Automated Dynamic Mechanism Design

Hanrui Zhang
Duke University
hrzhang@cs.duke.edu

Vincent Conitzer
Duke University
conitzer@cs.duke.edu

Abstract

We study Bayesian automated mechanism design in unstructured dynamic environments, where a principal repeatedly interacts with an agent, and takes actions based on the strategic agent's report of the current state of the world. Both the principal and the agent can have arbitrary and potentially different valuations for the actions taken, possibly also depending on the actual state of the world. Moreover, at any time, the state of the world may evolve arbitrarily depending on the action taken by the principal. The goal is to compute an optimal mechanism which maximizes the principal's utility in the face of the self-interested strategic agent.

We give an efficient algorithm for computing optimal mechanisms, with or without payments, under different individual-rationality constraints, when the time horizon is constant. Our algorithm is based on a sophisticated linear program formulation, which can be customized in various ways to accommodate richer constraints. For environments with large time horizons, we show that the principal's optimal utility is hard to approximate within a certain constant factor, complementing our algorithmic result. These results paint a relatively complete picture for automated dynamic mechanism design in unstructured environments. We further consider a special case of the problem where the agent is myopic, and give a refined efficient algorithm whose time complexity scales linearly in the time horizon.

In the full version of the paper, we show that memoryless mechanisms, which are without loss of generality optimal in Markov decision processes without strategic behavior, do not provide a good solution for our problem, in terms of both optimality and computational tractability. Moreover, we present experimental results where our algorithms are applied to synthetic dynamic environments with different characteristics, which not only serve as a proof of concept for our algorithms, but also exhibit intriguing phenomena in dynamic mechanism design.

1 Introduction

Consider the following scenario. A company assembles an internal research group to develop key technologies to be used in the company's next-generation product in 5 years. The more progress the group makes, the more successful the product is likely to be. Since research progress is hard to monitor, the company manages the group based on its annual reports. At the beginning of each year, the group submits a report, summarizing its progress in the preceding year, as well as its needs for the current year. Taking into consideration this report (and possibly also reports from previous years), the company then decides the compensation level and the headcount of the group in the current year. Moreover, after the product launches, the company may also pay a bonus to members of the group, depending on how successful the product is.

For simplicity, suppose an annual report consists of two items: research progress (satisfactory/unsatisfactory), and need to expand (no request/request for an intern/request for a full-time employee). The company's goal is to encourage and facilitate research progress while keeping the expenses reasonable. So, a natural managing strategy is to increase (resp. decrease) the compensation

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

level when the reported research progress is satisfactory (resp. unsatisfactory), and to allow the group to expand only when necessary, i.e., when the reported research progress is unsatisfactory. However, the research group may have a different goal than the company's. Suppose members of the group do not care about the success of the product *per se*. Instead, their primary goal is to maximize the total compensation received from the company, and for this reason, they may be incentivized to *misreport* the situation. In other words, the company faces a *dynamic mechanism design* problem, where the *principal* (i.e., the company) needs to implement (and commit to) a *mechanism* (i.e., a managing strategy) that achieves its goal through *repeated* interactions, in the presence of strategic behavior of the *agent* (i.e., the research group).

Indeed this problem is nontrivial. For example, if the company implements the above strategy, then the group will report satisfactory progress regardless of the actual situation, which maximizes the group's total compensation over the 5 years, but also causes greater expenses for the company and jeopardizes the success of the product. To counter this, the company may additionally promise a significant bonus contingent on the success of the product. This creates incentives for the group to make more progress, and discourages overreporting the progress, because the group is not allowed to expand when the reported progress is satisfactory. That is, if actual progress is unsatisfactory, this introduces an incentive to report this truthfully so that the group may expand. However, this also runs the risk of encouraging the group to report unsatisfactory progress in order to expand even if actual progress is satisfactory, because more members always make more progress, which leads to a higher (chance of) bonus, whereas the cost of expanding is paid by the company and therefore irrelevant to the group.

One may try to fix this by introducing more rules, possibly replacing existing ones. For example, the company may allow the group to recruit an intern, but not a full-time employee, when the reported progress is unsatisfactory. Then, in the next year, if the reported progress improves, the company allows the group to make a return offer to the intern as a full-time employee. Or alternatively, the company may unconditionally allow the group to recruit interns (which are less costly), but never full-time employees. In addition to the above, the company could also temporarily decrease the compensation level when a new member joins, and later adjust the compensation based on how the reported progress improves. While all these ad hoc rules make intuitive sense, it is not immediately clear which (combinations of) rules are better, how to optimize parameters of these rules (e.g., the number of new members allowed per year and the amount by which the compensation is adjusted), or whether there is a better set of rules that look totally different.

As demonstrated by the foregoing discussion, in general, the problem of finding an optimal mechanism in *unstructured* dynamic environments, such as the above example, turns out to be extremely rich and challenging. In such environments, the actions of the principal may go beyond the allocation of items to the agent, and affect the state of the world in arbitrary ways. Moreover, both the principal and the agent may have arbitrary valuations for these actions, which also depend on the current state of the world. In economic theory, the *characterize-and-solve* approach [41, 20, 47] to mechanism design has achieved spectacular success in both static and dynamic environments, by exploiting structure of the environment to construct a characterization of optimal mechanisms, often leading to closed-form or computationally tractable solutions. However, since the environments under consideration here are loosely structured at best, the classical characterize-and-solve approach does not seem particularly suited. When disregarding the agent's incentives, one could treat the problem of finding an optimal strategy as a *planning* problem, which is known to be solvable efficiently [7, 33, 48]. However, as discussed above, the agent's strategic behavior can ruin the performance of such a strategy. From a computational perspective, while numerous methods for *automated mechanism design*, which efficiently compute optimal mechanisms without heavily exploiting structures of the environment, have been proposed [17, 18, 19], all existing methods work only for static environments with one-time interactions, and it is not immediately clear how to generalize these methods to dynamic environments. All this brings us to the following question:

Can we efficiently compute optimal mechanisms in unstructured dynamic environments?

1.1 Our Results

In this paper, we study the problem of computing optimal mechanisms in *single-agent*, *discrete-time* dynamic environments with a *finite time horizon*, without any further structural assumptions. Our main results (presented in Section 3) can be summarized as follows:

- **Efficient algorithm**: when the time horizon is fixed, there is a polynomial-time algorithm for computing optimal mechanisms, with or without payments, that maximize the principal's utility facing a strategic agent.
- **Inapproximability**: when the time horizon can be large, it is NP-hard to approximate the principal's optimal utility within a factor of $(7/8 + \varepsilon)$ for any $\varepsilon > 0$.

To the best of our knowledge, our algorithm for constant time horizons is the first that efficiently computes optimal mechanisms in unstructured dynamic environments. The fact that our algorithm cannot scale beyond constant time horizons is by no means surprising: optimal dynamic mechanisms generally depend on the entire history, and as a result, the straightforward description of such a mechanism is exponentially large in the time horizon. Our inapproximability result further rules out the possibility of computing succinct representations of approximately optimal mechanisms that can be efficiently evaluated. These results together paint a complete picture of the computational complexity of dynamic mechanism design in unstructured environments.

1.2 Further Related Work

Dynamic mechanism design. The problem we study can be situated in the broad area of dynamic mechanism design, and below we discuss some representative related work. For a more comprehensive exposition, see, e.g., the survey by Pavan [46] and the one by Bergemann and Välimäki [9]. In the context of efficient (i.e., welfare-maximizing) mechanisms, Bergemann and Välimäki [8] propose the dynamic pivot mechanism, which generalizes the VCG mechanism in static environments, and Athey and Segal [2] propose the team mechanism, which focuses on budget-balancedness. As for optimal (i.e., revenue-maximizing) mechanisms, which are more closely related to our results, following earlier work [6, 20, 25], Pavan et al. [47] generalize the classical characterization by Myerson [41] into dynamic environments, unifying previous results with continuous type spaces. Ashlagi et al. [1] study ex-post individual-rational dynamic mechanisms for repeated auctions, and give an efficient $(1 - \varepsilon)$-approximation to the optimal revenue for a single agent with independent valuations across items. Mirrokni et al. [39] study non-clairvoyant dynamic mechanism design, where future distributional knowledge is unavailable to the principal. All these results for optimal mechanisms follow the characterize-and-solve approach, which is quite different from the computational approach that we take.

Particularly related to our results is the work by Papadimitriou et al. [44], who study a setting where one item is sold at each time, and agents' valuations can be correlated across items. They show that designing an optimal deterministic mechanism is computationally hard even when there is only one agent and two items (thereby ruling out the possibility of efficiently computing optimal deterministic mechanisms in our model, which is more general). And moreover, they give a polynomial-size linear program formulation for optimal randomized mechanisms for independent agents when the number of agents and the time horizon are both constant. Restricted to a single agent, their LP formulation can be viewed as a special case of our main result: they focus on revenue maximization with a single item to be allocated at each time, in a model where the principal's actions cannot affect the future valuations of the agent; on the other hand, we allow the principal to care about actions as well as revenue, with actions being general and unstructured (as opposed to allocation/no allocation), where the future state of the world can depend arbitrarily on the principal's actions as well as the current state.

