - r(\pi_t) \geq 1/2, X_t Z_t = 0 \\
& \Rightarrow X_t \geq A_t, X_t Z_t = 0 \quad (A_t = 1 \Rightarrow S_t \geq S'_t + H - 1/2 - r(\pi_t) \geq S'_t + 1/2 \Rightarrow X_t = 1) \\
& \Rightarrow A_t Z_t = 0 \quad (0 = X_t Z_t \geq A_t Z_t) \\
& \Rightarrow W'_t = 0
\end{aligned}$$

We will use this property in the lemma that follows.

Lemma C.8.

$$P(Y_t = 0 | W_{:t-1} = 0) \geq 1 - \frac{t-1}{2^H}$$

Proof.

$$\begin{aligned}
& \sum_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1}) P(H_{t-1} | W_{:t-1} = 0) \\
& = \sum_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1}, \pi_t = \pi') P(H_{t-1} | W_{:t-1} = 0) \\
& \quad (\pi_t \text{ is a deterministic function of } H_{t-1}) \\
& = \sum_{H_{t-1}: W_{:t-1}=0} P(\pi^* \neq \pi' | H_{t-1}, \pi_t = \pi') P(H_{t-1} | W_{:t-1} = 0) \\
& = \sum_{H_{t-1}: W_{:t-1}=0} P(\pi^* \neq \pi' | Y_{:t-1} = 0, \pi_{:t-1}) P(H_{t-1} | W_{:t-1} = 0) \quad (\dagger) \\
& \geq 1 - \frac{t-1}{2^H} \quad (\text{posterior of } \pi^* \text{ is uniform over } \Pi \setminus \pi_{:t-1})
\end{aligned}$$

(\dagger) : First, we have that the observed trajectory has the specific functional form: $\tau_i = \tau^{\pi_i} \mathbb{1}\{W'_i = 1\} + \tau_{\text{left}} \mathbb{1}\{W'_i = 0\}$. Thus, conditioned on $W_{:t-1} = 0 \Rightarrow W'_{:t-1} = 0$, we have that $\tau_i = \tau_{\text{left}}$ for all $i \in [t-1]$. That is, $\pi^* \perp\!\!\!\perp \tau_i$ in the *conditional* joint distribution.

From this, we have that $\pi^* \perp\!\!\!\perp H_{t-1} \setminus \{Y_{:t-1} = 0, \pi_{:t-1}\} \mid \{Y_{:t-1} = 0, \pi_{:t-1}\}$ by checking D-separation of $\{S_i, A_i, \tau_i, W'_i\}$ with π^* . π^* 's only children are Y_i 's, thus conditioning on Y_i 's other parents (π_i 's) and Y_i 's themselves blocks every path to the rest of the random variables.

□

Lemma C.9. For every $t \leq T$:

$$P(W_{:t} = 0) \geq \frac{1}{2} \left(1 - \sum_{j=1}^t \epsilon P(X_j = 1 | W_{:j-1} = 0)\right)$$

Proof. $W_{:t} = 0 \Leftrightarrow Y_{:t} = 0, X_{:t} Z_{:t} = 0$. Towards bounding the product, we have:

$$\begin{aligned}
& P(Y_t = 0, X_t Z_t = 0 | W_{:t-1} = 0) \\
&= \sum_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0, X_t Z_t = 0 | H_{t-1}) P(H_{t-1} | W_{:t-1} = 0) \\
&\quad \text{(where } H_{t-1} \text{ is the history up until } t-1, W_{:t-1} \in H_{t-1}) \\
&= \sum_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1}) P(X_t Z_t = 0 | H_{t-1}) P(H_{t-1} | W_{:t-1} = 0) \\
&\quad (X_t \text{ deterministic function of } H_{t-1}, Z_t \text{ is independent of } Y_t \Rightarrow Y_t \perp\!\!\!\perp X_t Z_t | H_{t-1}) \\
&\geq \min_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1}) \\
&\quad \sum_{H_{t-1}: W_{:t-1}=0} (P(X_t Z_t = 0 | H_{t-1}, Z_t = 0) P(Z_t = 0 | H_{t-1}) + P(X_t = 0 | H_{t-1}, Z_t = 1) P(Z_t = 1 | H_{t-1})) P(H_{t-1} | W_{:t-1} = 0) \\
&\quad \text{(condition on } Z_t) \\
&= \min_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1}) \sum_{H_{t-1}: W_{:t-1}=0} (1 - \epsilon + P(X_t = 0 | H_{t-1}, Z_t = 1) \epsilon) P(H_{t-1} | W_{:t-1} = 0) \\
&= \min_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1}) (1 - \epsilon + \epsilon \sum_{H_{t-1}: W_{:t-1}=0} P(X_t = 0 | H_{t-1}) P(H_{t-1} | W_{:t-1} = 0)) \\
&\quad (X_t \text{ is deterministic function of } H_{t-1}, \text{ so independent of } Z_t) \\
&= \min_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1}) (1 - \epsilon + \epsilon \sum_{H_{t-1}: W_{:t-1}=0} P(X_t = 0 | H_{t-1}, W_{:t-1} = 0) P(H_{t-1} | W_{:t-1} = 0)) \\
&= \min_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1}) (1 - \epsilon + \epsilon P(X_t = 0 | W_{:t-1} = 0)) \\
&= \min_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1}) (1 - \epsilon P(X_t = 1 | W_{:t-1} = 0))
\end{aligned}$$

