
Stochastic Principal-Agent Problems: Computing and Learning Optimal History-Dependent Policies

Jiarui Gan

University of Oxford
jiarui.gan@cs.ox.ac.uk

Rupak Majumdar

MPI-SWS
rupak@mpi-sws.org

Debmalya Mandal

University of Warwick
debmalya.mandal@warwick.ac.uk

Goran Radanovic

MPI-SWS
gradanovic@mpi-sws.org

Abstract

We study a stochastic principal-agent model. A principal and an agent interact in a stochastic environment, each privy to observations about the state not available to the other. The principal has the power of commitment, both to elicit information from the agent and to signal her own information. The players communicate with each other and then select actions independently. Both players are *far-sighted*, aiming to maximize their total payoffs over the entire time horizon. We consider both the computation and learning of the principal’s optimal policy. The key challenge lies in enabling *history-dependent* policies, which are essential for achieving optimality in this model but difficult to cope with because of the exponential growth of possible histories as the size of the model increases; explicit representation of history-dependent policies is infeasible as a result. To address this challenge, we develop algorithmic techniques based on the concept of *inducible value set*. The techniques yield an efficient algorithm that computes an ϵ -approximate optimal policy in time polynomial in $1/\epsilon$. We also present an efficient learning algorithm for an episodic reinforcement learning setting with unknown transition probabilities. The algorithm achieves sublinear regret $\tilde{O}(T^{2/3})$ for both players over T episodes.

1 Introduction

Many problems in economic theory involve sequential reasoning between multiple parties with asymmetric access to information [37, 20, 4, 27]. For example, in contract theory, one party (the principal) delegates authority and decision-making power to another (the agent), and the goal is to design mechanisms to ensure that the agent’s actions align with the principal’s utilities. This broad class of *principal-agent problems* lead to many research questions about information design and optimal strategic behaviors, with broad-ranging applications from governance and public administration to e-commerce and financial services. In particular, *algorithmic* techniques for optimal decision-making and learning are crucial for obtaining effective solutions to real-world problems in this domain.

In this paper, we consider a general framework for stochastic principal-agent problems. We study the algorithmic problems related to the computation and learning of optimal solutions under this framework. In this framework, the interaction between the principal and the agent takes place in a stochastic environment over multiple time steps. In each step, both players are privy to information not available to the other and make partial observations about the environment. The players can communicate their private information to influence each other and, based on this communication, play actions that jointly influence the state of the environment. Each player has their own payoff, and we study the *general-sum case* with *far-sighted* players: each player aims to maximize the (expected)

sum of their rewards over the entire horizon of the game. Technically, the setting is a stochastic game with partial information on both sides [1, 31].

In line with the principal-agent framework, we assume that the principal has the power of *commitment*, both to elicit information from the agent and to signal the agent about her own private information. A commitment is a binding agreement for the principal to act according to the committed policy. The agent acts optimally in response to the commitment, deciding what information to share and what actions to perform at their own discretion. At a high level, we have a Stackelberg game and we aim to find a Stackelberg equilibrium [40]. The possibility of information exchange further results in a model that incorporates both sequential Bayesian persuasion (or information design [22]) [16, 41] and sequential mechanism design [45]. We adopt *stochastic games* (or Markov games) [39] as our modeling basis, whose transition models are graph-structured. Hence, our model is more expressive than tree-structured principal-agent problems such as those based on extensive-form games (EFGs), in the sense that exponentially many possible history trajectories can be unrolled from our graph-based model as the length of the time horizon increases, whereas in tree-structured ones, the number is bounded by the size of the tree.

Indeed, *history-dependence* is a key consideration in our policy design. Unlike in single-player Markov decision processes (MDPs) or zero-sum stochastic games, stationary (or history-independent) policies are no longer without loss of generality in our setting. For example, strategies like tit-for-tat—which are essential for achieving optimal outcomes in certain scenarios (such as the repeated prisoner’s dilemma)—depend on remembering past actions and thus cannot be implemented by stationary policies. Enabling history-dependent policies in our graph-based model, however, begets a major challenge caused by the exponential growth of possible histories. Explicitly representing a history-dependent policy requires specifying a strategy for every possible history, which quickly becomes infeasible as the number of histories increases exponentially with the length of the time horizon. To further optimize such a policy is even more challenging as the output of this computational task would be a function over an exponentially large domain. In contrast, the size of a stationary policy is linear in the number of states.

1.1 Our Contributions

Surprisingly though, as we discover in this work, history-dependent policies can be represented as value vectors and can be efficiently unrolled provided polytopes consisting of all inducible value vectors. Building on this insight, we develop algorithmic techniques that work by constructing the *inducible value sets*. These techniques make it possible to implement and optimize history-dependent strategies in dynamic principal-agent settings. More surprisingly, they are computationally efficient, so not only do they significantly expand the policy space—from stationary policies to the far more expressive class of history-dependent ones—but they also address a key computational barrier posed by stationary policies, to optimize which, even approximately, is known to be NP-hard [15].

In summary, we make the following contributions in this paper.

- We introduce a generalized framework for principal-agent problems in stochastic environments, which incorporates the stochastic extensions of various subcases of principal-agent problems such as information design [22] and automated mechanism design [38].
- We develop algorithmic techniques that enable the implementation and optimization of history-dependent strategies in the stochastic principal-agency framework. The techniques are based on a compact representation of fully history-dependent policies and a novel approach that works by constructing the inducible value sets. Together, they yield an efficient near-optimal algorithm that computes an ϵ -approximate optimal policy in time polynomial in $1/\epsilon$, while the policy it produces ensures *exact* incentive compatibility (IC).
- We also consider an episodic RL setting where the transition dynamics are unknown and must be learned through interaction. We present a learning algorithm that guarantees sublinear regret $\tilde{O}(\text{poly}(M, H) \cdot T^{2/3})$ for both players, where M is the model size, H is the episode length, and T is the number of episodes. The algorithm builds on recent advances in *reward-free exploration* from the RL literature and it leverages efficient computation of *approximately* IC policies, using a variant of our algorithm for the full-information setting.

1.2 Related Work

The principal-agent problem is a well-known concept in economics studies [see, e.g., 37, 33, 32, 29]. Models featuring sequential interactions have also been proposed and studied [34, 12]. Our work follows the same modeling approach as these early works and extends the one-shot versions of the respective models—such as information design [22], automated mechanism design [38], as well as mixtures of the two [33, 7, 14]—into the sequential setting. There has been a growing interest in the algorithmic aspects of these sequential models, focusing on their computation and learning (e.g., information design [8, 15, 16, 41, 2], automated mechanism design [45, 6], other types of sequential Stackelberg games [25, 26, 5, 18, 10], and even more recently, contract design [42, 19]).

More specifically, Gan et al. [15] first introduced an infinite-horizon information design model based on an MDP. They showed that optimal *stationary* strategies are inapproximable, unless the receiver is myopic. This work left open the tractability of optimal *history-dependent* strategies, especially in finite-horizon models, which we consider in this paper. Wu et al. [41] later studied the reinforcement learning problem against a myopic agent in the same sequential information design model. Bernasconi et al. [2] also studied the same problem in a model based on an EFG; efficient computation and learning algorithms were presented. Similar EFG-based models have also been explored in the recent literature [43, 44]. EFGs are tree-structured and hence easier than MDP-based models in the sense that all history sequences are modeled explicitly in the model. The number of possible histories is bounded by the size of the problem as a result, where as this can be exponential in an MDP. Hence, efficient algorithms for EFG-based models do not directly translate to efficient algorithms for our MDP-based model. In the domain of automated mechanism design, Zhang and Conitzer [45] studied a finite-horizon model that is a POMDP for the principal and MDP for the agent. They presented a linear program for computing optimal mechanisms, whose size is exponential in the problem size.

As mentioned earlier, our algorithm leverages compact representation of history-dependent policies and construction of inducible value sets. Similar techniques have been proposed in earlier works by Dermed and Isbell [11] and MacDermed et al. [28] to compute optimal correlated equilibria of stochastic games. Our algorithm extends these techniques into the principal-agent setting, with adaptations that ensure exact incentive compatibility (IC) in the full-information setting. Concurrently with our work [17], Bernasconi et al. [3] also used a similar approximation approach to solve an information design problem (as a special case of our model). Compared to their results, our algorithm guarantees *exact* IC, with a more straightforward approach. Moreover, we also study the learning problem where the transition model needs to be learned, in addition to the computation problem where this is given. While all the above works (including ours) only guarantee near-optimality, exact solutions are possible in some special cases. In a recent work of Zhang et al. [46], they presented a sophisticated exact algorithm for computing optimal correlated equilibria in two-player turn-based stochastic games. In infinite-horizon games, however, such exact solutions appear computationally infeasible because of irrational numbers involved in optimal policies; we refer the reader to a more recent work by Gan and Majumdar [13].

2 Preliminaries

A principal (P) and an agent (A) interact in a finite-horizon POMDP $\mathcal{M} = \langle S, A, \Omega, p, \mathbf{r} \rangle$, where: S is a finite state space; $A = A^P \times A^A$ is a finite joint action space; $\Omega = \Omega^P \times \Omega^A$ is a finite joint observation space; $p = (p_h)_{h=0}^{H-1}$ and $\mathbf{r} = (\mathbf{r}_h)_{h=1}^H$ are two tuples, each consisting of an element for every time step h . Specifically, $p_0 \in \Delta(S \times \Omega)$ is a distribution of the initial state-observation pairs, and each $p_h, h \geq 1$, is a transition function $p_h : S \times A \rightarrow \Delta(S \times \Omega)$. Each $\mathbf{r}_h = (r_h^P, r_h^A)$ is a pair of reward functions $r_h^P : S \times A \rightarrow \mathbb{R}$ and $r_h^A : S \times A \rightarrow \mathbb{R}$, for the principal and the agent, respectively. W.l.o.g., we assume that all rewards are in $[0, 1]$, and all rewards in the last time step H are 0.

The interaction proceeds as follows. At the beginning, an initial state-observation pair $(s_1, \omega_1) \sim p_0$ is drawn. Then, each time step $h = 1, \dots, H$ involves the following stages.

1. **Observation:** The principal and the agent observe, privately, ω_h^P and ω_h^A , respectively (but not s_h).
2. **Communication:** The principal elicits the agent's observation. The agent reports $\tilde{\omega}_h^A \in \Omega^A$ (possibly different from ω_h^A). Then, based on ω_h^P and $\tilde{\omega}_h^A$ the principal recommends an action a_h^A for the agent to play. The recommendation is sent to the agent as a coordination signal.

3. **Action:** Based on the information exchange above, the principal and the agent, simultaneously, each perform an action, say a_h^P and $\tilde{a}_h^A$, respectively. (The action $\tilde{a}_h^A$ the agent actually performs may be different from the recommended one a_h^A .)
4. **Rewards and next state:** Rewards $r_h^P(s_h, a_h^P, \tilde{a}_h^A)$ and $r_h^A(s_h, a_h^P, \tilde{a}_h^A)$ are generated for the principal and agent, respectively. The next state is drawn: $s_{h+1} \sim p_h(\cdot | s_h, a_h^P, \tilde{a}_h^A)$.