Automated Mechanism Design. There is a rich body of research regarding automated mechanism design (AMD) in (essentially) static environments. Conitzer and Sandholm [17, 18] initiated the study of automated mechanism design. They consider various specific static setups, and show that computing optimal deterministic mechanisms, even with a single agent, is often NP-hard (which also rules out the possibility of efficiently computing optimal deterministic mechanisms in our model, since the 1-period case is a special case), while computing optimal randomized mechanisms is often tractable. Conceptually related to our model, Hajiaghayi et al. [31] consider a model where agents

enter and leave the mechanism online (but still have one-time interactions with the mechanism), and provide an algorithm for computing mechanisms that are competitive against the optimal allocation from hindsight. Sandholm et al. [52] study automated design of multistage mechanisms, but these are not for dynamic settings; instead, the motivation is to implement static mechanisms using multiple rounds of queries in order to minimize the communication cost. Sandholm and Likhodedov [51] study automated design of combinatorial auction mechanisms, and Balcan et al. [3, 4] study the sample complexity thereof. Kephart and Conitzer [34, 35] and Zhang et al. [56] study AMD with partial verification and/or reporting costs. More recently, various methods have been proposed for automated mechanism design via machine learning [22, 43], and in particular, deep learning [23, 26, 53, 49]. All these results are essentially for static environments, whereas in this paper, we focus solely on AMD in dynamic environments. Another emerging research direction is Bayesian persuasion in dynamic environments [24, 50]. In particular, Celli et al. [12] study an algorithmic persuasion problem in extensive-form games, where a single signal is sent at the very beginning, and Gan et al. [28] study an algorithmic persuasion problem in infinite-horizon discounted MDPs, where a new signal is sent at every time. These persuasion problems can be viewed as a dual problem of ours: in our problem, the principal has the commitment power, and tries to encourage the agent to truthfully report their private information, whereas in (dynamic) Bayesian persuasion, the agent has the commitment power, and tries to induce the principal to act in favor of the agent by selectively revealing their private information.

2 Preliminaries

Dynamic environments. Throughout this paper, we consider single-agent, discrete-time environments with a finite time horizon. Below, we give a general definition of such a dynamic environment. Let T be the time horizon, $\mathcal{S}$ be the state space, and $\mathcal{A}$ be the action space. The agent observes the state, but the principal controls the action that is taken. For each $t \in [T]$, let $v_t^P : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ be the principal's valuation function, where for each state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$, $v_t^P(s, a)$ is the value of the principal when playing action a in state s, at time t; similarly, let $v_t^A : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ be the agent's value function. Let $P_0 \in \Delta(\mathcal{S})$ be the initial distribution over states, and for each $s \in \mathcal{S}$, denote by $P_0(s)$ the probability that the initial state is s. Moreover, for each $t \in [T]$, let $P_t : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ be the transition operator, which maps a state-action pair (s, a) at time t to the distribution of the next state at time $t + 1$, $P_t(s, a) \in \Delta(\mathcal{S})$. We denote by $P_t(s, a, s')$ the probability that the next state is s' when playing action a in state s at time $t \in [T]$. For notational simplicity, let $P_0(s, a, s') = P_0(s')$ for all $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$. (Note that the first *actual* action is taken at $t = 1$ — not $t = 0$ — possibly based on the state at $t = 1$.)

Histories. A t-step history is a sequence of states and actions $(s_1, a_1, s_2, \ldots, a_{t-1}, s_t, a_t)$, where for each $i \in [t]$, it is the case that $s_i \in \mathcal{S}$ and $a_i \in \mathcal{A}$. For each $t \in [T]$, let $\mathcal{H}_t$ be the set of all possible t-step histories, i.e.,

$$\mathcal{H}_t = \{(s_1, a_1, \ldots, s_t, a_t) \mid s_i \in \mathcal{S}, a_i \in \mathcal{A} \text{ for all } i \in [t]\}.$$

For each $h = (s_1, a_1, \ldots, s_t, a_t) \in \mathcal{H}$, let $|h| = t$, and moreover, for any $s_{t+1} \in \mathcal{S}$, $a_{t+1} \in \mathcal{A}$, let $h + (s_{t+1}, a_{t+1}) = (s_1, a_1, \ldots, s_{t+1}, a_{t+1})$. Let $\mathcal{H}_0 = \{\emptyset\}$, where $\emptyset$ corresponds to the empty history with $|\emptyset| = 0$. Let $\mathcal{H} = \mathcal{H}_0 \cup \bigcup_{t \in [T-1]} \mathcal{H}_t$ be the set of all possible histories of length at most $T - 1$ in the dynamic environment. Note that $\mathcal{H}$ does not contain histories of length T.

Dynamic mechanisms. Dynamic mechanisms are more powerful than static ones, in that they may depend on the *entire history*, rather than only the current state. A (randomized) dynamic mechanism $M = (\pi, p)$ consists of an action policy π and a payment function p. The action policy $\pi : \mathcal{H} \times \mathcal{S} \to \Delta(\mathcal{A})$ maps each history $h \in \mathcal{H}$, extended with the reported current state $s \in \mathcal{S}$, to a distribution over actions $\pi(h, s) \in \Delta(\mathcal{A})$. We denote by $\pi(h, s, a)$ the probability that the action taken by the mechanism is a for (h, s). The payment function $p : \mathcal{H} \times \mathcal{S} \to \mathbb{R}$ maps the extended history (h, s) to a real number, i.e., the payment, made from the agent to the principal (but it can be negative). We remark that in principle, one can absorb payments into the action space. However, doing so would make the action space uncountable, introducing subtleties into the computational problem (which is the main focus of this paper). Here, we keep payments separate and explicit to avoid such issues. Also, our algorithm allows linear constraints on feasible payments, including but not limited to: nonnegative payments, no payments, etc. See Section 3.2 for more details.

Utilities. Fixing a mechanism $M = (\pi, p)$, we can then define the onward utility of the principal and the agent. Let $u_P^M : \mathcal{H} \times \mathcal{S} \to \mathbb{R}$ be the principal's onward utility function under mechanism M, defined inductively such that

$$u_P^M(h, s) = \sum_a \pi(h, s, a) \cdot \left(v^P_{|h|+1}(s, a) + \sum_{s'} P_{|h|+1}(s, a, s') \cdot u_P^M(h + (s, a), s') \right) + p(h, s),$$

with the boundary condition that $u_P^M(h, s) = 0$ for all $h \in \mathcal{H}_T$ and $s \in \mathcal{S}$. Here, all summations are over the entire state/action space. Let $u_P^M(\emptyset)$ be the overall utility of the principal, i.e.,

$$u_P^M(\emptyset) = \sum_s P_0(s) \cdot u_P^M(\emptyset, s).$$

Similarly, let $u_A^\pi : \mathcal{H} \times \mathcal{S} \to \mathbb{R}$ be the agent's onward utility function under mechanism M, defined such that

$$u_A^M(h, s) = \sum_a \pi(h, s, a) \cdot \left(v^A_{|h|+1}(s, a) + \sum_{s'} P_{|h|+1}(s, a, s') \cdot u_A^M(h + (s, a), s') \right) - p(h, s),$$

where $u_A^M(h, s) = 0$ for all $h \in \mathcal{H}_T$ and $s \in \mathcal{S}$. And let $u_A^M(\emptyset)$ be the overall utility of the agent, i.e.,

$$u_A^M(\emptyset) = \sum_s P_0(s) \cdot u_A^M(\emptyset, s).$$

We remark that while the above definition assumes that the principal cares about payments as much as the agent does, in fact, our algorithm allows for the principal to care about payments in an arbitrary linear way (including possibly not at all). See Section 3.2 for a detailed discussion.

Incentive-compatible mechanisms. We say a mechanism M is incentive-compatible (IC) if the agent can never achieve a higher overall utility by misreporting the state, even in sophisticated ways. Formally, a reporting strategy $r : \mathcal{H} \times \mathcal{S} \to \mathcal{S}$ maps each history h extended with the current state s to a reported state s', which is possibly different from s. Note that the agent only (mis)reports the current state, since the principal can memorize all historical reports. This reporting strategy induces a reported history $r(h) = (s'_1, a_1, \ldots, s'_t, a_t)$ for each actual history $h = (s_1, a_1, \ldots, s_t, a_t)$, where for each $i \in [t]$,

$$s'_i = r((s_1, a_1, \ldots, s_{i-1}, a_{i-1}), s_i).$$

Note that we abuse notation here: in particular, $r(h, s)$ denotes a reported state, whereas $r(h)$ denotes a reported history. And without loss of generality, we only consider deterministic reporting strategies. Given a mechanism M and a reporting strategy r, we can define the agent's utility function $u_A^{M,r}$ under M and r inductively such that

$$\begin{aligned} u_A^{M,r}(h, s) = \sum_a \pi(r(h), r(h, s), a) \cdot \left(v^A_{|h|+1}(s, a) + \sum_{s'} P_{|h|+1}(s, a, s') \cdot u_A^{M,r}(h + (s, a), s') \right) \\ - p(r(h), r(h, s)), \end{aligned}$$

where $u_A^{M,r}(h, s) = 0$ for all $h \in \mathcal{H}_T$ and $s \in \mathcal{S}$. And let $u_A^{M,r}(\emptyset)$ be the overall utility of the agent, i.e.,

$$u_A^{M,r}(\emptyset) = \sum_s P_0(s) \cdot u_A^{M,r}(\emptyset, s).$$

In words, $u_A^{M,r}$ is the utility function of the agent applying the reporting strategy r in response to the mechanism M. The mechanism M is IC iff for any such reporting strategy r,

$$u_A^M(\emptyset) \geq u_A^{M,r}(\emptyset).$$

Since the revelation principle holds in dynamic environments (see, e.g., [42]), we focus on IC mechanisms in the rest of the paper.[1]

[1] Of course, the revelation principle will not hold in our dynamic setting if we allow it to generalize a static setting in which the revelation principle does not hold. For example, in the case of partial verification — not every type being able to misreport every other type — or costly misreporting, the revelation principle is known to hold only under certain conditions [35]. In this paper, we only consider the standard mechanism design setting in which every type can freely misreport any other type, but our techniques can be generalized to the other settings as well.