From before, for any history H_{t-1} where $W_{:t-1} = 0$:

$$\begin{aligned}
& P(Y_t = 0 | H_{t-1}, W_{:t-1} = 0) \\
&= P(\pi^* \neq \pi' | Y_{:t-1} = 0, \pi_{:t-1}) \quad \text{(from prior lemma)} \\
&\geq 1 - \frac{t-1}{2^H}
\end{aligned}$$

Putting it together,

$$P(Y_t = 0, X_t Z_t = 0 | W_{:t-1} = 0) \geq (1 - \frac{t-1}{2^H}) (1 - \epsilon P(X_t = 1 | W_{:t-1} = 0))$$

and so,

$$\begin{aligned}
& P(Y_{:t} = 0, X_{:t} Z_{:t} = 0) \\
&\geq \frac{1}{2} \prod_{i=1}^t (1 - \epsilon P(X_i = 1 | W_{:i-1} = 0)) \quad \text{(using that } 2^H \gg T) \\
&\geq \frac{1}{2} (1 - \sum_{j=1}^t \epsilon P(X_j = 1 | W_{:j-1} = 0))
\end{aligned}$$

□

Note that the optimal expected return is: $(1 - \epsilon)(H - 1/2) + \epsilon H$. When the follower chooses a_L , the instantaneous regret is: $(1 - \epsilon)(H - 1/2) + \epsilon H - (H - 1/2) = \epsilon/2$. If the follower chooses a_R and $\pi_t \neq \pi^*$, then the instantaneous regret is at least ϵ . Overall, the instantaneous regret is at least $\epsilon/2$ when $\pi_t \neq \pi^*$. And so, the cumulative regret bound is lower bounded by:

$$\begin{aligned}
& \sum_{t=1}^T P(\pi_t \neq \pi^*)\epsilon/2 + \mathbb{E}[\sum_{t=1}^T S_t + S'_t] \\
& \geq \sum_{t=1}^T P(\pi_t \neq \pi^*)\epsilon/2 + 1/2\mathbb{E}[\sum_{t=1}^T X_t] \\
& \geq \epsilon/2 \sum_{t=1}^T P(\pi_t \neq \pi^* | W_{:t-1} = 0) P(W_{:t-1} = 0) + 1/2\mathbb{E}[\sum_{t=1}^T X_t] \\
& \geq \epsilon/2 \sum_{t=1}^T (1 - \frac{t-1}{2^H}) P(W_{:t-1} = 0) + 1/2\mathbb{E}[\sum_{t=1}^T X_t] \\
& \geq \frac{\epsilon}{4} \sum_{t=1}^T P(W_{:t-1} = 0) + 1/2\mathbb{E}[\sum_{i=1}^T X_i]
\end{aligned}$$

We consider two cases:

1. Case 1: $\mathbb{E}[\sum_{i=1}^T X_i] = \sum_{i=1}^T P(X_i = 1) \geq 1/16\epsilon$

In this case, the regret is at least $1/32\epsilon$.

2. Case 2: $\sum_{i=1}^T P(X_i = 1) < 1/16\epsilon$

Then, we claim that $\sum_{j=1}^T P(X_j = 1 | W_{:j-1} = 0) \leq 1/4\epsilon$.