Following the general paradigm of principal-agent problems, we consider the principal’s *commitment* to a coordination policy. The agent best-responds to the principal’s commitment. Both players are *far-sighted* and aim to maximize the sum of their rewards over the H time steps.¹ We take the principal’s perspective and the goal, as we will shortly formalize, is to compute the principal’s optimal commitment. At a high level, this is a Stackelberg game between the principal and the agent and we aim to compute a Stackelberg equilibrium.

2.1 Hindsight Observability

We assume *hindsight observability*, whereby both players observe the full interaction history (including s_h , ω , and $\mathbf{a}$) at the *end* of each time step. This condition is essential for circumventing intrinsic computational complexity barriers: without it, our model would directly subsume partially observable MDPs (POMDPs), which is famously known to be PSPACE-hard [36], in which case any efficient algorithms would be hopeless unless $P = PSAPCE$. (See a further discussion in Appendix C.) Similar assumptions have been adopted in the POMDP literature [24]. Importantly, our model remains highly expressive under hindsight observability, capturing a broad range of relevant subcases, including: scenarios where the state is immediately observable, e.g., repeated games, stochastic games with full state observability [10], as well as scenarios where observations can be interpreted as external parameters generated based on an internal Markovian state observable to both players (e.g., [15, 41]).

Indeed, in many real-world scenarios, it is natural for players to observe each other’s actions after they are taken (consider, e.g., repeated rock–paper–scissors). Beyond actions, players may also receive regular updates about each other’s private observations. This occurs in particular when each player observes a different part of the state and the full state is revealed at the end of the time step—at which point all private observations effectively become public. For example, traders may each observe local or regional market information that is later aggregated and released publicly; in energy markets, regional grid operators obtain first-hand information about supply and demand within their own regions and later receive system-wide reports covering the entire market; and in an R&D consortium, each firm privately evaluates prototypes or test results and acts on that information during the reporting cycle, after which all results are disclosed to members under agreed rules.

2.2 History-dependent Policy

We consider history-dependent policies, which are more general than stationary policies and hence typically yield higher payoffs. For example, to punish the agent for performing a certain action requires a history-dependent policy that remember the agent’s action in the previous time step. History-dependent policies are also a natural choice for finite-horizon models, where the memory required to track the history is bounded by the horizon length.

A *history* up to time step h is a sequence $\sigma = (s_\ell, \omega_\ell, \tilde{\omega}_\ell^A, \mathbf{a}_\ell, \tilde{a}_\ell^A)_{\ell=1}^h$, containing elements in the four stages of each step described above (and we write $\omega_\ell = (\omega_\ell^P, \omega_\ell^A)$ and $\mathbf{a}_\ell = (a_\ell^P, a_\ell^A)$). We let Σ_h denote the set of all sequences till time step h , and let $\Sigma = \bigcup_{h=0}^H \Sigma_h$, where $\Sigma_0 = \{\emptyset\}$ contain only the empty sequence. Moreover, we denote by $\bar{\Sigma} := S \times \Omega \times \Omega^A \times A \times A^A$ the set of all possible interactions within one time step. We can now write the transition function as $p_h(\cdot | \sigma) = p_h(\cdot | s_h, \mathbf{a}_h)$ for any given sequence $\sigma \in \Sigma_h$ (specially, $p_0(\cdot | \emptyset) = p_0(\cdot)$).

Principal’s Policy A history-dependent policy takes the form $\pi : \Sigma \times \Omega \rightarrow \Delta(A)$, whereby upon seeing σ in the previous steps, observing ω^P , and receiving the agent’s report $\tilde{\omega}^A$ in the current step, the principal draws a joint action $\mathbf{a} = (a^P, a^A) \sim \pi(\sigma; \omega^P, \tilde{\omega}^A)$, sends a^A to the agent as an action recommendation, and performs a^P herself. We denote by $\pi(\mathbf{a} | \sigma; \omega^P, \tilde{\omega}^A)$ the probability of each $\mathbf{a}$ in $\pi(\sigma; \omega^P, \tilde{\omega}^A)$.

¹While we assume no discounting, all our results easily extend to discounted rewards.

Agent's Response The principal's commitment results in a meta-POMDP for the agent. The agent reacts by playing optimally in this meta-POMDP. When the principal's policy is IC, this simply means responding truthfully. Formally, the agent's strategy can be described by a *deviation plan* $\rho : (\sigma, \omega^A) \mapsto (\tilde{\omega}^A, f : A^A \rightarrow A^A)$, such that given any history σ and observation ω^A in the current step, the agent reports $\tilde{\omega}^A$ and then plays $\tilde{a}^A = f(a^A)$ if subsequently the principal recommends playing a^A . For simplicity, we write $\tilde{\omega}^A = \rho(\sigma; \omega^A)$ and $\tilde{a}^A = \rho(\sigma; \omega^A, a^A)$. We denote by $\perp$ the special deviation plan where no deviation is made, i.e., $\perp(\sigma; \omega^A) \equiv \omega^A$ and $\perp(\sigma; \omega^A, a^A) \equiv a^A$.

The agent's value (i.e., total reward) induced by a policy π and a deviation strategy ρ can be defined recursively via the value function as follows. For every $h = 1, \dots, H - 1$ and $\sigma \in \Sigma_{h-1}$:

$$V_h^{A, \pi, \rho}(\sigma) := \mathbb{E} \left[r_h^A(s, a^P, \tilde{a}^A) + V_{h+1}^{A, \pi, \rho}(\sigma; s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A) \right], \quad (1)$$

where the expectation is taken over $(s, \omega) \sim p_{h-1}(\cdot | \sigma)$ and $\mathbf{a} \sim \pi(\cdot | \sigma, \omega^P, \tilde{\omega}^A)$; moreover, $\tilde{\omega}^A = \rho(\sigma^A, \omega^A)$ and $\tilde{a}^A = \rho(\sigma^A, \omega^A, a^A)$, and by assumption $V_H^A(\sigma) \equiv 0$ for the last time step. The principal's value is defined the same way by changing the labels.

Our goal is to find a policy π that maximizes the principal's value under the agent's best response:

$$\max_{\pi, \rho} V_1^{P, \pi, \rho}(\emptyset) \quad (2)$$

$$\text{subject to } \rho \in \arg \max_{\rho'} V_1^{A, \pi, \rho'}(\emptyset) \quad (2-1)$$

In other words, we look for π and ρ that form a Stackelberg equilibrium. We say that π is ϵ -optimal if $V_1^{P, \pi, \rho}(\emptyset) \geq V^* - \epsilon$ for some ρ satisfying (2-1), where V^* denotes the optimal value of (2).

As we will demonstrate, under hindsight observability, it is without loss of optimality to consider policies that are IC (incentive compatible), which incentivize $\perp$ as an optimal response of the agent.

Definition 1 (IC policy). A policy π is IC if $V_1^{A, \pi, \perp}(\emptyset) \geq V_1^{A, \pi, \rho}(\emptyset)$ for every possible deviation plan ρ of the agent.

3 Computing an Optimal Policy

We compute a near-optimal policy by constructing the *inducible value sets*. The approach is similar to value-based approaches to solving MDPs, which operate by reasoning about the values of the states. However, it involves the following crucial differences:

1. Since we are in a two-player setting, both player's values need to be incorporated. We use a two-dimensional value vector instead of a scalar value.
2. Since we consider history-dependent policies, the game may proceed differently from the same state based on different histories. In this case, a single value (vector) is no longer sufficient for characterizing a state, unlike the case with stationary policies. We use the set consisting of all values can be induced by some valid policy to characterize a state. We refer to these sets are *inducible value sets*.

Moreover, since the actual state of the underlying MDP is not observable at the beginning of each time step, we will instead characterize the inducible value sets of *state-action pairs* $o \in O := S \times A$. Under hindsight observability, these pairs function as the states in a Markovian process, which the players observe before they each decide an action to play.

Definition 2 (Inducible value set). The inducible value set $\mathcal{V}_h(o) \subseteq \mathbb{R}^2$ of a state-action pair $o \in O := S \times A$ at time step h consists of all vectors $\mathbf{v} = (v^P, v^A)$ such that $v^P = V_h^{P, \pi, \rho}(\sigma)$ and $v^A = V_h^{A, \pi, \rho}(\sigma)$ for some policy π , deviation plan $\rho \in \arg \max_{\rho'} V_h^{A, \pi, \rho'}(\sigma)$, and sequence $\sigma \in \Sigma_{h-1}$ ending with o .

It is straightforward that once we obtain $\mathcal{V}_1(\emptyset)$, then the principal's optimal value in (2) can be computed by solving $\max_{(v^P, v^A) \in \mathcal{V}_1(\emptyset)} v^P$. We present a dynamic programming approach to constructing the inducible sets next. We will later also show that, once we obtain all the inducible sets, every inducible value vector can be efficiently unrolled into a history-dependent policy that induces it, and the policy can be executed efficiently on the fly. Hence, each value vector serves as a compact representation of a fully specified history-dependent policy.

3.1 Computing Inducible Value Sets

Instead of computing the exact value sets, we use convex polytopes to approximate them, so as to avoid possible exponential growth of the representational complexity. Let $\hat{\mathcal{V}}_h(o)$ denote the approximation of $\mathcal{V}_h(o)$ we aim to obtain. Recall that in the last time step all rewards are 0, so trivially we use $\hat{\mathcal{V}}_H(o) = \mathcal{V}_H(s) = \{(0, 0)\}$ for all $o \in O$ as the base case.

Dynamic Programming Now suppose that we have obtained the polytopes $\hat{\mathcal{V}}_{h+1}(o')$ for all $o' \in O$. We move to time step h and construct each $\hat{\mathcal{V}}_h(o)$ based on the $\hat{\mathcal{V}}_{h+1}(o')$'s. Central to the approach is the following characterization, which describes an IC condition at time step h : for every $\mathbf{v} \in \mathbb{R}^2$, it holds that $\mathbf{v} \in \mathcal{V}_h(o)$ if and only if there exist a one-step policy $\bar{\pi} : \Omega \rightarrow \Delta(A)$ and a set of onward value vectors $\{\mathbf{v}'(\bar{\sigma}) \in \mathbb{R}^2 : \bar{\sigma} \in \bar{\Sigma}\}$ that satisfy the following constraints.

1. **A value function constraint** based on (1), which expresses $\mathbf{v}$ via the immediate rewards and onward value vectors $\mathbf{v}'$ to be induced next, assuming truthful response of the agent:

$$\mathbf{v} = \sum_{s, \omega, \mathbf{a}} p_{h-1}(s, \omega | o) \cdot \bar{\pi}(\mathbf{a} | \omega) \cdot \left(\mathbf{r}_h(s, \mathbf{a}) + \mathbf{v}'(s, \omega, \omega^A, \mathbf{a}, a^A) \right), \quad (3)$$

The onward value vectors represent the subsequent part of the principal's commitment, which is contingent on the interaction $(s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A)$ in step h . Under the truthful response of the agent, we have $\tilde{\omega}^A = \omega^A$ and $\tilde{a}^A = a^A$ in (3).