Individually-rational mechanisms. When payments are allowed, it is standard to impose individual-rationality (IR) (also known as voluntary-participation) constraints on the mechanism, which roughly say that the agent should never be made worse off by participating in the mechanism. In this paper, we consider two versions of IR constraints:

- A mechanism M is overall IR if the overall utility of the agent is nonnegative, i.e., $u_A^M(\emptyset) \geq 0$. This ensures that the agent is willing to participate in the overall mechanism.
- A mechanism M is dynamic IR if the onward utility of the agent is nonnegative for every history h and current state s, i.e., $u_A^M(h, s) \geq 0$ for all $h \in \mathcal{H}$ and $s \in \mathcal{S}$. This stronger notion of IR further ensures that the agent has no incentive to leave the mechanism at any time.

As discussed in later sections, our algorithms work for all 3 cases regarding IR constraints: no IR (which results in an unbounded objective value if payments are allowed and valued by the principal), overall IR, and dynamic IR.

3 Computation of Optimal Mechanism

In this section, we investigate the computational problem of finding an optimal dynamic mechanism, which maximizes the principal's overall utility. For concreteness, we assume that all components of the dynamic environment, including the time horizon T, state and action spaces $\mathcal{S}$ and $\mathcal{A}$, valuation functions v^P and v^A, and transition operator P, are given explicitly as input.

3.1 Hardness Result for Long-Horizon Environments

First we show that the problem with an arbitrarily large time horizon T is intractable. In general, it takes exponentially many parameters in T to describe a dynamic mechanism, which immediately rules out the possibility of computing a flat representation of an optimal mechanism in polynomial time. However, this leaves the possibility of computing succinct representations, e.g., an oracle which maps extended histories to distributions over actions. Our hardness result shows that it is hard to approximate the principal's maximum utility within a constant factor, which rules out the possibility of such succinct representations that can be efficiently evaluated, assuming $\mathsf{P} \neq \mathsf{NP}$. The proof of the theorem, as well as all other proofs, are deferred to the appendices.

Theorem 1. *When the time horizon T can be arbitrarily large, it is* NP*-hard to approximate the principal's maximum utility within a factor of $7/8 + \varepsilon$ for any $\varepsilon > 0$.*

3.2 Algorithm for Short-Horizon Environments

Now we give a polynomial-time algorithm for computing an optimal mechanism when T is a constant. Our algorithm is based on a delicate linear program (LP) formulation, which relies on the following notation and concepts.

Feasible history-state pairs. A history-state pair (h, s), where $h = (s_1, a_1, \ldots, s_t, a_t)$, is i-feasible if $P_j(s_j, a_j, s_{j+1}) > 0$ for every $j \in \{i, i+1, \ldots, t-1\}$, and $P_t(s_t, a_t, s) > 0$. In other words, starting from s_i and taking the actions specified in h, there is a positive probability that the rest of the history and the state s are generated from the transition operator. We say a pair (h, s) is feasible if it is 1-feasible.

Feasible extensions. For two history-state pairs (h, s) and (h', s') where $h = (s_1, a_1, \ldots, s_t, a_t)$ and $h' = (s'_1, a'_1, \ldots, s'_{t'}, a'_{t'})$, we say that (h', s') feasibly extends (h, s), i.e., $(h, s) \subseteq (h', s')$, if $(h, s) = (h', s')$, or the following conditions hold simultaneously:

- $t = |h| < |h'| = t'$.
- For any $i \in [t]$, $(s_i, a_i) = (s'_i, a'_i)$ (this holds automatically when $h = \emptyset$ and therefore $|h| = 0$).
- $s = s'_{t+1}$.
- (h', s') is $(|h| + 1)$-feasible (note that this does not require h itself to be feasible).

objective: $$\max \sum_{h\in\mathcal{H},s\in\mathcal{S}:(h,s) \text{ is feasible}} \left(\sum_{a\in\mathcal{A}} v^P_{|h|+1}(s,a)\cdot x(h,s,a) + y(h,s)\right) \tag{1}$$

flow constraints:
$$z(h,s) = \sum_{a\in\mathcal{A}} x(h,s,a) \qquad \forall h\in\mathcal{H}, s\in\mathcal{S} \tag{2}$$
$$z(\emptyset,s) = P^E_0(s) \qquad \forall s\in\mathcal{S} \tag{3}$$
$$z(h+(s,a),s') = P^E_{|h|+1}(s,a,s')\cdot x(h,s,a) \qquad \forall h\in\mathcal{H}, s,s'\in\mathcal{S}, a\in\mathcal{A} \tag{4}$$

utility:
$$u(h,s) = \sum_{h'\in\mathcal{H},s'\in\mathcal{S}:(h,s)\subseteq(h',s')} \left(\sum_{a\in\mathcal{A}} v^A_{|h'|+1}(s',a)\cdot x(h',s',a) - y(h',s')\right) \quad \forall h\in\mathcal{H}, s\in\mathcal{S} \tag{5}$$

IC constraints:
$$u(h,s,s') = \sum_{a\in\mathcal{A}} v^A_{|h|+1}(s,a)\cdot x(h,s',a) - y(h,s') + \sum_{a\in\mathcal{A},s''\in\mathcal{S}} \frac{P_{|h|+1}(s,a,s'')}{P^E_{|h|+1}(s',a,s'')}\cdot u(h+(s',a),s'') \qquad \forall h\in\mathcal{H}, s,s'\in\mathcal{S} \tag{6}$$
$$u(h,s) \geq \frac{P^E_{|h|}(s_p,a_p,s)}{P^E_{|h|}(s_p,a_p,s')}\cdot u(h,s,s'), \text{where } (s_p,a_p)=\text{last}(h) \quad \forall h\in\mathcal{H}, s,s'\in\mathcal{S} \tag{7}$$

IR constraints:
$$u(h,s)\geq 0 \qquad \forall h\in\mathcal{H}, s\in\mathcal{S} \tag{8}$$

feasible actions:
$$x(h,s,a)\geq 0 \qquad \forall h\in\mathcal{H}, s\in\mathcal{S}, a\in\mathcal{A} \tag{9}$$

feasible payments:
$$y(h,s)\geq 0 \qquad \forall h\in\mathcal{H}, s\in\mathcal{S} \tag{10}$$

Figure 1: Linear program for computing an optimal dynamic mechanism.

Extended transition operator. For notational simplicity we define the following extended transition operator $P^E_t : \mathcal{S}\times\mathcal{A}\to\Delta(\mathcal{S})$ for all $t\in\{0\}\cup[T]$, such that

$$P^E_t(s,a,s') = \begin{cases} P_t(s,a,s'), & \text{if } P_t(s,a,s')>0 \\ 1, & \text{otherwise} \end{cases}.$$

In words, the extended transition operator assigns phantom probability 1 to each way of transitioning that happens with probability 0 (so $P^E_t(s,a)$ does not always normalize to 1). As a shorthand, let $P^E_0(s') = P^E_0(s,a,s')$ for some $s\in\mathcal{S}$ and $a\in\mathcal{A}$ (the specific choice does not matter). The extended transition operator helps in constructing the flow and IC constraints below and simplifies the formulation. In particular, we always have $P^E_t(s,a,s')>0$.

Last state-action pair. For a history $h\in\mathcal{H}$ where $h=(s_1,a_1,\ldots,s_t,a_t)$, we use $\text{last}(h)$ as a shorthand for the last state-action pair, i.e., $\text{last}(h)=(s_t,a_t)$. In particular, when $h=\emptyset$, $\text{last}(h)$ can be any state-action pair (the choice does not affect our results — it is merely a simplifying shorthand).

Now we are ready to describe the LP formulation. The complete formulation is given in Figure 1. The formulation is for nonnegative payments and dynamic IR constraints — we will discuss later

how the formulation can be modified to allow other types of constraints. Below, we describe each of its components.

Variables, flow constraints, and the corresponding mechanism. There are 5 classes of variables in the LP:

- $x(h, s, a)$: the absolute, unconditional probability that the mechanism reaches state s via history h, and takes action a.
- $y(h, s)$: the payment for history-state pair (h, s), scaled by the probability that the mechanism reaches s via h (i.e., $z(h, s)$).
- $z(h, s)$: the probability that the mechanism reaches state s via history h, which by definition satisfies
$$z(h, s) = \sum_{a \in \mathcal{A}} x(h, s, a).$$
- $u(h, s)$: the onward utility of the agent at state s with history h assuming truthful reporting, scaled by the probability that the mechanism reaches s via h (i.e., $z(h, s)$).
- $u(h, s, s')$: the onward utility of the agent at state s with history h if the agent misreports s', assuming truthful reporting in the future, scaled by the probability that the mechanism reaches s' via h (i.e., $z(h, s')$).

The flow constraints (Eq. (2)-(4)) enforce roughly the above interpretation of variables to $x(h, s, a)$ and $z(h, s)$, except for ways of transition that have probability 0. For each way of transition with probability 0, the extended transition operator assigns phantom probability 1. This phantom probability is not counted in the objective function (because only feasible history-state pairs are counted) or in the utility variables $u(h, s)$ (because only feasible extensions are counted). So, the phantom probability does not affect the principal's or the agent's utility assuming truthful reporting. Instead, together with other constraints, it guarantees that the mechanism behaves well even for history-state pairs that appear with probability 0 under truthful reporting, which is necessary for the mechanism to be IC (see later paragraphs). Under the above interpretation, the LP variables (and in particular, $x(h, s, a)$, $y(h, s)$ and $z(h, s)$) naturally correspond to a mechanism $M = (\pi, p)$. Formally, for each $h \in \mathcal{H}$, $s \in \mathcal{S}$:

- If $z(h, s) > 0$, then
$$p(h, s) = y(h, s)/z(h, s),$$
and for each $a \in \mathcal{A}$,
$$\pi(h, s, a) = x(h, s, a)/z(h, s).$$
- If $z(h, s) = 0$, then let $\pi(h, s)$ be an arbitrary distribution over $\mathcal{A}$, and $p(h, s) = 0$.