Suppose not and $\sum_{j=1}^T P(X_j = 1 | W_{:j-1} = 0) > 1/4\epsilon$.

Then, there must exist $t < T$ such that $\sum_{j=1}^t P(X_j = 1 | W_{:j-1} = 0) \in [1/4\epsilon - 1, 1/4\epsilon]$.

This implies:

$$\begin{aligned}
& \sum_{i=1}^T P(X_i = 1) \\
& \geq \sum_{i=1}^t P(X_i = 1 | W_{:i-1} = 0) P(W_{:i-1} = 0) \\
& \geq \sum_{i=1}^t P(X_i = 1 | W_{:i-1} = 0) \frac{1}{2} (1 - \epsilon \sum_{j=1}^{i-1} P(X_j = 1 | W_{:j-1} = 0)) \\
& \geq \sum_{i=1}^t P(X_i = 1 | W_{:i-1} = 0) \frac{1}{2} (1 - \epsilon \cdot 1/4\epsilon) \\
& = \frac{3}{8} \sum_{i=1}^t P(X_i = 1 | W_{:i-1} = 0) \\
& \geq \frac{3}{8} (1/4\epsilon - 1) \\
& > 1/16\epsilon
\end{aligned}$$

which is a contradiction.

Now, because $\sum_{j=1}^T P(X_j = 1|W_{:j-1} = 0) \leq 1/4\epsilon$, the cumulative regret from before is:

$$\begin{aligned}
&\geq \frac{\epsilon}{4} \sum_{t=1}^T P(W_{:t-1} = 0) + \mathbb{E}[\sum_{i=1}^T X_i] \\
&\geq \frac{\epsilon}{4} TP(W_{:T-1} = 0) \\
&\geq \frac{\epsilon}{4} T \frac{1}{2} (1 - \epsilon \sum_{j=1}^{T-1} P(X_j = 1|W_{:j-1} = 0)) \\
&\geq \frac{\epsilon}{4} T \frac{1}{2} (1 - \epsilon \frac{1}{4\epsilon}) \\
&\geq \frac{3T\epsilon}{32}
\end{aligned}$$

In conclusion, the cumulative regret is at least $\min(\frac{1}{32\epsilon}, \frac{3T\epsilon}{32}) = \Omega(T^{1/2})$ when we let $\epsilon = T^{-1/2}$.

□

D Experiments

Setup: We consider a turn-based Markov Game, where the leader is solving a RL problem and the follower solves a bandit problem in its BR [Bai et al., 2021]. This class of Markov Games includes the hard instance construction in Section 5. Our goal is to examine whether learning without payment can get stuck even in more “average” (and not worst) case Markov Games.

For experimentation, the leader is learning in a toy MDP with $H = 5$. For the follower, arm a_1 leads to a MDP, whose optimal return is the optimal return in the (turn-based) Markov game. On the other hand, arm a_2 has a deterministic high reward that is α that of the optimal return.

By varying α , we can make the follower get “stuck” in myopically choosing a_2 , thus preventing the leader from exploring and learning the actual optimal policy. This is the intuition behind the negative results, Theorem 5.1 and Theorem 5.2, where we set α very high.

For the baseline, we use the single-agent learning algorithm UCB-VI in the without payment case, and compare it against UCB-VI-FP in the with payment case. We track the cumulative regret of the two learning algorithms over 40000 episodes and across 20 runs.

Finding: We experiment with different α ’s, finding that learning without payment can get stuck in the myopic optimum even when α is as low as 0.5. Interestingly, this suggests that there are “non-worst-case” Markov games, where exploration can be difficult without payment. Under $\alpha = 0.5$, Figure 5 shows that:

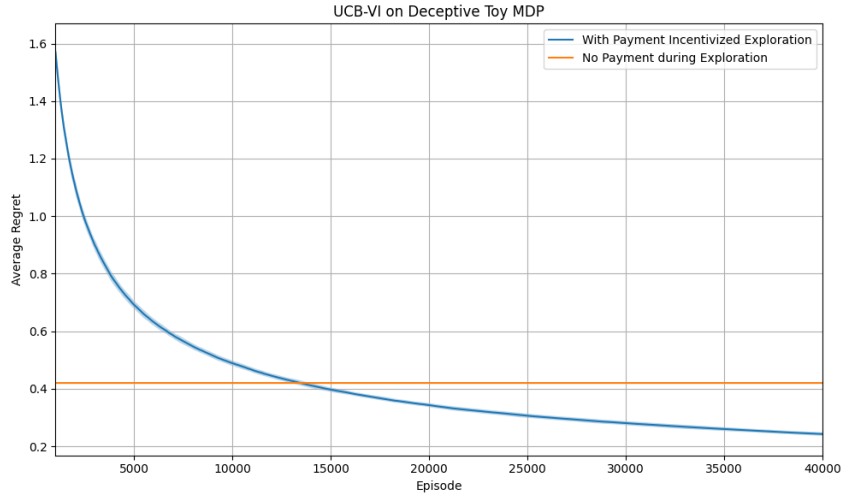


Figure 5: Regret plot between episodes 1000 to 40000

1. In absence of payment, even if the leader is using a (one-sided) no-regret algorithm (UCB-VI), the leader may not be able to explore adequately and incur linear regret.
2. UCB-VI-FP attains sublinear regret, showing the importance of payment in incentivizing exploration needed to learn the optimal policy.
3. UCB-VI-FP initially incurs a higher regret, which we expect due to the additional payment used by the algorithm to incentivize exploration. Over time, its regret improves due to exploration shrinking the policy regret, and reduced incentivization (and thus payment regret). Eventually, UCB-VI-FP’s regret dips below that of UCB-VI, with a crossover point at around episode 13500.

E Additional Related Works

Thematically, our paper belongs to the intersection of literature on Stackelberg policy computation in Markov games and literature on contracting through reward shaping in MDPs.

Other Variants of Follower best response in Stackelberg Games: In our paper, we consider the standard assumption of the follower best responding as well as its generalization, the λ -entropy regularized best response model. Moving further away from this canonical formulation in Stackelberg games, there have also been other formulations of follower behavior. Zhong et al. [2023], Chen et al. [2023] study learning the optimal policy in face of a myopic follower that greedily best responds. Furthermore, there is also a growing line of work on learning in face of an agent (follower) who is also learning. For example, Guruganesh et al. [2024] studies how to contract an agent that is learning. Kao et al. [2022] studies learning in face of a follower that is also learning in cooperative games.

Broader Stackelberg games literature: Our paper focuses specifically on Stackelberg Markov games, and adds to the body of work that builds a closer connection with RL theory. Zooming back out, the broader Stackelberg game literature is vast and varied. For example, there is an extensive body of work studying Stackelberg games in normal-form games (horizon-one games), often inspired by security games, as well as empirical methods for Stackelberg games. We mention [Balcan et al., 2015] and [Gerstgrasser and Parkes, 2023] as an example of each line of work, in which one may find further relevant references.

Learning the Optimal Payment Scheme in MDPs: It is natural to ask if there are any implications from prior papers on contracting in MDPs [Ben-Porat et al., 2024, Bollini et al., 2024, Ivanov et al., 2024, Wu et al., 2024]. Do papers in the single-agent setting have any implications for the more general two-player Stackelberg Markov game setup we consider?

In the full information setting that we consider, we have already covered the difference between the Markov game setting we consider and the bandit setting studied in [Scheid et al., 2024]. Another paper that studies the full information setting is that of [Ben-Porat et al., 2024], which proves that planning is NP-hard, albeit under a different formulation where the leader aims to maximize the return subject to the payment being capped by some budget. By contrast, similar to previous works by Scheid et al. [2024], Wu et al. [2024], our paper studies maximizing the return minus the total payment. Hence, it is not immediately clear how the results carry over to our setting.

Besides this, other papers [Bollini et al., 2024, Ivanov et al., 2024, Wu et al., 2024] focus on the imperfect information settings, where the payment cannot be a function of the follower’s action (hidden at the time of payment). And so, in this case, it is also not clear how results transfer due to differing setups.

As our paper is the first to study Stackelberg Markov games with payment where the leader can set both the policy and the payment, the generality of our setup means that new results arise. For instance, we show hardness results in Cooperative Markov games that do not exist under existing contracting in MDP settings [Ben-Porat et al., 2024, Bollini et al., 2024, Ivanov et al., 2024, Wu et al., 2024]. This finding motivates us to study how to learn efficiently with payment in cooperative games, and we develop no-regret algorithms to this end.