2. **IC constraints**, which ensure that the agent's truthful behavior assumed in (3) is indeed incentivized, where we denote by $p_{h-1}(s, \omega^P | o, \omega^A) \propto p_{h-1}(s, \omega | o)$ the conditional probability defined by p_{h-1} . For all $\omega^A, \tilde{\omega}^A \in \Omega^A$,

$$\begin{aligned} & \sum_{s, \omega^P, \mathbf{a}} p_{h-1}(s, \omega^P | o, \omega^A) \cdot \bar{\pi}(\mathbf{a} | \omega) \cdot \left(r_h^A(s, \mathbf{a}) + v'^A(s, \omega, \omega^A, \mathbf{a}, a^A) \right) \geq \\ & \sum_{a^A} \max_{\tilde{a}^A \in A^A} \sum_{s, \omega^P, a^P} p_{h-1}(s, \omega^P | o, \omega^A) \cdot \bar{\pi}(\mathbf{a} | \omega^P, \tilde{\omega}^A) \cdot \left(r_h^A(s, a^P, \tilde{a}^A) + v'^A(s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A) \right). \end{aligned} \quad (4)$$

Namely, the constraint says, upon observing ω^A , the agent's expected payoff under their truthful response is at least as much as what they could have obtained, had they: 1) reported a different observation $\tilde{\omega}^A$, 2) performed a best action $\tilde{a}^A$ in response to every possible recommendation a^A of the principal, and 3) responded optimally in the subsequent time steps (whereby the onward values are given by $\mathbf{v}'$).

3. **Onward value constraints**, which ensures that the onward values given by $\mathbf{v}'$ are also inducible:

$$\mathbf{v}'(s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A) \in \mathcal{V}_{h+1}(s, a^P, \tilde{a}^A) \quad \text{for all } (s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A) \in \bar{\Sigma}. \quad (5)$$

The following lemma indicates the correctness of the above characterization.

Lemma 3. $\mathbf{v} \in \mathcal{V}_h(o)$ if and only if (3) to (5) hold for some $\bar{\pi} : \Omega \rightarrow \Delta(A)$ and $\mathbf{v}' : \bar{\Sigma} \rightarrow \mathbb{R}^2$.

Therefore, to decide whether $\mathbf{v} \in \mathcal{V}_h(o)$ amounts to deciding whether the above constraints are satisfied by some $\bar{\pi}$ and $\mathbf{v}'$ (highlighted in blue in the constraints). Note that since the inductive hypothesis assumes an approximation $\hat{\mathcal{V}}_{h+1}(o')$ instead of the exact set $\mathcal{V}_{h+1}(o')$, we will in fact impose the following *approximate* onward value constraint, instead of the exact version in (5):

$$\mathbf{v}'(s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A) \in \hat{\mathcal{V}}_{h+1}(s, a^P, \tilde{a}^A) \quad \text{for all } (s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A) \in \bar{\Sigma}. \quad (6)$$

Linearizing (3) and (4) The constraint satisfiability problem defined above is non-linear due to the quadratic terms and the maximization operator in (3) and (4). Nevertheless, it can be linearized as long as every polytope $\hat{\mathcal{V}}_{h+1}(o')$, $o' \in O$, is given by the *half-space representation*, i.e., by linear constraints in the form $\mathbf{H} \cdot \mathbf{x} \leq \mathbf{b}$ for some matrix $\mathbf{H}$ and vector $\mathbf{b}$. Due to the space limit, we leave the details in Appendix B.

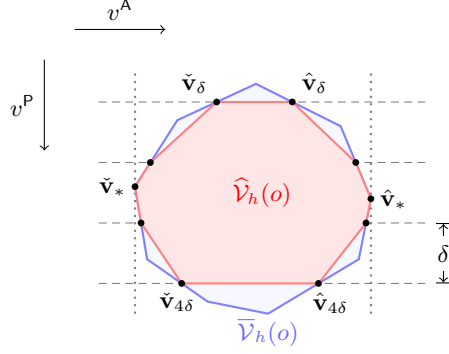


Figure 1: Constructing $\hat{\mathcal{V}}_h(o)$ as a δ -approximation of $\bar{\mathcal{V}}_h(o)$. The black points constitute $\mathcal{W}$.

For $h = H - 1, \dots, 1$, do the following for all $o \in O$:

1. Plug in (6) the half-space representation of $\hat{\mathcal{V}}_{h+1}(o')$, $o' \in O$. Then linearize (3) and (4).
2. Discretize the space $[0, H]^2$ into a finite point set (see Lemma 4 for more detail). Check the inducibility of each point $\mathbf{v}$ in this set by solving the linear constraint satisfiability problem defined by (the linearized version of) (3), (4) and (6).
3. Compute $\hat{\mathcal{V}}_h(o)$ as the convex hull of the inducible points obtained above, in half-space representation.

Figure 2: Computing approximate value polytopes via dynamic programming.

Constructing $\hat{\mathcal{V}}_h(o)$ As a result, we obtain a polytope $\mathcal{P}$ defined by a set of linear constraints equivalent to (3), (4) and (6). The projection of $\mathcal{P}$ onto the dimensions of $\mathbf{v}$ is (approximately) $\mathcal{V}_h(o)$. To ensure that the projection can be plugged back into (6) in the next induction step, we need the half-space representation of the projection, too. In particular, we want to eliminate the additional variables in the representation so that only $\mathbf{v}$ remains. (Otherwise, the number of variables may grow exponentially as the induction step increases.) This can be done approximately in polynomial time given that $\mathbf{v}$ is two-dimensional. Roughly speaking, we discretize the box $[0, H]^2$ into a finite set of points (recall that rewards in each time step are bounded in $[0, 1]$, so $[0, H]^2$ contains $\mathcal{V}_h(o)$), check the inducibility of each point, and compute the convex hull of the inducible points in half-space representation. The specific way we discretize the space (see Fig. 1) ensures that IC is satisfied *exactly* (which can otherwise not be achieved by using standard grid-based discretization). The details can be found in the proof of Lemma 4.

Repeating the induction procedure till $h = 1$, we obtain $\hat{\mathcal{V}}_1(\emptyset)$ as well as a near-optimal value of the principal by solving the LP $\max_{\mathbf{v} \in \hat{\mathcal{V}}_1(\emptyset)} v^P$. Fig. 2 summarizes this dynamic programming approach.

Lemma 4. *For any constant $\epsilon > 0$, it can be computed in time $\text{poly}(|S| \cdot |A| \cdot |\Omega|, H, 1/\epsilon)$ the half-space representations of a set of polytopes $\hat{\mathcal{V}}_h(o) \subseteq \mathcal{V}_h(o)$, $o \in O \cup \{\emptyset\}$ and $h = 1, \dots, H$, such that (3), (4) and (6) are satisfiable for every $\mathbf{v} \in \hat{\mathcal{V}}_h(o)$ and $\max_{\mathbf{v} \in \hat{\mathcal{V}}_1(\emptyset)} v^P \geq \max_{\mathbf{v} \in \mathcal{V}_1(\emptyset)} v^P - \epsilon$.*

Proof sketch. We construct the approximate sets $\hat{\mathcal{V}}_h(o)$ by backward induction. Assume for all o that $\hat{\mathcal{V}}_{h+1}(o)$ is defined by polynomially many constraints and is an ϵ -approximation of $\mathcal{V}_{h+1}(o)$. Let $\bar{\mathcal{V}}_h(o)$ be the feasible set defined by (3), (4) and (6) using $\hat{\mathcal{V}}_{h+1}(o')$. We discretize the principal's value into slices of width δ (see Fig. 1). We include the following points into a collection $\mathcal{W}$ use the convex hull of $\mathcal{W}$ as $\hat{\mathcal{V}}_h(o)$: 1) for every line w separating the slices, the two points $\check{\mathbf{v}}_w$ and $\hat{\mathbf{v}}_w$ at the intersection of w and the boundary of $\bar{\mathcal{V}}_{h+1}(o)$; and 2) the extreme points $\check{\mathbf{v}}_*$ and $\hat{\mathbf{v}}_*$ of $\bar{\mathcal{V}}_h(o)$ minimizing and maximizing the agent's value, respectively. Each of these points can be obtained

Input: a sequence $(\sigma; \omega^P, \tilde{\omega}^A)$, where $\sigma = (s_\ell, \omega_\ell, \tilde{\omega}_\ell^A, \mathbf{a}_\ell, \tilde{a}_\ell^A)_{\ell=1}^{h-1}$.

1. Initialize: $\mathbf{v} \leftarrow \arg \max_{\mathbf{v} \in \hat{\mathcal{V}}_1(\emptyset)} v^P$ and $o \leftarrow \emptyset$.
2. For $\ell = 1, \dots, h-1$:
 - Fix $\mathbf{v}$ and o , and solve (3), (4) and (6), where we use the polytopes $\hat{\mathcal{V}}_h(o)$ described in Lemma 4. Let the solution be $\bar{\pi}$ and $\mathbf{v}'$.
 - Update: $\mathbf{v} \leftarrow \mathbf{v}'(s_\ell, \omega_\ell, \tilde{\omega}_\ell^A, \mathbf{a}_\ell, \tilde{a}_\ell^A)$ and $o \leftarrow (s_\ell, a_\ell^P, a_\ell^A)$.
3. Output $\pi(\cdot | \sigma; \omega^P, \tilde{\omega}^A) = \bar{\pi}(\cdot | \omega^P, \tilde{\omega}^A)$.

Figure 3: Computing a near-optimal policy based on approximations of the value polytopes.

efficiently by solving a polynomial-size LP. The inclusion of the extreme points ensures in particular that we do not miss the agent’s extreme values, which is critical for achieving exact IC.

It can be verified that $\hat{\mathcal{V}}_h(o) \subseteq \bar{\mathcal{V}}_h(o)$ and that any $\mathbf{v} \in \bar{\mathcal{V}}_h(o)$ can be approximated by some $\mathbf{x} \in \hat{\mathcal{V}}_h(o)$ with $x^A = v^A$ and $x^P \geq v^P - \delta$. So $\hat{\mathcal{V}}_h(o)$ is a δ -approximation of $\bar{\mathcal{V}}_h(o)$ and $x^A = v^A$ ensures exact IC for the agent. Since $\bar{\mathcal{V}}_h(o)$ itself is an ε -approximation of $\mathcal{V}_h(o)$, we obtain an $(\varepsilon + \delta)$ -approximation overall. Choosing $\delta = \varepsilon/H$ and noting the trivial base case $\hat{\mathcal{V}}_H(o) = \{(0, 0)\}$, the construction yields $\hat{\mathcal{V}}_1(\emptyset)$ as an ε -approximation of $\mathcal{V}_1(\emptyset)$, computable in time $\text{poly}(|S| \cdot |A| \cdot |\Omega|, H, 1/\varepsilon)$. $\square$

3.2 Unrolling the Optimal Policy

The above procedure yields the maximum inducible value of the principal but not yet an optimal policy that achieves this value. We next demonstrate how to compute an optimal policy based on $\hat{\mathcal{V}}_1(\emptyset)$. Rather than obtaining an explicit description of a history-dependent policy π —which would be exponentially large as the policy specifies a distribution for each possible sequence—we present an efficient procedure that computes the distribution $\pi(\cdot | \sigma; \omega^P, \tilde{\omega}^A)$ for any given sequence $(\sigma; \omega^P, \tilde{\omega}^A)$. This means that, when playing the game, the principal can compute an optimal policy on the fly based on the realized history.