The feasibility of the mechanism (i.e., every $\pi(h, s)$ is a distribution over $\mathcal{A}$ and every $p(h, s)$ is nonnegative) is guaranteed by constraints (2), (9) and (10). We remark that while the mechanism constructed from the LP variables may not be unique, effectively this makes no difference, since the parts of the mechanism that are chosen arbitrarily can never be accessed when executing the mechanism. This is because $z(h, s) = 0$ only if at some point in the history h, there is an action that the mechanism would never play given the reported states and actions before that. In particular, the above does not simply apply to all history-state pairs (h, s) that are reached with probability 0 under truthful reporting, in which case $z(h, s)$ may still be positive due to the extended transition operator. Moreover, given any mechanism, one can construct LP variables in a similar way, such that the mechanism constructed from these variables is the same as the original mechanism (modulo the unreachable parts). In other words, the above correspondence is effectively bijective.

The objective. The objective function of the LP (Eq. (1)) is precisely the overall utility of the principal under the mechanism constructed above, assuming truthful reporting. This is captured by the following lemma.

Lemma 1. *Let $M = (\pi, p)$ be the mechanism constructed from variables $x(h, s, a)$, $y(h, s)$, and $z(h, s)$ which satisfy the flow constraints. Then*

$$u_P^M(\emptyset) = \sum_{h \in \mathcal{H}, s \in \mathcal{S}: (h,s) \text{ is feasible}} \left(\sum_{a \in \mathcal{A}} v_{|h|+1}^P(s, a) \cdot x(h, s, a) + y(h, s) \right).$$

From this lemma, it is clear that the objective of the LP is the natural quantity to maximize.

Utility. The utility constraints (Eq. (5)) collect the agent's onward utility, where $u(h, s)$ is equal to the agent's onward utility in state s from history h, assuming truthful reporting, scaled by $z(h, s)$. This is captured by the following lemma.

Lemma 2. *Let $M = (\pi, p)$ be the mechanism constructed from variables $x(h, s, a)$, $y(h, s)$, and $z(h, s)$ which satisfy the flow and utility constraints. For all $h \in \mathcal{H}$, $s \in \mathcal{S}$,*

$$u(h, s) = z(h, s) \cdot u_A^M(h, s).$$

The proof of Lemma 2 is essentially the same as that of Lemma 1. Given the correspondence to the agent's utility $u_A^M(h, s)$, the utility variables $u(h, s)$ act as auxiliary variables in IC constraints.

IC constraints. IC constraints are a key component of the LP formulation. There are two families of IC constraints: collecting the agent's scaled utility from single-step misreporting (Eq. (6)), and subsequently restricting the mechanism such that there is no incentive for misreporting (Eq. (7)). In Eq. (6), we build variables $u(h, s, s')$, which is supposed to be the onward utility of the agent in state s from history h misreporting s', assuming truthful reporting in the future, scaled by $z(h, s')$ (rather than $z(h, s)$). This is captured by the following lemma.

Lemma 3. *Let $M = (\pi, p)$ be the mechanism constructed from variables $x(h, s, a)$, $y(h, s)$, and $z(h, s)$ which satisfy the flow constraints, the utility constraints, and Eq.* (6)*. Then the following statement holds: for all $h \in \mathcal{H}$, $s, s' \in \mathcal{S}$, let reporting strategy $r_{h,s,s'}$ be such that*

$$r_{h,s,s'}(h', s'') = \begin{cases} s', & \text{if } h = h' \text{ and } s = s'' \\ s'', & \text{otherwise} \end{cases}.$$

That is, $r_{h,s,s'}$ misreports s' only in state s from history h, and reports truthfully otherwise. Then for all $h \in \mathcal{H}$, $s, s' \in \mathcal{S}$,

$$u(h, s, s') = z(h, s') \cdot u_A^{M, r_{h,s,s'}}(h, s).$$

Given Lemma 3, Eq. (7) then guarantees that the mechanism M is robust against single-step misreporting for all reachable history-state pairs.

Lemma 4. *Let $M = (\pi, p)$ be the mechanism constructed from variables $x(h, s, a)$, $y(h, s)$, and $z(h, s)$ which satisfy the flow constraints, the utility constraints, and Eq.* (6)*. The following is true if and only if the LP variables also satisfy Eq.* (7)*: for all $h \in \mathcal{H}$, $s, s' \in \mathcal{S}$ where (h, s) is reachable by the mechanism M,*

$$u_A^M(h, s) \geq u_A^{M, r_{h,s,s'}}(h, s).$$

We then show that a mechanism is IC if and only if there is no incentive for single-step misreporting, which directly implies that the mechanism M constructed from the LP variables is IC. This is captured by the following lemma.

Lemma 5. *Let $M = (\pi, p)$ be the mechanism constructed from variables $x(h, s, a)$, $y(h, s)$, and $z(h, s)$ which satisfy the flow constraints, the utility constraints, and Eq.* (6)*. Then M is IC if and only if the LP variables also satisfy Eq.* (7)*.*

IR constraints, feasible actions, and feasible payments. These constraints are straightforward given the correspondence between the LP variables and the mechanism that we have discussed above. Note that while Eq. (8) is for dynamic IR (i.e., the agent has no incentive to leave the mechanism at any point) and Eq. (10) is for nonnegative payments, it is easy to replace them with similar constraints that correspond to overall IR or no payments. See Appendix C for more details.

Optimality of LP solution. Given the above facts, we are ready to state and prove the main result of the paper.

Theorem 2. *There is an algorithm which computes an optimal IC and (optionally) IR dynamic mechanism, with or without payments, in time $O(\text{poly}(|\mathcal{S}|^T, |\mathcal{A}|^T, L))$, where L is the number of bits required to encode each of the input parameters. In particular, when T is constant, the algorithm runs in polynomial time.*

4 The Case of Myopic Agents: Characterization and Faster Algorithm

In this section, we briefly discuss a special case of the problem of computing optimal dynamic mechanisms, namely the case where the agent is myopic, or, equivalently, the agent has a discount factor of 0. While our LP-based algorithm still applies, as we will see below, optimal mechanisms for myopic agents enjoy a succinct representation in this case, which also enables a faster algorithm that scales only linearly in the time horizon T. See Appendix D for more details, including the formal definition of myopic agents and the complete description of the algorithm.

4.1 Characterization of Optimal Mechanisms

We first show that when the agent is myopic, without loss of generality, the actions and payments specified by an optimal mechanism depend only on the time, the previous state, the previous action and the current state (we call such a mechanism a *succinct mechanism*), instead of the entire history-state pair.

Lemma 6. *Fix a dynamic environment. When the agent is myopic, for any IC mechanism $M = (\pi, p)$, there is another IC mechanism $M' = (\pi', p')$ (which is IR whenever M is) such that*

- $u_P^{M'}(\emptyset) \geq u_P^M(\emptyset)$, *and*
- *for all $h \in \mathcal{H}$, $s \in \mathcal{S}$, π' and p' depend only on $|h|$, s_p, a_p and s, where $(s_p, a_p) = \text{last}(h)$.*

Moreover, the above is true regardless of whether payments are allowed, or which IR constraints are required.

4.2 Faster Algorithm for Myopic Agents

Based on the above characterization, we present a faster algorithm for computing an optimal mechanism in the face of a myopic agent. In particular, the time complexity of this algorithm depends only linearly on the time horizon T, making it feasible for dynamic environments with a long time horizon. This is in contrast with the case of patient agents, for which, as we have seen, the long-horizon problem is hard to approximate. The algorithm uses as a subroutine a blackbox algorithm that computes an optimal IC (and optionally IR) mechanism in static environments, with or without payments. It is known that such an algorithm can be implemented using linear programming, and in some cases in more efficient ways [17, 19, 56].

Theorem 3. *When the agent is myopic, Algorithm 1 computes an optimal IC and (optionally) IR dynamic mechanism, with or without payments, in time*

$$O(T|\mathcal{S}||\mathcal{A}| \cdot T_{\text{stat}}(|\mathcal{S}|, |\mathcal{A}|, L)) = O(T \cdot \text{poly}(|\mathcal{S}|, |\mathcal{A}|, L)),$$

where T_{stat} is the time complexity of the blackbox algorithm used for computing an optimal IC (and optionally IR) mechanism in static environments, and L is the number of bits required to encode each of the input parameters.

5 Conclusion

We studied automated dynamic mechanism design and showed that, while it is computationally hard to find (even approximately) optimal mechanisms when (1) facing a patient agent and (2) the horizon is long, when either of these two conditions is dropped, an optimal mechanism can be found efficiently. An interesting future direction is to generalize our results to related problems with a stronger learning flavor, e.g., reinforcement learning with IC and/or IR constraints.

Besides using our algorithms directly for appropriate applications, the experimental results that they enable (including those that we presented in Appendix F) can guide new theory. For example, can we rigorously prove the benefit of facing a patient agent when the setting is not all too adversarial, and perhaps even characterize the transition point at which facing a patient agent becomes better than facing a myopic one? Analytically derived mechanisms can also be compared to these experimental results to see how close to optimal in performance they are. Finally, close inspection of the actual mechanisms generated by our algorithms may reveal insights that can be used to analytically design new mechanisms.

Funding Transparency Statement

Funding in direct support of this work: NSF grant IIS-1814056.

References

[1] Itai Ashlagi, Constantinos Daskalakis, and Nima Haghpanah. Sequential mechanisms with ex-post participation guarantees. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 213–214, 2016.

[2] Susan Athey and Ilya Segal. An efficient dynamic mechanism. *Econometrica*, 81(6):2463–2485, 2013.

[3] Maria-Florina Balcan, Tuomas Sandholm, and Ellen Vitercik. Sample complexity of automated mechanism design. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2091–2099, 2016.

[4] Maria-Florina Balcan, Tuomas Sandholm, and Ellen Vitercik. A general theory of sample complexity for multi-item profit maximization. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 173–174, 2018.