UCB-VI-FP is a notable multi-agent algorithm as the algorithm can only control one player when exploring. Indeed, our results show that even if the leader is using a no-regret learning algorithm, then learning can still be inefficient. And so, a new learning algorithm needs to be developed here, using payment to incentivize *collective* exploration. Finally, we add that we also obtain results under upfront payment, which is a new form of payment that has not been considered in previous contracting in MDP literature.

Bi-level Optimization: While the primary goal of our paper is to study global optimum, we note that bi-level optimizers can tractably compute *local optimum* in planning under known rewards and dynamics Dempe and Zemkoho [2012], Naveiro and Insua [2019], Thoma et al. [2024], Shen et al. [2024]. To handle the learning setting with unknown rewards and dynamics, we develop a new algorithm (UCB-VI-FP) for adaptive exploration while minimizing regret.

F Incentive Effects when Follower Reward is Unobservable

Incentive Effects: A key underlying assumption in our setup is that the leader can readily observe the follower's reward and/or trust that the follower has reported their true reward. Truthfulness is important in the partnership, but suppose we allow the follower to misreport all rewards up to Δ , what may happen then? We have the following result in the direct-payment case studied by Scheid et al. [2024].

Proposition F.1. *Suppose the follower can misreport r^F up to Δ , $\|r'^F - r^F\|_1 \leq \Delta$. In the bandit setting, the follower's return can change by at most:*

$$|V^{\pi^*(r^F), \mu(\pi^*(r^F))}(s_0; r^F + b^*(r^F)) - V^{\pi^*(r'^F), \mu(\pi^*(r'^F))}(s_0; r^F + b^*(r'^F))| \leq 2\Delta$$

and the leader's return can change by at most:

$$|V^{\pi^*(r^F), \mu(\pi^*(r^F))}(s_0; r^L - b^*(r^F)) - V^{\pi^*(r'^F), \mu(\pi^*(r'^F))}(s_0; r^L - b^*(r'^F))| \leq 2\Delta$$

Proof. We show that in the bandit setting, the follower's return differs by at most 2Δ , as does the leader's return.

The bandit Stackelberg setting is such that the leader optimizes:

$$\begin{aligned} \max_i \quad & r_i^L - b_i \\ \text{s.t.} \quad & r_i^F + b_i \geq \max_{j \neq i} r_j^F \end{aligned} \tag{19}$$

The follower may instead report r'^F s.t. $\|r'^F - r^F\|_1 \leq \Delta$. Let i^* be the optimal arm under r^F and arm i' under r'^F .

We observe that in all bandit games with reward r^F , the follower's return is $\max_j r_j^F$. If $i^* = \arg\max_j r_j^F$, then it's clear that $b_{i^*} = 0$ as lowering it to zero preserves the follower choosing arm i^* , while increasing the leader's return. In the other case, $r_{i^*}^F + b_{i^*} \geq \max_j r_j^F$. If this is not tight, then we can lower b_{i^*} s.t. it is tight and preserve the follower choosing arm i^* , while increasing the leader's return.

With this, the return of the follower under truthful reporting is $\max_j r_j^F$. Under r'^F reporting, it's $r_{i'}^F + \max_j r_j'^F - r_{i'}^F$ (note that it gets the true reward $r_{i'}^F$). The difference is thus:

$$|r_{i'}^F + \max_j r_j'^F - r_{i'}^F - \max_j r_j^F| \leq |r_{i'}^F - r_{i'}^F| + |\max_j r_j'^F - \max_j r_j^F| \leq 2\Delta$$

since

$$\max_j r_j'^F \geq r_k' \geq r_k - \Delta = \max_j r_j^F - \Delta$$

and

$$\max_j r_j^F = r_k \geq r_l \geq r_l' - \Delta = \max_j r_j'^F - \Delta$$

Moreover, the leader's return also differs by at most 2Δ :

$$r_{i^*}^L - (\max_j r_j^F - r_{i^*}^F) \geq r_{i'}^L - (\max_j r_j^F - r_{i'}^F) \geq r_{i'}^L - (\max_j r_j'^F - r_{i'}^F) - 2\Delta$$

and

$$r_{i'}^L - (\max_j r_j'^F - r_{i'}^F) \geq r_{i^*}^L - (\max_j r_j'^F - r_{i^*}^F) \geq r_{i^*}^L - (\max_j r_j^F - r_{i^*}^F) - 2\Delta$$

□