We use a forward computation procedure presented in Fig. 3. Starting from time step 1, the procedure repeatedly computes a one-step policy $\bar{\pi}$ and a set of onward vectors, to induce the target value vector $\mathbf{v}$. The onward vectors define the target values to be induced in the next time step, contingent on the interaction in the current, which is given by σ . Hence, the target vector is updated to one of the onward vectors according to σ at the end of each iteration. In other words, in each time step, we expand the target vector into a set of onward vectors, and then select one of them as the next target vector according to the realized interaction given by σ . This leads to the following main result.

Theorem 5. *There exists an ε -optimal IC policy π such that, for any given sequence $(\sigma; \omega^P, \tilde{\omega}^A) \in \Sigma \times \Omega$, the distribution $\pi(\cdot | \sigma; \omega^P, \tilde{\omega}^A)$ can be computed in time $\text{poly}(|S| \cdot |A| \cdot |\Omega|, H, 1/\varepsilon)$.*

4 Learning to Commit

We now turn to an episodic online learning setting where the transition model $p : S \times A \rightarrow \Delta(S \times \Omega)$ is not known to the players beforehand. Let there be T episodes. At the beginning of each episode, the principal commits to a new policy based on the outcomes of the previous episodes. Each episode proceeds in H time steps the same way as the model defined in Section 2.

We present a learning algorithm that guarantees sublinear regrets for both players under hindsight observability. The algorithm is *centralized* and relies on the agent behaving truthfully. It does not guarantee exact IC during the course of learning but IC in the limit when the number of episodes approaches infinity. Indeed, since the model is unknown to both players, IC in the limit is a more relevant concept as the agent cannot decide how to optimally deviate from their truthful response, either. In this case, the sublinear regret the algorithm guarantees for the agent should in many

scenarios be sufficient for incentivizing for the agent to participate and follow the centralized learning protocol.

The players' regrets are defined as follows:

$$\text{Reg}^P = \sum_{t=1}^T \left(V^* - V_1^{P, \pi_t, \perp}(\emptyset) \right) \quad \text{and} \quad \text{Reg}^A = \sum_{t=1}^T \left(\max_{\rho} V_1^{A, \pi_t, \rho}(\emptyset) - V_1^{A, \pi_t, \perp}(\emptyset) \right),$$

where V^* is the optimal value of (2) and π_t denotes the policy the principal commits to in the t -th episode. In words, the principal's regret Reg^P is defined with respect to the optimal policy under the true model. The agent's regret Reg^A is defined with respect to his optimal response to each π_t , which is a dynamic regret as the benchmark changes across the episodes.

4.1 Learning Algorithm

Reward-free Exploration Our learning algorithm is based on *reward-free exploration*, which is an RL paradigm where learning happens before a reward function is provided [21]. It has been shown in a series of works that efficient learning is possible under this paradigm [21, 23, 30]. In particular, we will use the sample complexity bound in Lemma 6. At a high level, our algorithm proceeds by first conducting reward-free exploration to learn a sufficiently accurate estimate of the true model. Based on the estimate we then solve a relaxed version of the policy optimization problem (2) to obtain a policy. Using this policy in the remaining episodes guarantees sublinear regret for both players.

Lemma 6 ([21, Lemma 3.6 restated]). *Consider an (single-player) MDP (S, A, p) (without any reward function specified) with horizon length H . There exists an algorithm which learns a model $\hat{p}$ after $\tilde{O}\left(\frac{H^5 |S|^2 \cdot |A|}{\delta^2}\right)$ episodes of exploration, such that with probability at least $1 - q$, for any reward function r and policy π , it holds that $|V_1^{r, \pi}(s) - \hat{V}_1^{r, \pi}(s)| \leq \delta/2$ for all states s , where $V_1^{r, \pi}$ and $\hat{V}_1^{r, \pi}$ denote the value functions under reward function r and models p and $\hat{p}$, respectively.²*

With the above result, we can learn a model $\hat{p}$ for our purpose. In what follows, we let $\hat{V}_h^{P, \pi, \rho}$ and $\hat{V}_h^{A, \pi, \rho}$ denote the players' value functions in model $\hat{p}$ (i.e., by replacing p in (1) with $\hat{p}$). Lemma 7 then translates Lemma 6 to our setting. Note that under hindsight observability the process facing the principal and the agent jointly during the learning process is effectively an MDP, where the effective state space is $O \times \Omega$. An effective state, say $\theta = (s, \mathbf{a}, \omega)$, consists of the state-action pair $(s, \mathbf{a})$ in the previous step and the observations ω in the current. When a joint action $\mathbf{a}'$ is performed, θ transitions to $\theta' = (s', \mathbf{a}', \omega')$ with probability $p_{h-1}(s', \omega' | s, \mathbf{a})$.

Lemma 7. *A model $\hat{p}$ can be learned after $\tilde{O}\left(H^5 |S|^2 |A|^3 |\Omega|^2 / \delta^2\right)$ episodes of exploration, such that $|V_1^{A, \pi, \rho}(\emptyset) - \hat{V}_1^{A, \pi, \rho}(\emptyset)| \leq \delta/2$ and $|V_1^{P, \pi, \rho}(\emptyset) - \hat{V}_1^{P, \pi, \rho}(\emptyset)| \leq \delta/2$ with probability at least $1 - q$ for any policy π and deviation plan ρ .*

Therefore, the value functions change smoothly as the learned model $\hat{p}$ approaches p . However, this smoothness is insufficient for deriving a sublinear bound on the principal's regret because of the agent's incentive constraints in our problem. Roughly speaking, the set of IC policies does not change smoothly with $\hat{p}$, even though the value functions do. Hence, even an infinitesimal difference between $\hat{p}$ and p may lead to a jump between the IC policy sets under these two models and, in turn, a gap between the values of the optimal policies.

Approximate IC Relaxation To deal with this issue, we relax the incentive constraints, allowing small violations to the constraints. Such violations are inevitable if we aim to achieve a near-optimal value under the true model p but only know an estimate $\hat{p}$ of the true model. On the positive side, given the sublinear regret guarantee for the agent, the violation diminishes with the number of episodes. We define δ -IC policies below.

²The notation $\tilde{O}$ omits logarithmic factors. In the original statement of Jin et al. [21], π is non-stationary (time-dependent) but independent of the history. However, the proof of the lemma also applies to history-dependent policies. The dependence on H in the sample complexity can be further improved with better reward-free exploration algorithms [23, 30], but this is not a focus of ours.

Definition 8 (δ -IC policy). A policy π is δ -IC (w.r.t. model $\hat{p}$) if $\hat{V}_1^{A,\pi,\perp}(\emptyset) \geq \hat{V}_1^{A,\pi,\rho}(\emptyset) - \delta$ for every possible deviation plan ρ of the agent. A δ -IC policy is said to be ϵ -optimal if $\hat{V}_1^{P,\pi,\perp}(\emptyset) \geq V^* - \epsilon$, where V^* is the optimal value of (2) (under p).

That is, in response to a δ -IC policy, the agent can improve his overall expected payoff by no more than δ if he deviates from the truthful response. We assume that the agent will not deviate for such a small benefit, and we evaluate the value of a δ -IC policy based on the agent's truthful response. This is how the ϵ -optimality is defined above, where we compare against the optimal value V^* in (2), which is obtained under a more stringent setting without any relaxation of the agent's incentive. In other words, we relax the feasible space and compare the solution obtained in this relaxed space with the optimum over the smaller original feasible space. Such relaxations are common in the optimization literature, and they are crucial for resolving the non-smooth issue.

Let $\hat{\Pi}_\delta$ and Π_δ denote the set of δ -IC policies under $\hat{p}$ and p , respectively. The relaxation immediately results in $\hat{\Pi}_\delta \supseteq \Pi_0$ for the model $\hat{p}$ stated in Lemma 7. As a result, optimizing over $\hat{\Pi}_\delta$ ensures that the optimal value yielded is as much (up to a small error) as the optimal value V^* over Π_0 . Meanwhile, the value loss introduced by this relaxation for the agent is also small (bounded by δ).

With the above results, our learning algorithm proceeds as follows.

1. Run reward-free exploration to obtain a model $\hat{p}$ as stated in Lemma 7.
2. Compute a δ -optimal δ -IC policy in $\hat{p}$ and use it in the remaining episodes.

The near-optimal policy in Step 2 can be computed efficiently according to Lemma 9, via an approach similar to the one in Section 3.1. This gives an efficient algorithm with sublinear regrets for both players. We present Theorem 10.

Lemma 9. *There exists an ϵ -optimal δ -IC policy π such that, for any given sequence $(\sigma; \omega^P, \tilde{\omega}^A) \in \Sigma \times \Omega$, the distribution $\pi(\cdot | \sigma; \omega^P, \tilde{\omega}^A)$ can be computed in time $\text{poly}(|S| \cdot |A| \cdot |\Omega|, H, 1/\epsilon, \log(1/\delta))$.*

Theorem 10. *There exists an algorithm that guarantees regret $\tilde{O}(\zeta^{1/3} T^{2/3})$ for both players with probability $1 - q$, where $\zeta = H^5 |S|^2 |A|^3 |\Omega|^2$. The computation involved in implementing the algorithm takes time $\text{poly}(|S| \cdot |A| \cdot |\Omega|, H, T)$.*

5 Conclusion

We studied a stochastic principal-agent framework and presented efficient computation and learning algorithms. Our model can be further extended to the setting with n agents. The algorithms we presented remain efficient for any constant n if approximate IC solutions are considered. Computing optimal exact IC policies for n agents remain an interesting open question, as our discretization method, which operates by slicing the space, does not generalize to n agents. When n is not a constant, representing games in normal-form requires space exponential in n , so more succinct representations are typically considered. However, it is known that in succinctly represented games even to compute an optimal correlated equilibrium in one-shot games may be NP-hard [35].) Our results indicate how a policy designer might interact with agents optimally. In particular implementations, the designer's incentives may not be aligned with societal benefits. In these cases, a careful analysis of the incentives and their moral legitimacy must be considered. Besides this, since the paper is theory focused, we do not feel any other potential impacts must be specifically highlighted here.