[5] Santiago R Balseiro, Huseyin Gurkan, and Peng Sun. Multiagent mechanism design without money. *Operations Research*, 67(5):1417–1436, 2019.

[6] David P Baron and David Besanko. Regulation and information in a continuing relationship. *Information Economics and policy*, 1(3):267–302, 1984.

[7] Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, 6(5): 679–684, 1957.

[8] Dirk Bergemann and Juuso Välimäki. The dynamic pivot mechanism. *Econometrica*, 78(2): 771–789, 2010.

[9] Dirk Bergemann and Juuso Välimäki. Dynamic mechanism design: An introduction. *Journal of Economic Literature*, 57(2):235–74, 2019.

[10] Branislav Bosansky and Jiri Cermak. Sequence-form algorithm for computing stackelberg equilibria in extensive-form games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[11] Branislav Bošanskỳ, 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.

[12] Andrea Celli, Stefano Coniglio, and Nicola Gatti. Private bayesian persuasion with sequential games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1886–1893, 2020.

[13] Jiri Cermak, Branislav Bosansky, Karel Durkota, Viliam Lisy, and Christopher Kiekintveld. Using correlated strategies for computing stackelberg equilibria in extensive-form games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[14] Jakub Černỳ, Branislav Boỳanskỳ, and Christopher Kiekintveld. Incremental strategy generation for stackelberg equilibria in extensive-form games. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 151–168, 2018.

[15] Jakub Černỳ, Branislav Bosanskỳ, and Bo An. Finite state machines play extensive-form games. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 509–533, 2020.

[16] Xi Chen and Xiaotie Deng. Settling the complexity of two-player nash equilibrium. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 261–272. IEEE, 2006.

[17] Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. *arXiv preprint cs/0205075*, 2002.

[18] Vincent Conitzer and Tuomas Sandholm. Self-interested automated mechanism design and implications for optimal combinatorial auctions. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 132–141, 2004.

[19] Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 82–90, 2006.

[20] Pascal Courty and Li Hao. Sequential screening. *The Review of Economic Studies*, 67(4): 697–717, 2000.

[21] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

[22] P Dütting, F Fischer, P Jirapinyo, J Lai, B Lubin, and DC Parkes. Payment rules through discriminant-based classifiers. *ACM Transactions on Economics and Computation*, 2015.

[23] Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning. In *International Conference on Machine Learning*, pages 1706–1715. PMLR, 2019.

[24] Jeffrey C Ely. Beeps. *American Economic Review*, 107(1):31–53, 2017.

[25] Péter Eső and Balazs Szentes. Optimal information disclosure in auctions and the handicap auction. *The Review of Economic Studies*, 74(3):705–731, 2007.

[26] Zhe Feng, Harikrishna Narasimhan, and David C Parkes. Deep learning for revenue-optimal auctions with budgets. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, pages 354–362, 2018.

[27] Rupert Freeman, Seyed Majid Zahedi, Vincent Conitzer, and Benjamin C Lee. Dynamic proportional sharing: A game-theoretic approach. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(1):1–36, 2018.

[28] Jiarui Gan, Rupak Majumdar, Goran Radanovic, and Adish Singla. Bayesian persuasion in sequential decision-making. *arXiv preprint arXiv:2106.05137*, 2021.

[29] Artur Gorokh, Siddhartha Banerjee, and Krishnamurthy Iyer. From monetary to non-monetary mechanism design via artificial currencies. *Available at SSRN 2964082*, 2019.

[30] Mingyu Guo, Vincent Conitzer, and Daniel M Reeves. Competitive repeated allocation without payments. In *International Workshop on Internet and Network Economics*, pages 244–255. Springer, 2009.

[31] Mohammad Taghi Hajiaghayi, Robert Kleinberg, and Tuomas Sandholm. Automated online mechanism design and prophet inequalities. In *AAAI*, volume 7, pages 58–65, 2007.

[32] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4): 798–859, 2001.

[33] Ronald A Howard. Dynamic programming and markov processes. 1960.

[34] Andrew Kephart and Vincent Conitzer. Complexity of mechanism design with signaling costs. In *AAMAS*, pages 357–365. Citeseer, 2015.

[35] Andrew Kephart and Vincent Conitzer. The revelation principle for mechanism design with reporting costs. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 85–102, 2016.

[36] Daphne Koller, Nimrod Megiddo, and Bernhard Von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and economic behavior*, 14(2):247–259, 1996.

[37] Christian Kroer, Gabriele Farina, and Tuomas Sandholm. Robust stackelberg equilibria in extensive-form games and extension to limited lookahead. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[38] Joshua Letchford and Vincent Conitzer. Computing optimal strategies to commit to in extensive-form games. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 83–92, 2010.

[39] Vahab Mirrokni, Renato Paes Leme, Pingzhong Tang, and Song Zuo. Non-clairvoyant dynamic mechanism design. *Econometrica*, 88(5):1939–1963, 2020.

[40] Martin Mundhenk, Judy Goldsmith, Christopher Lusena, and Eric Allender. Complexity of finite-horizon markov decision process problems. *Journal of the ACM (JACM)*, 47(4):681–720, 2000.

[41] Roger B Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58–73, 1981.

[42] Roger B Myerson. Multistage games with communication. *Econometrica: Journal of the Econometric Society*, pages 323–358, 1986.

[43] Harikrishna Narasimhan, Shivani Brinda Agarwal, and David C Parkes. Automated mechanism design without money via machine learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2016.

[44] Christos Papadimitriou, George Pierrakos, Christos-Alexandros Psomas, and Aviad Rubinstein. On the complexity of dynamic mechanism design. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1458–1475. SIAM, 2016.

[45] Christos H Papadimitriou and John N Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

[46] Alessandro Pavan. Dynamic mechanism design: Robustness and endogenous types. In *Advances in Economics and Econometrics: Eleventh World Congress*, pages 1–62, 2017.

[47] Alessandro Pavan, Ilya Segal, and Juuso Toikka. Dynamic mechanism design: A myersonian approach. *Econometrica*, 82(2):601–653, 2014.

[48] Martin L Puterman and Moon Chirl Shin. Modified policy iteration algorithms for discounted markov decision problems. *Management Science*, 24(11):1127–1137, 1978.

[49] Jad Rahme, Samy Jelassi, Joan Bruna, and S Matthew Weinberg. A permutation-equivariant neural network architecture for auction design. *arXiv preprint arXiv:2003.01497*, 2020.

[50] Jérôme Renault, Eilon Solan, and Nicolas Vieille. Optimal dynamic information provision. *Games and Economic Behavior*, 104:329–349, 2017.

[51] Tuomas Sandholm and Anton Likhodedov. Automated design of revenue-maximizing combinatorial auctions. *Operations Research*, 63(5):1000–1025, 2015.

[52] Tuomas Sandholm, Vincent Conitzer, and Craig Boutilier. Automated design of multistage mechanisms. In *IJCAI*, volume 7, pages 1500–1506, 2007.

[53] Weiran Shen, Pingzhong Tang, and Song Zuo. Automated mechanism design via neural networks. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, pages 215–223, 2019.

[54] Eva Tardos and Vijay V Vazirani. Basic solution concepts and computational issues. *Algorithmic game theory*, pages 3–28, 2007.

[55] Bernhard von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2):220–246, 1996.

[56] Hanrui Zhang, Yu Cheng, and Vincent Conitzer. Automated mechanism design for classification with partial verification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See appendices.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Overview of Results in Appendices

In Section C, we discuss ways of customizing the LP formulation given in Section 3 to accommodate richer objectives and/or constraints, such as feasible intervals of payments, different IR constraints and discount factors. We also present an integer LP formulation for finding optimal deterministic mechanisms.

In Section D, we zoom into a special case of the problem where the agent is *myopic*, i.e., where the agent cares only about immediate value when making decisions. This is still practically meaningful, since it is commonly assumed and observed that the principal is often much more patient than the agent in dynamic environments. (This could also correspond to the agent really being a sequence of short-lived agents; for example, there may be high turnover in the research group in the example above, where each researcher is there only for one period.) We show that in such cases, without loss of generality, optimal mechanisms admit succinct representations, i.e., they depend only on the current state and time, the previous state, and the previous action. Based on this characterization, we provide an improved algorithm for finding optimal mechanisms in the face of a myopic agent, whose time complexity depends *linearly* on the time horizon. As a result, this algorithm scales well in

dynamic environments with long time horizons, which is in sharp contrast to the general case where long time horizons lead to inapproximability.

As discussed above, without strategic behavior, our problem degenerates to the problem of planning in (finite episodic) Markov Decision Processes (MDPs). It is known that in MDPs, optimal strategies are without loss of generality *memoryless*: they depend only on the current time and state. To this end, one may wonder if memoryless mechanisms are also (approximately) optimal and/or easier to compute in dynamic environments with strategic behavior. In Section E, we give a negative answer to the above question, by showing that (1) the principal's optimal utility achieved by memoryless mechanisms can be arbitrarily worse than that achieved by general dynamic mechanisms, and (2) it is NP-hard to approximate the principal's optimal utility achieved by memoryless mechanisms within a factor of $(7/8 + \varepsilon)$ for any $\varepsilon > 0$. In other words, memoryless mechanisms do not provide a good solution for our problem, in terms of both optimality and computational tractability.

Finally, in Section F, we apply our algorithms to synthetic dynamic environments with different characteristics, in order to provide a proof of concept for the methods we propose, as well as to explore various phenomena in dynamic environments and their implications for (automated) dynamic mechanism design. Below are some of our key findings:

- As in static environments, taking into consideration the agent's incentives in dynamic environments can greatly improve the principal's utility.
- In dynamic environments, optimal mechanisms are remarkably robust to misaligned interests between the principal and the agent, whereas the performance of naïve mechanisms (which disregard the agent's incentives) degrades much faster.
- Even when the principal's and the agent's valuations are perfectly aligned, an agent acting myopically can still considerably hurt the principal's utility in naïve mechanisms, but this can be largely corrected by using mechanisms that are optimal in the face of a myopic agent.
- As one would expect, patient agents are easier to cooperate with, and myopic agents are easier to exploit; however, even when the principal's and the agent's valuations are negatively correlated, it is possible to find a middle ground where cooperation is more beneficial than exploitation in the long run.

B Additional Related Work

Repeated allocation without money. Another related line of work is devoted to studying the design of repeated allocation mechanisms without money, motivated for example by allocating shared computing resources over time [30, 27, 5, 29]. When there is no money, repeated allocation allows one to better take current preferences for the items into account, because one can "pay" for one's current allocation with one's future allocations. Indeed, a common theme of this line of work is to introduce artificial currencies or to approximate mechanisms *with* money via the use of future allocations. The algorithms we present here can be used to find optimal mechanisms without money directly.

Equilibrium computation. Our main result can be viewed as an efficient algorithm for computing Stackelberg equilibria in a special class of extensive-form games. Equilibrium computation is quite well understood in normal-form games, where there are polynomial-time algorithms for computing a Stackelberg equilibrium [19], or a Nash equilibrium when the game is zero-sum (see, e.g., [54]), in two-player games, whereas finding a Nash equilibrium in general-sum two-player games is already PPAD-complete [21, 16]. For extensive-form games, von Stengel [55] and Koller et al. [36] propose the sequence-form representation, which leads to an efficient algorithm for finding a Nash equilibrium (which is also a Stackelberg equilibrium) in two-player zero-sum games. However, as shown by Letchford and Conitzer [38], computing a Stackelberg equilibrium in two-player general-sum extensive-form games is NP-hard in general. Polynomial-time (exact or $(1 - \varepsilon)$-approximation) algorithms are known only for highly restrictive cases, e.g., in perfect-information settings [11], or when the follower is a finite state machine with limited memory [15] (although practically scalable algorithms exist for more general settings [10, 13, 14, 37]). Our results push the boundary of tractability of Stackelberg equilibrium in extensive-form games, by enabling efficient computation in a nontrivial class of *general-sum* extensive-form games with *imperfect information*.

C Customizing the LP Formulation.

The LP formulation in Figure 1 allows for nonnegative payments, assumes that the principal cares about payments as much as the agent, and enforces dynamic IR constraints. As mentioned above, one can customize all these components by modifying the corresponding parts of the LP formulation. Below we discuss several ways of customization.

- **Unequal valuations for payments**: in the case where the principal has utility c for one unit of payment (whereas without loss of generality the agent has utility 1), one may replace the objective function (Eq. (1)) with

$$\max \sum_{h\in\mathcal{H},s\in\mathcal{S}:(h,s) \text{ is feasible}} \left(\sum_{a\in\mathcal{A}} v^P_{|h|+1}(s,a)\cdot x(h,s,a) + c\cdot y(h,s) \right).$$

  Note that our formulation works only when the principal cares linearly about payments. Notably, the principal may not care about payments at all (as in the case of paying the agent in "brownie points"), or even dislike payments made by the agent (as in the case where the agent is asked to expend useless effort or "burn money" and the principal cares in part about the resulting loss of welfare).

- **No payments**: to forbid payments in the mechanism, one can simply replace Eq. (10) with

$$y(h,s) = 0,\ \forall h\in\mathcal{H}, s\in\mathcal{S}.$$

- **Feasible intervals of payments**: more generally, one may wish to specify a feasible interval $[a_{h,s}, b_{h,s}]$ for the payment at each history-state pair (h,s) such that $a_{h,s}\le p(h,s)\le b_{h,s}$, which subsumes both nonnegative payments and no payments as special cases. This can be done by replacing Eq. (10) with

$$a_{h,s}\cdot z(h,s) \le y(h,s) \le b_{h,s}\cdot z(h,s),\ \forall h\in\mathcal{H}, s\in\mathcal{S}.$$

- **Overall/no IR**: when the agent can choose whether to participate in the mechanism, but cannot leave halfway (corresponding to an overall IR constraint), one can replace Eq. (8) with

$$\sum_{s\in\mathcal{S}} u(\emptyset, s)\ge 0.$$

  Also, when leaving the mechanism is not an option for the agent from the very beginning (corresponding to no IR constraint), one may remove IR constraints simply by removing Eq. (8).

- **Discount factors**: to accommodate the case where the agent has a discount factor $0\le\delta<1$, one can modify the LP formulation in the following way:
  - Replace Eq. (5) with

$$u(h,s) = \sum_{h'\in\mathcal{H},s'\in\mathcal{S}:(h,s)\subseteq(h',s')} \delta^{|h'|-|h|}\cdot\left(\sum_{a\in\mathcal{A}} v^A_{|h'|+1}(s',a)\cdot x(h',s',a) - y(h',s') \right),\ \forall h\in\mathcal{H}, s\in\mathcal{S}.$$

  - Replace Eq. (6) with

$$u(h,s,s') = \sum_{a\in\mathcal{A}} v^A_{|h|+1}(s,a)\cdot x(h,s',a) - y(h,s') + \delta\cdot\sum_{a\in\mathcal{A},s''\in\mathcal{S}} \frac{P_{|h|+1}(s,a,s'')}{P^E_{|h|+1}(s',a,s'')}\cdot u(h+(s',a),s''),\ \forall h\in\mathcal{H}, s,s'\in\mathcal{S}$$

- **Deterministic mechanisms**: the problem of computing an optimal deterministic mechanism is NP-hard even in static environments [17, 18]. Nevertheless, given our LP formulation, one can restrict the mechanism to be deterministic by introducing Boolean variables, resulting in a mixed integer LP. While integer LPs are hard to solve in a worst-case sense, real-world problems often admit certain structures which can be exploited by commercial solvers such as CPLEX and Gurobi. To be specific, we introduce a Boolean variable $c(h,s,a)$ which controls $x(h,s,a)$

for all $h \in \mathcal{H}$, $s \in \mathcal{S}$, and $a \in \mathcal{A}$, and ensures that fixing h and s, $x(h,s,a)$ can be positive for at most one action $a \in \mathcal{A}$. This is implemented by the following constraints (in addition to the existing ones):

$$\begin{aligned} & x(h,s,a) \le c(h,s,a) && \forall h \in \mathcal{H}, s \in \mathcal{S}, a \in \mathcal{A} \\ & \sum_{a \in \mathcal{A}} c(h,s,a) = 1 && \forall h \in \mathcal{H}, s \in \mathcal{S} \\ & c(h,s,a) \in \{0,1\} && \forall h \in \mathcal{H}, s \in \mathcal{S}, a \in \mathcal{A}. \end{aligned}$$

We also remark that the above discussion is non-exhaustive: one can impose richer restrictions by modifying the LP formulation in other linear ways, and/or combining the above modifications.

D The Case of Myopic Agents: Characterization and Faster Algorithm

In this section, we consider a special case of the problem of computing optimal dynamic mechanisms, namely the case where the agent is myopic, or, equivalently, the agent has a discount factor of 0. While our LP-based algorithm still applies, as we will see below, optimal mechanisms for myopic agents enjoy a succinct representation in this case, which also enables a faster algorithm that scales only linearly in the time horizon T.

Myopic agents. The utility u_A^M of a myopic agent under mechanism M is such that

$$u_A^M(h,s) = \sum_a \pi(h,s,a) \cdot v^A_{|h|+1}(s,a) - p(h,s),$$

where $u_A^M(h,s) = 0$ for all $h \in \mathcal{H}_T$ and $s \in \mathcal{S}$. Given a reporting strategy r, the utility $u_A^{M,r}$ of the agent under mechanism M and reporting strategy r is

$$u_A^{M,r}(h,s) = \sum_a \pi(r(h), r(h,s), a) \cdot v^A_{|h|+1}(s,a) - p(r(h), r(h,s)).$$

M is IC if and only if for all $h \in \mathcal{H}$ and $s \in \mathcal{S}$, there are no future reporting strategies that lead to better utility, i.e., for every reporting strategy r where $r(h',s') = s'$ whenever $|h'| < |h|$,

$$u_A^M(h,s) \ge u_A^{M,r}(h,s).$$

Note that since the agent is myopic, it is insufficient to simply require $u_A^M(\emptyset) \ge u_A^{M,r}(\emptyset)$. Also, it is necessary to restrict misreporting to the future, since otherwise the agent would be allowed and sometimes incentivized to change the past, leading to unrealistically strong IC requirements. Again, since the revelation principle holds, we focus only on IC mechanisms.

D.1 Characterization of Optimal Mechanisms

We first show that when the agent is myopic, without loss of generality, the actions and payments specified by an optimal mechanism depend only on the time, the previous state, the previous action and the current state (we call such a mechanism a *succinct mechanism*), instead of the entire history-state pair.

Lemma 7. *Fix a dynamic environment. When the agent is myopic, for any IC mechanism $M = (\pi, p)$, there is another IC mechanism $M' = (\pi', p')$ (which is IR whenever M is) such that*

- $u_P^{M'}(\emptyset) \ge u_P^M(\emptyset)$*, and*
- *for all $h \in \mathcal{H}$, $s \in \mathcal{S}$, π' and p' depend only on $|h|$, s_p, a_p and s, where $(s_p, a_p) = \mathrm{last}(h)$.*

Moreover, the above is true regardless of whether payments are allowed, or which IR constraints are required.

Algorithm 1: Algorithm for computing an optimal mechanism against a myopic agent.

Input: Time horizon T, transition probabilities $\{P_t\}_{t\in[T]}$, principal's valuation functions $\{v_t^P\}_{t\in[T]}$, agent's valuation functions $\{v_t^A\}_{t\in[T]}$.