Acknowledgement

The work of Goran Radanovic was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 467367360.

References

- [1] Robert J. Aumann and Michael B. Maschler. *Repeated Games with Incomplete Information*. MIT Press, 1995.
- [2] Martino Bernasconi, Matteo Castiglioni, Alberto Marchesi, Nicola Gatti, and Francesco Trovò. Sequential information design: learning to persuade in the dark. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS'22)*, 2022.
- [3] Martino Bernasconi, Matteo Castiglioni, Alberto Marchesi, and Mirco Mutti. Persuading farsighted receivers in MDPs: the power of honesty. *Advances in Neural Information Processing Systems (NeurIPS'23)*, 36, 2024.
- [4] Patrick Bolton and Mathias Dewatripont. *Contract Theory*. Mit Press, 2004.
- [5] Branislav Bošanský, Simina Brânzei, Kristoffer Arnsfelt Hansen, Troels Bjerre Lund, and Peter Bro Miltersen. Computation of stackelberg equilibria of finite sequential games. *ACM Transactions on Economics and Computation (TEAC)*, 5(4):1–24, 2017.
- [6] Federico Cacciamani, Matteo Castiglioni, and Nicola Gatti. Online mechanism design for information acquisition. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*, 2023.
- [7] Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Bayesian persuasion meets mechanism design: Going beyond intractability with type reporting. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS'22)*, page 226–234, 2022.
- [8] Andrea Celli, Stefano Coniglio, and Nicola Gatti. Private bayesian persuasion with sequential games. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'20)*, volume 34, pages 1886–1893, 2020.
- [9] Timothy M Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & Computational Geometry*, 16(4):361–368, 1996.
- [10] Natalie Collina, Eshwar Ram Arunachaleswaran, and Michael Kearns. Efficient stackelberg strategies for finitely repeated games. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS'23)*, page 643–651, 2023.
- [11] Liam Dermed and Charles Isbell. Solving stochastic games. *Advances in Neural Information Processing Systems (NIPS'09)*, 22, 2009.
- [12] Francoise Forges. An approach to communication equilibria. *Econometrica: Journal of the Econometric Society*, pages 1375–1385, 1986.
- [13] Jiarui Gan and Rupak Majumdar. Value-set iteration: Computing optimal correlated equilibria in infinite-horizon multi-player stochastic games, 2025. URL <https://arxiv.org/abs/2506.07186>.
- [14] Jiarui Gan, Minbiao Han, Jibang Wu, and Haifeng Xu. Optimal coordination in generalized principal-agent problems: A revisit and extensions. *arXiv preprint arXiv:2209.01146*, 2022.
- [15] Jiarui Gan, Rupak Majumdar, Goran Radanovic, and Adish Singla. Bayesian persuasion in sequential decision-making. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI'22)*, pages 5025–5033, 2022.
- [16] Jiarui Gan, Rupak Majumdar, Goran Radanovic, and Adish Singla. Sequential decision making with information asymmetry. In *33rd International Conference on Concurrency Theory (CONCUR'22)*, volume 243 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:18, 2022.
- [17] Jiarui Gan, Rupak Majumdar, Debmalya Mandal, and Goran Radanovic. Sequential principal-agent problems with communication: Efficient computation and learning, 2023. URL <https://arxiv.org/abs/2306.03832v1>.

- [18] Keegan Harris, Hoda Heidari, and Steven Z Wu. Stateful strategic regression. *Advances in Neural Information Processing Systems*, 34:28728–28741, 2021.
- [19] Dima Ivanov, Paul Dütting, Inbal Talgam-Cohen, Tonghan Wang, and David C. Parkes. Principal-agent reinforcement learning, 2024. URL <https://arxiv.org/abs/2407.18074>.
- [20] Michael C. Jensen and William H. Meckling. Theory of the firm: Managerial behavior, agency costs and ownership structure. *Journal of Financial Economics*, 3(4):305–360, 1976.
- [21] 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.
- [22] Emir Kamenica and Matthew Gentzkow. Bayesian persuasion. *American Economic Review*, 101(6):2590–2615, 2011.
- [23] Emilie Kaufmann, Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Edouard Leurent, and Michal Valko. Adaptive reward-free exploration. In *Algorithmic Learning Theory*, pages 865–891. PMLR, 2021.
- [24] Jonathan Lee, Alekh Agarwal, Christoph Dann, and Tong Zhang. Learning in POMDPs is sample-efficient with hindsight observability. In *International Conference on Machine Learning*, pages 18733–18773. PMLR, 2023.
- [25] 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 (EC’10)*, pages 83–92, 2010.
- [26] Joshua Letchford, Liam MacDermid, Vincent Conitzer, Ronald Parr, and Charles L. Isbell. Computing optimal strategies to commit to in stochastic games. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI’12)*, page 1380–1386, 2012.
- [27] Lars Ljungqvist and Thomas J. Sargent. *Recursive Macroeconomic Theory, fourth edition*. Mit Press, 2018.
- [28] Liam MacDermid, Karthik Sankaran Narayan, Charles Lee Isbell Jr., and Lora Weiss. Quick polytope approximation of all correlated equilibria in stochastic games. In Wolfram Burgard and Dan Roth, editors, *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI’11)*, 2011.
- [29] Miltiadis Makris. The theory of incentives: The principal-agent model. *The Economic Journal*, 113(488):F394–F395, 2003.
- [30] Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Emilie Kaufmann, Edouard Leurent, and Michal Valko. Fast active learning for pure exploration in reinforcement learning. In *International Conference on Machine Learning*, pages 7599–7608. PMLR, 2021.
- [31] Jean-Francois Mertens, Sylvain Sorin, and Shmuel Zamir. *Repeated Games (Econometric Society Monographs Book 55)*. Cambridge University Press, 2015.
- [32] Robert Milgrom and John Roberts. Relying on the information of interested parties. *Rand Journal of Economics*, 17:18–32, 1986.
- [33] Roger B Myerson. Optimal coordination mechanisms in generalized principal–agent problems. *Journal of mathematical economics*, 10(1):67–81, 1982.
- [34] Roger B Myerson. Multistage games with communication. *Econometrica: Journal of the Econometric Society*, pages 323–358, 1986.
- [35] Christos H Papadimitriou and Tim Roughgarden. Computing equilibria in multi-player games. In *SODA*, volume 5, pages 82–91. Citeseer, 2005.
- [36] Christos H Papadimitriou and John N Tsitsiklis. The complexity of Markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

- [37] Stephen A. Ross. The economic theory of agency: The principal’s problem. *The American economic review*, 63(2):134–139, 1973.
- [38] Tuomas Sandholm. Automated mechanism design: A new application area for search algorithms. In *Principles and Practice of Constraint Programming—CP 2003: 9th International Conference, CP 2003, Kinsale, Ireland, September 29–October 3, 2003. Proceedings 9*, pages 19–36. Springer, 2003.
- [39] Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10): 1095–1100, 1953.
- [40] Heinrich Von Stackelberg. *Market structure and equilibrium*. Springer Science & Business Media, 2010.
- [41] Jibang Wu, Zixuan Zhang, Zhe Feng, Zhaoran Wang, Zhuoran Yang, Michael I. Jordan, and Haifeng Xu. Sequential information design: Markov persuasion process and its efficient reinforcement learning. In *Proceedings of the 23rd ACM Conference on Economics and Computation (EC’22)*, page 471–472, 2022.
- [42] Jibang Wu, Siyu Chen, Mengdi Wang, Huazheng Wang, and Haifeng Xu. Contractual reinforcement learning: Pulling arms with invisible hands, 2024. URL <https://arxiv.org/abs/2407.01458>.
- [43] Brian Zhang and Tuomas Sandholm. Polynomial-time optimal equilibria with a mediator in extensive-form games. *Advances in Neural Information Processing Systems*, 35:24851–24863, 2022.
- [44] Brian Zhang, Gabriele Farina, Ioannis Anagnostides, Federico Cacciamani, Stephen McAleer, Andreas Haupt, Andrea Celli, Nicola Gatti, Vincent Conitzer, and Tuomas Sandholm. Computing optimal equilibria and mechanisms via learning in zero-sum extensive-form games. *Advances in Neural Information Processing Systems*, 36, 2023.
- [45] Hanrui Zhang and Vincent Conitzer. Automated dynamic mechanism design, 2021.
- [46] Hanrui Zhang, Yu Cheng, and Vincent Conitzer. Efficiently solving turn-taking stochastic games with extensive-form correlation. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC’23)*, pages 1161–1186, 2023.

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: We outline the key claims and contributions of this work in the abstract and in the introduction section. All claims are supported by rigorous proofs.

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 mentioned in the main paper limitations of our results and assumptions required. Additionally, we also discussed potential future directions to address these limitations in the conclusion 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: Every theoretical result is supported by rigorous proofs, with all details provided in the appendix.

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 paper 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: This paper does not include experiments.

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: This 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: This 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: This 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: This research conforms to 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: [Yes]

Justification: We discussed potential positive and negative societal impacts in the conclusion section. In particular implementations, the designer's incentives may not be aligned with societal benefits. In these cases, a careful analysis of the incentives and their moral legitimacy must be considered.

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: [NA]

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: [NA]

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: [NA]

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: [NA]

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: [NA]

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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: [NA]

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 Omitted Proofs

A.1 Omitted Proofs in Section 3

For simplicity, we write $\vec{V}_h^{\pi, \rho}(\sigma) = (V_h^{\text{P}, \pi, \rho}(\sigma), V_h^{\text{A}, \pi, \rho}(\sigma))$ in the following proofs.

Lemma 3. $\mathbf{v} \in \mathcal{V}_h(o)$ if and only if (3) to (5) hold for some $\bar{\pi} : \Omega \rightarrow \Delta(A)$ and $\mathbf{v}' : \bar{\Sigma} \rightarrow \mathbb{R}^2$.

Proof. First, consider the “only if” direction of the statement. Suppose that $\mathbf{v} \in \mathcal{V}_h(o)$. By definition, we have $\mathbf{v} = \vec{V}_h^{\pi, \rho}(o)$ for some $\pi, \rho \in \arg \max_{\rho'} \vec{V}_h^{\text{A}, \pi, \rho'}(\sigma)$, and $\sigma \in \Sigma_{h-1}$ ending with o . According to a standard revelation principle argument, we can assume w.l.o.g. that ρ is IC in step h . Hence, by (1), we have

$$\mathbf{v} = \sum_{s, \omega, \mathbf{a}} p_{h-1}(s, \omega | o) \cdot \pi(\mathbf{a} | \sigma; \omega) \cdot \left(\mathbf{r}_h(s, \mathbf{a}) + \vec{V}_{h+1}^{\pi, \rho}(\sigma; s, \omega, \omega^{\text{A}}, \mathbf{a}, a^{\text{A}}) \right). \quad (7)$$

Letting $\bar{\pi}(\mathbf{a} | \omega) = \pi(\mathbf{a} | \sigma; \omega)$ for every $\omega \in \Omega$, and $\mathbf{v}'(\bar{\sigma}) = \vec{V}_{h+1}^{\pi, \rho}(\sigma; \bar{\sigma})$ for every $\bar{\sigma} \in \bar{\Sigma}$, we establish (3). (Recall that $\bar{\Sigma} := S \times \Omega \times \Omega^{\text{A}} \times A \times A^{\text{A}}$ denotes the set of all possible interactions within one time step.)