Output: An optimal IC (for a myopic agent) mechanism $M = (\pi, p)$.

for $t = T, T-1, \ldots, 1$ **do**
  for $s \in \mathcal{S}$, $a \in \mathcal{A}$ **do**
    let $u(t, s, a) \leftarrow v_t^P(s, a) + \sum_{s'\in\mathcal{S}} P_t(s, a, s') \cdot u_P^M(t+1, s, a, s')$;
    /* the above operation is well-defined, in particular because $u_P^M(t+1, s, a, s')$ depends only on the part of M that has already been computed */
  end
  for $s_p \in \mathcal{S}, a_p \in \mathcal{A}$ **do**
    let $(\pi', p') \leftarrow$ OptStatMech$(\mathcal{S}, \mathcal{A}, \{P_{t-1}(s_p, a_p, s)\}_s, \{u(t, s, a)\}_{s,a}, \{v_t^A(s, a)\}_{s,a})$;
    /* call OptStatMech to compute an optimal static mechanism (π', p'), in a static environment with type space $\mathcal{S}$, action space $\mathcal{A}$, population distribution $\{P_{t-1}(s_p, a_p, s)\}_s$, principal's utility function $\{u(t, s, a)\}_{s,a}$, and agent's utility function $\{v_t^A(s, a)\}_{s,a}$ */
    for $s \in \mathcal{S}$ **do**
      let $\pi(t, s_p, a_p, s) \leftarrow \pi'(s)$, and $p(t, s_p, a_p, s) \leftarrow p'(s)$;
    end
  end
end
return $M = (\pi, p)$;

D.2 Faster Algorithm for Myopic Agents

Based on the above characterization, we present below a faster algorithm for computing an optimal mechanism in the face of a myopic agent. In particular, the time complexity of this algorithm depends only linearly on the time horizon T, making it feasible for dynamic environments with a long time horizon. This is in contrast with the case of patient agents, for which, as we have seen, the long-horizon problem is hard to approximate.

To improve readability, we use the following shorthand notation for succinct mechanisms. For a succinct mechanism $M = (\pi, p)$, for any $h \in \mathcal{H}$ and $s \in \mathcal{S}$, let $\pi(t, s_p, a_p, s) = \pi(h, s)$ be the action policy at (h, s), and $p(t, s_p, a_p, s) = p(h, s)$ be the payment function, where $(s_p, a_p) = \mathrm{last}(h)$ and $t = |h| + 1$. Also, observe that the principal's onward utility at any history-state pair (h, s) depends only on the previous state s_p, the previous action a_p, and the current state s. In such cases, we also denote this utility by $u_P^M(t, s_p, a_p, s) = u_P^M(h, s)$.

The full algorithm is given as Algorithm 1. It uses as a subroutine an algorithm OptStatMech which computes an optimal IC (and optionally IR) mechanism in static environments, with or without payments. It is known that such an algorithm can be implemented using linear programming, and in some cases in more efficient ways [17, 19, 56]. Algorithm 1 proceeds in an inductive fashion, building a succinct mechanism backwards, one layer at a time. It repeatedly solves the problem of maximizing the principal's expected onward utility over the current state s, given the previous state s_p and the previous action a_p. Since s_p and a_p together induce a roll-in distribution over the state space, this problem can be reduced to computing an optimal static mechanism, where the valuation function of the principal depends on the optimal mechanism in the following layers. This can then be solved by calling OptStatMech, the algorithm for computing an optimal static mechanism. Below we state and prove the correctness and computational efficiency of Algorithm 1.

Theorem 4. *When the agent is myopic, Algorithm 1 computes an optimal IC and (optionally) IR dynamic mechanism, with or without payments, in time*

$$O(T|\mathcal{S}||\mathcal{A}| \cdot T_{\mathrm{stat}}(|\mathcal{S}|, |\mathcal{A}|, L)) = O(T \cdot \mathrm{poly}(|\mathcal{S}|, |\mathcal{A}|, L)),$$

where T_{stat} is the time complexity of OptStatMech*, and L is the number of bits required to encode each of the input parameters.*

Customizing Algorithm 1. We remark that Algorithm 1 can also be customized to allow for unequal valuations of payments, feasible intervals of payments, etc. Moreover, it can be adapted to compute an optimal deterministic mechanism, by requiring OptStatMech to compute an optimal deterministic static mechanism. Again, while this is generally hard to compute, for practical purposes, it is reasonable to expect that OptStatMech implemented using commercial mixed integer LP solvers (or in other practically efficient ways) can find an optimal mechanism efficiently.

E Infeasibility of Memoryless Mechanisms

From a planning perspective, automated dynamic mechanism design can be viewed equivalently as planning in MDPs where the current state cannot be directly observed, but instead, has to be reported by a strategic agent whose interest may not align with the planner's. In particular, when the planner and the agent share the same valuation function, automated dynamic mechanism design degenerates to the classical problem of planning in episodic MDPs with a finite planning horizon. In the latter problem, it is well known that without loss of generality, any optimal policy depends only on the time and the current state, i.e., it is memoryless. And moreover, such optimal policies can be computed in polynomial time. In light of the above facts, the following questions arise naturally: are there (approximately) optimal mechanisms that are also memoryless, and can we find optimal memoryless mechanisms efficiently? In this section, we give negative answers to both questions, which means memoryless mechanisms are generally infeasible for dynamic environments. We first show that memoryless mechanisms can be arbitrarily worse than general, history-dependent mechanisms, against both patient and myopic agents.

Theorem 5. *Regardless of whether the agent is myopic, for any $\varepsilon > 0$, there is a dynamic environment where the principal's utility under an optimal memoryless mechanism is at most an ε fraction of the principal's optimal utility.*

Now we show that on top of the suboptimality, optimal memoryless mechanisms are computationally hard to approximate.

Theorem 6. *Regardless of whether the agent is myopic, it is* NP*-hard to approximate the principal's maximum utility under memoryless mechanisms within a factor of $7/8 + \varepsilon$ for any $\varepsilon > 0$.*

F Experimental Results

In this section, we present experimental results where our algorithms are applied to synthetic dynamic environments of different characteristics. The main goals of the experiments are

- to provide a proof of concept for the methods proposed in this paper,
- to illustrate the necessity of considering incentives when planning in dynamic environments (as opposed to disregarding the agent's valuations and treating the problem simply as an MDP based on the principal's valuations),
- to study the effect of cooperation and competition in dynamic mechanism design, and
- to understand the difference between patient and myopic agents from the principal's perspective, especially when the parameters of the environment vary.

F.1 Setup of Experiments

Mechanisms/models of the agent under consideration. For each dynamic environment examined, we consider the following quantities from different combinations of mechanisms and models of the agent:

- **Naïve mechanisms facing a naïve agent**: the principal's optimal utility facing a naïve agent who always reports truthfully, i.e., the optimal utility when treating the problem simply as an MDP based on the principal's valuations.
- **Naïve mechanisms facing a patient agent**: the principal's utility, when executing the optimal mechanism/policy for naïve agents, facing a strategic agent who is patient.

- **Naïve mechanisms facing a myopic agent**: the principal's utility, when executing the optimal mechanism/policy for naïve agents, facing a strategic agent who is myopic.
- **Patient mechanisms facing a patient agent**: the principal's optimal utility facing a strategic agent who is patient.
- **Myopic mechanisms facing a myopic agent**: the principal's optimal utility facing a strategic agent who is myopic.

For simplicity, payments are not allowed in any of our experiments.

Dynamic environments. To manifest the effect of cooperation and competition, we generate synthetic dynamic environments in the following way:

- Fix the time horizon T, number of states $|\mathcal{S}|$, number of actions $|\mathcal{A}|$, and correlation parameter $\eta \in [-1, 1]$ (explained below).
- Let the initial distribution P_0 be a random distribution generated in the following way: for each state s, we generate a uniformly random real number $\mathrm{rand}(s)$ between 0 and 1, which is proportional to $P_0(s)$. That is, $P_0(s) = \mathrm{rand}(s) / \left(\sum_{s'} \mathrm{rand}(s')\right)$.
- For each $t \in [T]$, $s \in \mathcal{S}$ and $a \in \mathcal{A}$, we generate the transition distribution $P_t(s, a)$ independently in the same way that P_0 is generated.
- For each $t \in [T]$, $s \in \mathcal{S}$ and $a \in \mathcal{A}$, let $v_t^P(s, a)$ be an independent, uniformly random real number between 0 and 1.
- For each $t \in [T]$, $s \in \mathcal{S}$ and $a \in \mathcal{A}$, let $v_t^A(s, a) = \eta \cdot v_t^P(s, a) + (1 - |\eta|) \cdot \mathrm{rand}(t, s, a)$, where $\mathrm{rand}(t, s, a)$ is an independent, uniformly random real number between 0 and 1.

The correlation parameter η controls the extent to which the interests of the principal and the agent are (mis)aligned. In particular, if $\eta = 1$, then the principal and the agent have exactly the same valuations, corresponding to full cooperation. If $\eta = -1$, then the principal and the agent are in a zero-sum situation, corresponding to full competition.

F.2 Summary of Experimental Results

Suboptimality of naïve mechanisms. As we can see from Figure 2, even when the state and action spaces are extremely simple, i.e., there are only 2 states and 2 actions, when the correlation parameter $\eta = -1$ (i.e., when the agent acts adversarially), naïve mechanisms facing a strategic agent can only achieve about 75% of the naïve benchmark, i.e., the optimal utility when the agent is naïve. When $\eta = 0$ (i.e., when the agent's and principal's valuations are independent), naïve mechanisms facing a strategic agent still achieve only 85% of the naïve benchmark. On the other hand, the respective optimal mechanisms facing a patient or myopic agent consistently achieve about 95% of the naïve benchmark. This gap is further amplified in Figure 3: as the environment becomes more and more complex (i.e., the numbers of states and actions become larger and larger), the utility of naïve mechanisms facing a strategic agent drops below 20% of the naïve benchmark when $\eta = -1$, and to about 50% when $\eta = 0$. In contrast, the respective optimal mechanisms facing a patient or myopic agent still achieve about 70% of the naïve benchmark even when $\eta = -1$. These phenomena suggest that when the agent is not fully cooperative, taking strategic behavior into consideration significantly improves the principal's utility, even in extremely simple dynamic environments. Moreover, the more complex the environment is, the larger this gap becomes.