Since ρ is IC in step h , (4) also follows immediately: the agent cannot benefit from any possible deviation. By definition, we have $\mathbf{v}'(\bar{\sigma}) = \vec{V}_{h+1}^{\pi, \rho}(\sigma; \bar{\sigma}) \in \mathcal{V}_{h+1}(o')$ for every $\bar{\sigma} \in \bar{\Sigma}$ that contains o' , so (5) holds.

Now consider the “if” direction. Suppose that (3) to (5) hold for some $\bar{\pi}$ and $\mathbf{v}'$. Pick arbitrary $\sigma \in \Sigma_{h-1}$ that ends with o . Consider a policy π such that: $\pi(\mathbf{a} | \sigma; \omega) = \bar{\pi}(\mathbf{a} | \omega)$ for all $\omega \in \Omega$, and $\pi(\mathbf{a} | \sigma; \bar{\sigma}; \omega) = \pi'(\mathbf{a} | \sigma; \bar{\sigma}; \omega)$ for all $\bar{\sigma} \in \bar{\Sigma}$ and $\omega \in \Omega$, where π' is an arbitrary policy that induces $\mathbf{v}'(\bar{\sigma})$ for every $\bar{\sigma}$ (which exists given (5)). Namely, π is the same as $\bar{\pi}$ in step h and switches to π' in the subsequent steps. Given (4), the agent cannot benefit from any deviation at step h , so (3) gives the players’ values for π and an optimal deviation plan of the agent. Hence, $\mathbf{v} \in \mathcal{V}_h(o)$. $\square$

Lemma 4. For any constant $\epsilon > 0$, it can be computed in time $\text{poly}(|S| \cdot |A| \cdot |\Omega|, H, 1/\epsilon)$ the half-space representations of a set of polytopes $\hat{\mathcal{V}}_h(o) \subseteq \mathcal{V}_h(o)$, $o \in O \cup \{\emptyset\}$ and $h = 1, \dots, H$, such that (3), (4) and (6) are satisfiable for every $\mathbf{v} \in \hat{\mathcal{V}}_h(o)$ and $\max_{\mathbf{v} \in \hat{\mathcal{V}}_1(\emptyset)} v^{\text{P}} \geq \max_{\mathbf{v} \in \mathcal{V}_1(\emptyset)} v^{\text{P}} - \epsilon$.

Proof. Throughout the proof, we say that the polytope $\hat{\mathcal{V}}_h(o)$ is an ϵ -approximation of $\mathcal{V}_h(o)$ if and only if:

- $\hat{\mathcal{V}}_h(o) \subseteq \mathcal{V}_h(o)$, and
- for every $\mathbf{v} \in \mathcal{V}_h(o)$, there exists $\mathbf{v}' \in \hat{\mathcal{V}}_h(o)$ such that $v'^{\text{P}} \geq v^{\text{P}} - \epsilon$ and $v'^{\text{A}} = v^{\text{A}}$.

We will show that an ϵ -approximation $\hat{\mathcal{V}}_1(\emptyset)$ of $\mathcal{V}_1(\emptyset)$ can be computed efficiently, so that $\max_{\mathbf{v} \in \hat{\mathcal{V}}_1(\emptyset)} v^{\text{P}} \geq \max_{\mathbf{v} \in \mathcal{V}_1(\emptyset)} v^{\text{P}} - \epsilon$ follows readily.³ Meanwhile, we also show that the polytopes we compute ensures that (3), (4) and (6) are satisfiable for every $\mathbf{v} \in \hat{\mathcal{V}}_h(o)$.

We now prove by induction. The key is the following induction step. Suppose that the following conditions hold for all $o \in O$:

1. $\hat{\mathcal{V}}_{h+1}(o)$ is defined by $\mathcal{O}(H/\delta)$ many linear constraints.
2. $\hat{\mathcal{V}}_{h+1}(o)$ is an ϵ -approximation of $\mathcal{V}_{h+1}(o)$.

³In the definition of ϵ -approximation, we require additionally that the projections of $\hat{\mathcal{V}}_h(o)$ and $\mathcal{V}_h(o)$ onto the dimension of v^{A} are the same (i.e., $v'^{\text{A}} = v^{\text{A}}$), so that the approximation compromises only on the principal’s value. This is crucial for ensuring exact IC and smooth changes of the approximation throughout the induction process we present below.

We show that, given the above conditions, for every $o \in O$ we can compute in time polynomial in $1/\delta$ a polytope $\widehat{\mathcal{V}}_h(o)$ (in half-space representation) that satisfies the above conditions (for h), with an approximation factor $\varepsilon' = \varepsilon + \delta$ in the second condition. Once this holds, picking $\delta = \varepsilon/H$ then gives, by induction, that $\widehat{\mathcal{V}}_1(\emptyset)$ is an ε -approximation of $\mathcal{V}_1(\emptyset)$ (where ε is the target constant in the statement of the lemma). Note that as a based case, $\{(0, 0)\}$ is readily a 0-approximation of $\mathcal{V}_H(o)$ and can be defined by three linear constraints.

We proceed as follows. For every $o \in O$, let $\overline{\mathcal{V}}_h(o)$ denote the set of vectors $\mathbf{v}$ satisfying (3), (4) and (6).⁴ We follow the algorithm presented in Fig. 2 and discretize $[0, H]^2$ to construct $\widehat{\mathcal{V}}_h(o)$. Specifically, we slice the space along the dimension of the principal's value. We compute the intersection points of the slice lines and (the boundary of) $\overline{\mathcal{V}}_h(o)$, and construct $\widehat{\mathcal{V}}_h(o)$ as the convex hull of the intersection points to approximate $\overline{\mathcal{V}}_h(o)$. Specifically, let $W = \{0, \delta, 2\delta, \dots, H - \delta, H\}$ contain the principal's values on the slice lines we use, and let $\mathcal{W}$ be the set consisting of the following points.

- First, for each $w \in W$, the two intersection points of the slice line at w and $\overline{\mathcal{V}}_h(o)$:

$$\check{\mathbf{v}}_w \in \arg \min_{\mathbf{v} \in \overline{\mathcal{V}}_h(o): v^P = w} v^A \quad \text{and} \quad \hat{\mathbf{v}}_w \in \arg \max_{\mathbf{v} \in \overline{\mathcal{V}}_h(o): v^P = w} v^A.$$

- Moreover, two vertices of $\overline{\mathcal{V}}_h(o)$ with the minimum and maximum values for the agent:

$$\check{\mathbf{v}}_* \in \arg \min_{\mathbf{v} \in \overline{\mathcal{V}}_h(o)} v^A \quad \text{and} \quad \hat{\mathbf{v}}_* \in \arg \max_{\mathbf{v} \in \overline{\mathcal{V}}_h(o)} v^A.$$

If there are multiple maximum (or minimum) vertices, we pick an arbitrary one.

An illustration is given in Fig. 1.

It shall be clear that the choice of these points ensures that we can approximate any inducible value vector with at most δ compromise on the principal's value and no compromise on the agent's. (In particular, the inclusion of $\check{\mathbf{v}}_*$ and $\hat{\mathbf{v}}_*$ ensures that we do not miss the agent's extreme values that may not be attained at any of the slice lines.) All the points can be computed efficiently by solving LPs that minimizes (or maximizes) v^A (where we also treat $\mathbf{v}$ as variables in addition to the other variables), subject to the linearized version of (3), (4) and (6), and additionally $v^P = w$ when we compute $\check{\mathbf{v}}_w$ or $\hat{\mathbf{v}}_w$. The hypothesis that $\widehat{\mathcal{V}}_{h+1}(o)$ is defined by $\mathcal{O}(H/\delta)$ linear constraints ensures that all the LPs are polynomial sized and hence can be solved efficiently.

We then compute $\widehat{\mathcal{V}}_h(o)$ by taking the convex hull of $\mathcal{W}$. Given that the space is two-dimensional, this can be done efficiently via standard algorithms in computational geometry (e.g., Chan's algorithm [9]). This way, the first condition in the inductive hypothesis holds for $\widehat{\mathcal{V}}_h(o)$ because $\widehat{\mathcal{V}}_h(o)$ has at most $\mathcal{O}(H/\delta)$ vertices while it is in $\mathbb{R}^2$. Meanwhile, $\widehat{\mathcal{V}}_h(o)$ is an δ -approximation of $\overline{\mathcal{V}}_h(o)$ according to the following arguments.

Claim 1. $\widehat{\mathcal{V}}_h(o)$ is an δ -approximation of $\overline{\mathcal{V}}_h(o)$.

Proof of Claim 1. First, since $\mathcal{W} \subseteq \overline{\mathcal{V}}_h(o)$ by construction, $\widehat{\mathcal{V}}_h(o) \subseteq \overline{\mathcal{V}}_h(o) \subseteq \mathcal{V}_h(o)$ holds readily. It remains to show that for any $\mathbf{v} \in \overline{\mathcal{V}}_h(o)$ there exists $\mathbf{x} \in \widehat{\mathcal{V}}_h(o)$ such that $x^A = v^A$ and $x^P \geq v^P - \delta$.

Let $\mathcal{B} = \{\mathbf{v}' \in \mathbb{R}^2 : i\delta \leq v'^P \leq (i+1)\delta\}$ be the band between two slice lines that contains $\mathbf{v}$. Consider the relation between v^A and the agent's minimum and maximum values attained at $\mathcal{W} \cap \mathcal{B}$. There can be the following possibilities.

- Case 1. v^A lies in between the minimum and maximum values, i.e.,

$$\min_{\mathbf{v}' \in \mathcal{W} \cap \mathcal{B}} v'^A \leq v^A \leq \max_{\mathbf{v}' \in \mathcal{W} \cap \mathcal{B}} v'^A.$$

This means that there must be a point $\mathbf{x} \in \text{ConvexHull}(\mathcal{W} \cap \mathcal{B})$ such that $x^A = v^A$. We have $\mathbf{x} \in \text{ConvexHull}(\mathcal{W} \cap \mathcal{B}) \subseteq \mathcal{B}$. So both $\mathbf{v}$ and $\mathbf{x}$ are inside $\mathcal{B}$. According to the definition of $\mathcal{B}$, this means $x^P \geq v^P - \delta$, as desired.

⁴Note that $\overline{\mathcal{V}}_h(o)$ is different from $\mathcal{V}_h(o)$: the latter, according to Lemma 3, is defined by (3) to (5), where (5) uses the exact value sets $\mathcal{V}_{h+1}(o')$, unlike the approximate ones $\widehat{\mathcal{V}}_{h+1}(o')$ in (6).

- Case 2. $v^A < \min_{\mathbf{v}' \in \mathcal{W} \cap \mathcal{B}} v'^A$. In this case, it must be that $\tilde{\mathbf{v}}_* \notin \mathcal{B}$ (otherwise, $\min_{\mathbf{v}' \in \mathcal{W} \cap \mathcal{B}} v'^A = \tilde{v}_*^A \leq v^A$). Now that $\mathbf{v} \in \mathcal{B}$, the line segment between $\mathbf{v}$ and $\tilde{\mathbf{v}}_*$ must intersect with the boundary of $\mathcal{B}$ (i.e., one of the slice lines) at some point $\mathbf{y}$. We have $y^A \leq v^A$ (because $\tilde{v}_*^A \leq v^A$ by definition) and $\mathbf{y} \in \bar{\mathcal{V}}_h(o)$ (because $\mathbf{v}, \tilde{\mathbf{v}}_* \in \bar{\mathcal{V}}_h(o)$). Pick $\tilde{\mathbf{v}}_w$ where $w = y^P$. By definition $\tilde{v}_w^A \leq y^A$. It follows that

$$\tilde{v}_w^A \leq y^A \leq v^A < \min_{\mathbf{v}' \in \mathcal{W} \cap \mathcal{B}} v'^A.$$

This is a contradiction because we have $\tilde{\mathbf{v}}_w \in \mathcal{W} \cap \mathcal{B}$ as $\mathbf{y}$ is on the boundary of $\mathcal{B}$.

- Case 3. $v^A > \max_{\mathbf{v}' \in \mathcal{W} \cap \mathcal{B}} v'^A$. An argument similar to that for Case 2 implies that this case is not possible, either.

Hence, only Case 1 is possible, where a desired point $\mathbf{x}$ exists. The claim then follows. $\square$

The fact that $\hat{\mathcal{V}}_h(o) \subseteq \bar{\mathcal{V}}_h(o)$ also implies that (3), (4) and (6) are satisfiable for every $\mathbf{v} \in \hat{\mathcal{V}}_h(o)$, as they are for every $\mathbf{v} \in \bar{\mathcal{V}}_h(o)$. We next confirm that $\hat{\mathcal{V}}_h(o)$ is eventually an $(\varepsilon + \delta)$ -approximation of $\mathcal{V}_h(o)$. Indeed, now Claim 1 indicates that $\hat{\mathcal{V}}_h(o)$ is an δ -approximation of $\bar{\mathcal{V}}_h(o)$, so $\hat{\mathcal{V}}_h(o)$ is an $(\varepsilon + \delta)$ -approximation of $\mathcal{V}_h(o)$ as long as $\bar{\mathcal{V}}_h(o)$ is an ε -approximation of $\mathcal{V}_h(o)$.

To see that $\bar{\mathcal{V}}_h(o)$ is an ε -approximation, consider an arbitrary $\mathbf{v} \in \mathcal{V}_h(o)$. By Lemma 3, $\mathbf{v}$ can be induced by some $\bar{\pi}$ and $\mathbf{v}'$ satisfying (3) to (5). By assumption, every $\hat{\mathcal{V}}_{h+1}(o')$ is an ε -approximation of $\mathcal{V}_{h+1}(o')$, so for every onward vector $\mathbf{v}'(\bar{o}) \in \mathcal{V}_{h+1}(o')$, there exists a vector $\tilde{\mathbf{v}}'(\bar{o}) \in \hat{\mathcal{V}}_{h+1}(o')$ such that $\tilde{v}'^P(\bar{o}) \geq v'^P(\bar{o}) - \varepsilon$ and $\tilde{v}'^A(\bar{o}) = v'^A(\bar{o})$. Using $\tilde{\mathbf{v}}'$ instead of $\mathbf{v}'$, the same policy $\bar{\pi}$ then induces a vector $\tilde{\mathbf{v}} \in \hat{\mathcal{V}}_h(o)$ to approximate $\mathbf{v}$. Indeed, the agent's values are exactly the same under $\tilde{\mathbf{v}}'$ and $\mathbf{v}'$, so the same response of the agent can be incentivized. This is why we require the approximation to not compromise on the agent's value. Moreover, according to (3), the overall difference between $\tilde{v}^P$ and v^P is at most ε because it holds for the coefficients that $\sum_{s, \omega, \mathbf{a}} p_{h-1}(s, \omega | o) \cdot \bar{\pi}(\mathbf{a} | \omega) = 1$. As a result, $\tilde{v}^P \geq v^P - \varepsilon$ and $\bar{\mathcal{V}}_h(o)$ is an ε -approximation of $\mathcal{V}_h(o)$.

Hence, the inductive hypothesis holds for h . By induction, $\hat{\mathcal{V}}_1(\emptyset)$ is an δH -approximation of $\mathcal{V}_1(\emptyset)$. Since $\delta H = \epsilon$, we get that $\max_{\mathbf{v} \in \hat{\mathcal{V}}_1(\emptyset)} v^P \geq \max_{\mathbf{v} \in \mathcal{V}_1(\emptyset)} v^P - \epsilon$. $\square$

Theorem 5. *There exists an ϵ -optimal IC policy π such that, for any given sequence $(\sigma; \omega^P, \tilde{\omega}^A) \in \Sigma \times \Omega$, the distribution $\pi(\cdot | \sigma; \omega^P, \tilde{\omega}^A)$ can be computed in time $\text{poly}(|S| \cdot |A| \cdot |\Omega|, H, 1/\epsilon)$.*

Proof. Consider the algorithm presented in Fig. 3. The outputs of the algorithm over all possible input sequences $(\sigma; \omega^P, \tilde{\omega}^A) \in \Sigma \times \Omega$ specify a policy π . The polynomial running time of the algorithm for computing each $\pi(\cdot | \sigma; \omega^P, \tilde{\omega}^A)$ follows by noting that it runs by solving at most H linear constraint satisfiability problems.

It remains to argue that π is IC and ϵ -optimal. Indeed, by Lemma 4 and an inductive argument, π is IC at each time step h and induces the corresponding values encoded in $\mathbf{v}'$ as the expected onward values. The ϵ -optimality of π follows given the condition $\max_{\mathbf{v} \in \hat{\mathcal{V}}_1(\emptyset)} v^P \geq \max_{\mathbf{v} \in \mathcal{V}_1(\emptyset)} v^P - \epsilon$ stated in Lemma 4 (and the choice of the initial $\mathbf{v}$ in Fig. 3). $\square$

A.2 Omitted Proofs in Section 4

Lemma 9. *There exists an ϵ -optimal δ -IC policy π such that, for any given sequence $(\sigma; \omega^P, \tilde{\omega}^A) \in \Sigma \times \Omega$, the distribution $\pi(\cdot | \sigma; \omega^P, \tilde{\omega}^A)$ can be computed in time $\text{poly}(|S| \cdot |A| \cdot |\Omega|, H, 1/\epsilon, \log(1/\delta))$.*

Proof. The proof is similar to the approach in Section 3.1, which computes a near-optimal and 0-IC policy. We describe the differences below.

Instead of maintaining two-dimensional sets of inducible values, we split the dimension of the agent's value into two dimensions v^A and v_*^A , which represent the agent's values under his truthful response (i.e., $\perp$) and his best deviation plan, respectively. Hence, each $\mathbf{v} \in \mathcal{V}(o)$ is now a tuple (v^P, v^A, v_*^A) .

(In Section 3.1, v^A and v_*^A are eventually forced to be the same, so there is no need to keep an additional dimension.)

The inducibility of a vector $\mathbf{v} = (v^P, v^A, v_*^A)$ is characterized by the following constraints. First, we impose the same constraint as (3) on the first two dimensions of $\mathbf{v}$, so that they capture the players' payoffs under the agent's truthful response. In order for the third dimension v_*^A to capture the agent's maximum attainable value, we use a constraint similar to (4):

$$v_*^A \geq \sum_{\omega^A} p_{h-1}(\omega^A | o) \max_{\tilde{\omega}^A} \sum_{a^A} \max_{\tilde{a}^A} \sum_{s, \omega^P, a^P} p_{h-1}(s, \omega^P | o, \omega^A) \cdot \bar{\pi}(\mathbf{a} | \omega^P, \tilde{\omega}^A) \cdot \left(r_h^A(s, a^P, \tilde{a}^A) + v_*'^A(s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A) \right). \quad (8)$$

The remaining constraint is the same as (6).

All the non-linear constraints can be linearized the same way as the approach described in Section 3.1. Hence, we can efficiently approximate $\mathcal{V}_h(o)$ by examining the inducibility of points on a sufficiently fine-grained grid in $[0, H]^3$, which contains $\text{poly}(H, 1/\epsilon)$ many points, and constructing the convex hull of these points. (Note that there is no need to ensure zero compromise on the agent's value as required in the proof of Lemma 4. This is because δ -IC is defined with respect to the agent's expected value at the beginning of the game instead of that at every time step. Hence, using points on a grid suffices the purpose of the approximation in this proof.) The half-space representation of the convex hull can be computed efficiently given that it is in $\mathbb{R}^3$ [9]. Eventually, an optimal $\pi \in \hat{\Pi}_\delta$ corresponds to a solution to $\max_{\mathbf{v} \in \mathcal{V}_1(o)} v^P$ subject to $v^A \geq v_*^A - \delta$, and we can use the same forward construction procedure in Section 3.2 to compute $\pi_h(\cdot | \sigma; \omega^P, \tilde{\omega}^A)$.

Note that (8) only enforces v_*^A as an upper bound of the maximum attainable value, instead of the exact value. This suffices for our purpose because any (v^P, v^A, v_*^A) in the feasible set $\mathcal{V}_1(o) \cap \{\mathbf{v} : v^A \geq v_*^A - \epsilon\}$ also implies the inclusion of $(v^P, v^A, \tilde{v}_*^A)$ in the same feasible set, where $\tilde{v}_*^A$ is the actual maximum attainable value induced by the policy that induces (v^P, v^A, v_*^A) according to our formulation. $\square$

Theorem 10. *There exists an algorithm that guarantees regret $\tilde{O}(\zeta^{1/3} T^{2/3})$ for both players with probability $1 - q$, where $\zeta = H^5 |S|^2 |A|^3 |\Omega|^2$. The computation involved in implementing the algorithm takes time $\text{poly}(|S| \cdot |A| \cdot |\Omega|, H, T)$.*

Proof. We run reward-free exploration to obtain a model $\hat{p}$ with error bound $\delta/2$. This can be achieved w.h.p. in $\tilde{O}(\zeta/\delta^2)$ episodes according to Lemma 7. Next, we compute an δ -optimal strategy $\pi \in \hat{\Pi}_\delta$ and use it in the remaining rounds. According to Lemma 9, this can be done in polynomial time.

By assumption, rewards are bounded in $[0, 1]$ so the regrets are at most 1 for both players in each of the exploration episodes. In each of the remaining episodes, the agent's regret is as follows, where we pick arbitrary $\rho^* \in \arg \max_\rho V_1^{A, \pi, \rho}(\emptyset)$:

$$\begin{aligned} V_1^{A, \pi, \rho^*}(\emptyset) - V_1^{A, \pi, \perp}(\emptyset) &\leq \underbrace{\left| \hat{V}_1^{A, \pi, \rho^*}(\emptyset) - \hat{V}_1^{A, \pi, \perp}(\emptyset) \right|}_{\leq \delta \text{ as } \pi \in \hat{\Pi}_\delta} + \\ &\quad \underbrace{\left| \hat{V}_1^{A, \pi, \rho^*}(\emptyset) - V_1^{A, \pi, \rho^*}(\emptyset) \right| + \left| \hat{V}_1^{A, \pi, \perp}(\emptyset) - V_1^{A, \pi, \perp}(\emptyset) \right|}_{\leq \delta \text{ by Lemma 7}} \leq 2\delta. \end{aligned}$$

The principal's regret is:

$$\begin{aligned} V^* - V_1^{P, \pi, \perp}(\emptyset) &= \max_{\pi' \in \Pi_0} V_1^{P, \pi', \perp}(\emptyset) - V_1^{P, \pi, \perp}(\emptyset) \\ &\leq \underbrace{\max_{\pi' \in \hat{\Pi}_\delta} V_1^{P, \pi', \perp}(\emptyset) - V_1^{P, \pi, \perp}(\emptyset)}_{\text{as } \Pi_0 \subseteq \hat{\Pi}_\delta} \leq \underbrace{\max_{\pi' \in \hat{\Pi}_\delta} \hat{V}_1^{P, \pi', \perp}(\emptyset) - \hat{V}_1^{P, \pi, \perp}(\emptyset)}_{\leq \delta \text{ as } \pi \text{ is } \delta\text{-optimal}} + \delta \leq 2\delta. \end{aligned}$$

The reason that $\Pi_0 \subseteq \hat{\Pi}_\delta$ is the following: Since $\hat{p}$ ensures error bound $\delta/2$, we have $|\hat{V}_1^{A,\pi',\rho}(\emptyset) - V_1^{A,\pi',\rho}(\emptyset)| \leq \delta/2$ for all ρ . By definition, $\pi' \in \Pi_0$ means that $V_1^{A,\pi',\perp}(\emptyset) \geq V_1^{A,\pi',\rho}(\emptyset)$. So, $\hat{V}_1^{A,\pi',\perp}(\emptyset) \geq \hat{V}_1^{A,\pi',\rho}(\emptyset) - \delta$ for all ρ ; hence, $\pi' \in \hat{\Pi}_\delta$.

The above bounds then lead to a total regret of at most $\tilde{\mathcal{O}}(\zeta/\delta^2) + \mathcal{O}(T\delta)$ for each player. Taking $\delta = (\zeta/T)^{1/3}$ gives the upper bound $\tilde{\mathcal{O}}(\zeta^{1/3}T^{2/3})$. $\square$

B Linearizing (3) and (4)

Specifically, to remove the maximization operator in (4), we introduce a set of auxiliary variables $y(a^A, \omega^A, \tilde{\omega}^A)$ to capture the maximum values on the right hand side of (4). We replace the right hand side of (4) with $\sum_{a^A \in A^A} y(a^A, \omega^A, \tilde{\omega}^A)$, and by adding the following constraint we force each $y(a^A, \omega^A, \tilde{\omega}^A)$ to be an upper bound of the corresponding maximum value: for all $\tilde{a}^A \in A^A$,

$$y(a^A, \omega^A, \tilde{\omega}^A) \geq \sum_{s, \omega^P, a^P} p_{h-1}(s, \omega^P | o, \omega^A) \cdot \bar{\pi}(\mathbf{a} | \omega^P, \tilde{\omega}^A) \cdot \left(r_h^A(s, a^P, \tilde{a}^A) + v'^A(s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A) \right) \quad (9)$$

To remove the quadratic terms in (3) and (9), we use an auxiliary variable $\mathbf{z}(s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A)$ to replace each term $\bar{\pi}(\mathbf{a} | \omega) \cdot \mathbf{v}'(s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A)$ and impose the following constraint on $\mathbf{z}$:

$$\mathbf{H} \cdot \mathbf{z}(s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A) \leq \bar{\pi}(\mathbf{a} | \omega) \cdot \mathbf{b}, \quad (10)$$

where $\mathbf{H}$ and $\mathbf{b}$ are taken from the half-space representation of the polytope $\hat{\mathcal{V}}_{h+1}$, i.e., $\hat{\mathcal{V}}_{h+1}(s, a^P, \tilde{a}^A) = \{\mathbf{x} : \mathbf{H} \cdot \mathbf{x} \leq \mathbf{b}\}$. It is straightforward that, when $\hat{\mathcal{V}}_{h+1}(s, a^P, \tilde{a}^A)$ is nonempty and bounded, (10) holds if and only if $\mathbf{z}(s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A) = \bar{\pi}(\mathbf{a} | \omega) \cdot \mathbf{x}$ for some $\mathbf{x} \in \hat{\mathcal{V}}_{h+1}(s, a^P, \tilde{a}^A)$.⁵ Hence, (10) is the only constraint needed (for each tuple $(s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A)$) after we replace the terms with $\mathbf{z}$.

The Complete Formulation In summary, the above approach yields the following linearized version of the constraint satisfiability problem in Section 3.1, where $\bar{\pi}$, $\mathbf{z} = (z^A, z^P)$, and y are the variables (highlighted in blue).

1. The value function constraint:

$$\mathbf{v} = \sum_{s, \omega, \mathbf{a}} p_{h-1}(s, \omega | o) \cdot \left(\mathbf{r}_h(s, \mathbf{a}) \cdot \bar{\pi}(\mathbf{a} | \omega) + \mathbf{z}(s, \omega, \omega^A, \mathbf{a}, a^A) \right).$$

2. An IC constraint for each $\omega^A \in \Omega^A$:

$$\sum_{s, \omega^P, \mathbf{a}} p_{h-1}(s, \omega^P | o, \omega^A) \cdot \left(r_h^A(s, \mathbf{a}) \cdot \bar{\pi}(\mathbf{a} | \omega) + z^A(s, \omega, \omega^A, \mathbf{a}, a^A) \right) \geq \sum_{a^A \in A^A} y(a^A, \omega^A, \tilde{\omega}^A).$$

Moreover, for each tuple $(a^A, \omega^A, \tilde{\omega}^A) \in A^A \times \Omega^A \times \Omega^A$:

$$y(a^A, \omega^A, \tilde{\omega}^A) \geq \sum_{s, \omega^P, a^P} p_{h-1}(s, \omega^P | o, \omega^A) \left(r_h^A(s, a^P, \tilde{a}^A) \cdot \bar{\pi}(\mathbf{a} | \omega^P, \tilde{\omega}^A) + z^A(s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A) \right).$$

3. An onward value constraint for each tuple $(s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A) \in \bar{\Sigma}$:

$$\mathbf{H}(s, a^P, \tilde{a}^A) \cdot \mathbf{z}(s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A) \leq \bar{\pi}(\mathbf{a} | \omega) \cdot \mathbf{b}(s, a^P, \tilde{a}^A),$$

where for every $o \in O$, the matrix $\mathbf{H}(o)$ and vector $\mathbf{b}(o)$ are given by the half-space representation of $\hat{\mathcal{V}}_{h+1}(o)$, i.e., $\hat{\mathcal{V}}_{h+1}(o) = \{\mathbf{v}' \in \mathbb{R} : \mathbf{H}(o) \cdot \mathbf{v}' \leq \mathbf{b}(o)\}$.

4. Additionally, we impose

$$\bar{\pi}(\cdot | \omega) \in \Delta(A)$$

for each $\omega \in \Omega$ to ensure that $\bar{\pi}(\cdot | \omega)$ is a valid distribution over A .

⁵Note that if $\bar{\pi}(\mathbf{a} | \omega) = 0$, then (10) imply that $\mathbf{z}(s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A) = \mathbf{0}$: otherwise, the fact that $\mathbf{x}' = c \cdot \mathbf{z}(s, \omega, \tilde{\omega}^A, \mathbf{a}, \tilde{a}^A) + \mathbf{x}$ satisfies $\mathbf{H} \cdot \mathbf{x}' \leq \mathbf{b}$ for any $c \geq 0$ and $\mathbf{x} \in \hat{\mathcal{V}}_{h+1}(s, a^P, \tilde{a}^A)$ would prevent $\hat{\mathcal{V}}_{h+1}(s, a^P, \tilde{a}^A)$ from being bounded.

C Additional Discussion about Intractability without Hindsight Observability

The PSPACE-hardness can be seen by thinking of a POMDP as an instance of our problem where only the principal can make observations and perform actions to influence the environment (essentially, the agent can neither influence the principal nor the environment in this instance).

The PSPACE-hardness remains in the case of information design, where the principal observes the state directly but does not act, while the agent makes no observation but acts; as well as the case of mechanism design, where the agent observes the state directly but does not act, while the principal does not observe but acts. This can be seen by considering zero-sum instances, where the principal's and the agent's rewards sum to zero.

More specifically, consider for example the case of information design. If the goal is to compute the principal's maximum attainable payoff, the PSPACE-hardness of the problem is immediate: Since the game is zero-sum, it is optimal for the principal to not send no signal (if signaling were to improve the principal's payoff, the agent would be better-off just ignoring the signals). Hence, computing the maximum attainable payoff of the principal in this case is equivalent to computing (the negative of) the agent's maximum attainable payoff, which amounts to solving a POMDP.

One may argue that while the above example demonstrates the hardness of determining the principal's maximum attainable payoff, computing the principal's optimal policy is actually trivial in the example (i.e., sending no signal is optimal). So it does not rule out the possibility of an efficient algorithm which, given any sequence, computes the signal distribution of an optimal policy, without computing the principal's payoff the policy yields. It turns out that this is not possible, either.

Consider a game where the agent can choose between two actions a and b in the first time step. Action a leads to a process where the principal's rewards are zero for all state-action pairs. Action b leads to another process with payoffs $1 - x$ for the principal and x for the agent, where $x \in [0, 1]$ depends on the principal's signaling strategy in this sub-process. For example, we can design this sub-process as a matching pennies game, where: nature flips a fair coin, the principal observes the outcome, and the agent must choose the same side of the coin to get a reward 1 and otherwise he gets -1 . If the agent plays this matching pennies game on his own, his expected payoff is 0. The principal can reveal her observation to help the agent to improve the payoff. And the principal can do so probabilistically, so that she can fine tune the agent's expected payoff x to any desired value in $[0, 1]$. To maximize the principal's payoff in the entire process requires finding an x that is sufficiently high, so that the agent is incentivized to choose b (otherwise, the principal only gets 0); at the same time, we would like x to be as low as possible to maximize the principal's payoff $1 - x$. This essentially requires knowing the agent's maximum attainable payoff in the sub-process following a , which is PSPACE-hard as we discussed above.