Another interesting fact to note is that even when the principal's and the agent's valuations are exactly the same (i.e., when $\eta = 1$), naïve mechanisms are still suboptimal facing a myopic agent, since the agent may sacrifice greater long-term gain in exchange for smaller immediate value. This phenomenon is more significant in Figure 2, especially in environments with longer time horizons. In such cases, taking into consideration the fact that the agent is myopic mitigates the loss, and recovers almost all the utility of the naïve benchmark.

Effect of cooperation and competition. As the correlation parameter increases, both Figure 2 and Figure 3 show clear upward trends in all the quantities that we consider (except for the naïve benchmark which is always normalized to 1), as one would expect. Nevertheless, we note the following facts from the figures: compared to naïve mechanisms, optimal mechanisms facing a

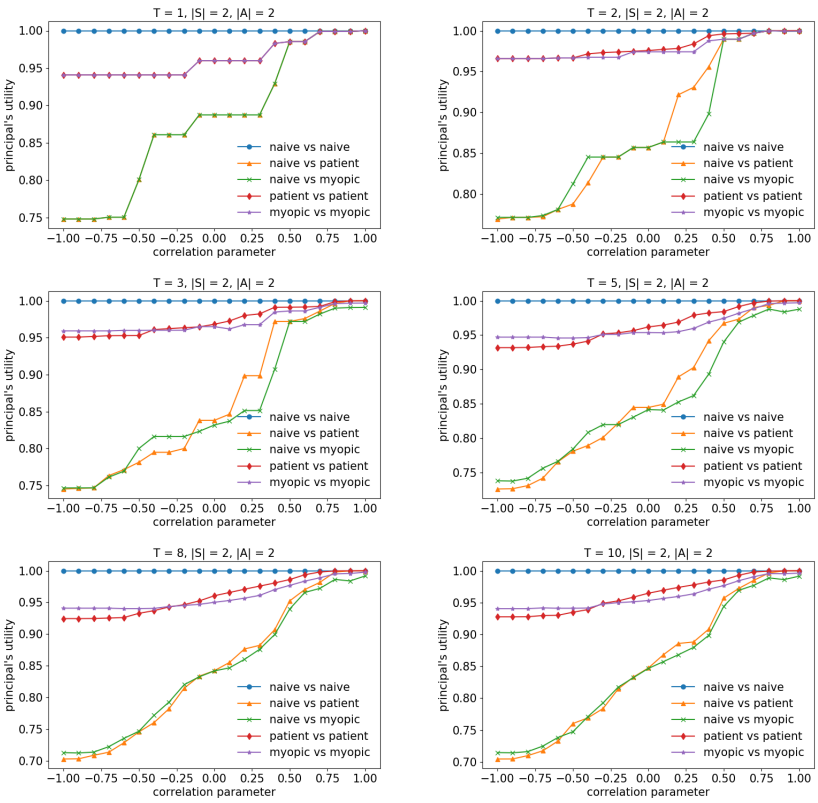

Figure 2: Performance of different mechanisms facing different types of agents when $|\mathcal{S}| = |\mathcal{A}| = 2$ and the time horizon T varies. All numbers are normalized by the optimal utility facing a naïve agent. Every point is an average of 10 independent runs using different random seeds.

strategic agent are much less affected by the correlation parameter. Moreover, as Figure 2 shows, the performance of optimal mechanisms facing a strategic agent is remarkably stable as the time horizon grows. In other words, in random dynamic environments, the utility loss caused by competing interests of the principal and the agent is only mildly amplified by long time horizons.

Difference between patient and myopic agents. As can be seen from the figures, regardless of whether the agent is patient or myopic, the principal's optimal utility is almost the same. Nevertheless, the difference appears to be amplified as the time horizon grows (see Figure 2). When the correlation parameter $\eta = -1$, the optimal utility facing a myopic agent is noticeably larger than that facing a patient agent — which makes sense as only the patient agent has interests truly opposite those of the principal. This gap shrinks as η becomes larger, and vanishes when η is around -0.25. Then, as η continues to grow, the optimal utility facing a myopic agent falls behind and never catches up. In particular, when $\eta = 1$, the optimal utility facing a patient agent is the same as the naïve benchmark, whereas that facing a myopic agent is slightly smaller. The above phenomena indicate that in environments with a long time horizon, myopic agents are easier to exploit, while patient agents are easier to cooperate with. Interestingly, the critical value of η, where the optimal utility facing a patient agent catches up, is about -0.25 instead of 0, which suggests that even when the principal's and the agent's valuations are mildly negatively correlated, it is possible to find a middle ground where cooperation is more beneficial than exploitation in the long run.

G Omitted Proofs from Section 3

Proof of Theorem 1. We consider the case where payments are not allowed, i.e., $p_t(h, s) = 0$ for all $h \in \mathcal{H}$ and $s \in \mathcal{S}$. The case with payments and dynamic IR constraints is essentially the same. We use a similar reduction from MAX-SAT to the ones in [45, 40] for partially observable

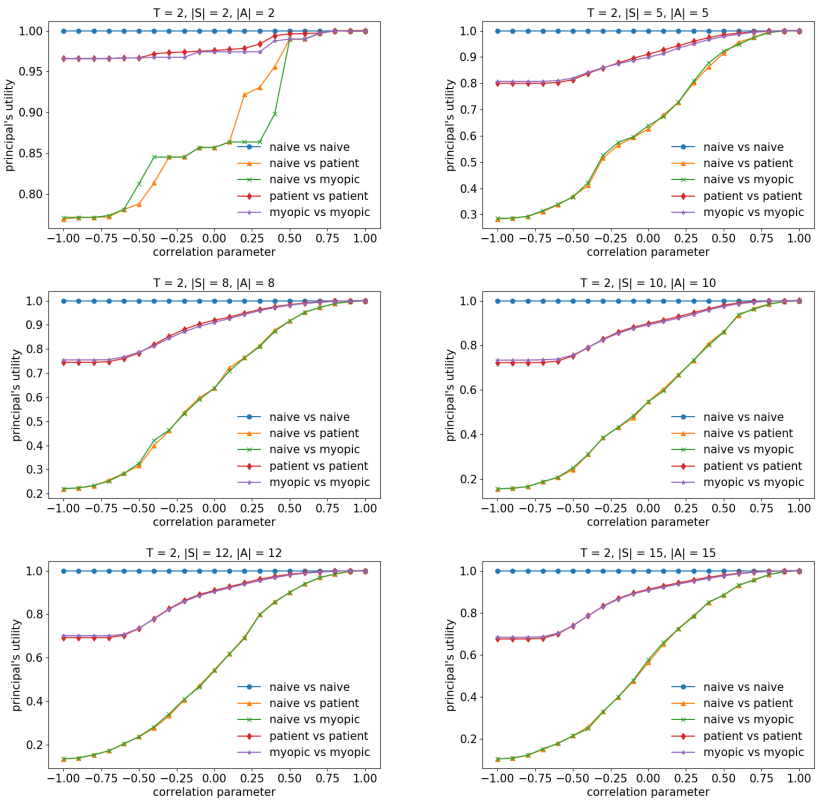

Figure 3: Performance of different mechanisms facing different types of agents when $T = 2$ and the numbers of states and actions, $|\mathcal{S}|$ and $|\mathcal{A}|$, vary. All numbers are normalized by the optimal utility facing a naïve agent. Every point is an average of 10 independent runs using different random seeds.

Markov decision processes (POMDPs). Given a MAX-SAT instance with n variables $x_1, \ldots, x_n$ and m clauses $c_1, \ldots, c_m$ where $c_i = \{\ell_{i,j}\}_{j \in [k_i]}$ and each $\ell_{i,j}$ is a literal, we construct a dynamic environment where $T = n$, $|\mathcal{S}| = m + 1$, and $|\mathcal{A}| = 2$. The goal is to show that the maximum utility is precisely the fraction of clauses that can be simultaneously satisfied. Without loss of generality, we assume that no clause contains both the positive literal and the negative literal of a same variable. We first describe $\mathcal{S}$ and $\mathcal{A}$. Each clause c_i corresponds to a unique state in $\mathcal{S}$, s_i. In addition to these m states, there is another state s_0. $\mathcal{A}$ consists of two actions: a_{pos} and a_{neg}. The transition operator and the principal's valuation function are such that:

- The initial distribution is uniform over $\{s_i\}_{i \in [m]}$, i.e., $P_0(s_i) = 1/m$ for each $i \in [m]$.
- For each $t \in [T]$ and $a \in \mathcal{A}$,

$$P_t(s_0, a, s_0) = 1 \quad \text{and} \quad v_t^P(s_0, a) = 0.$$

  Moreover, for each $t \in [T]$ and $i \in [m]$:

  - If $x_t^+ \in c_i$, then

$$P_t(s_i, a_{\text{pos}}, s_0) = P_t(s_i, a_{\text{neg}}, s_i) = 1,$$

    and

$$v_t^P(s_i, a_{\text{pos}}) = 1 \quad \text{and} \quad v_t^P(s_i, a_{\text{neg}}) = 0.$$

  - if $x_t^- \in c_i$, then

$$P_t(s_i, a_{\text{pos}}, s_i) = P_t(s_i, a_{\text{neg}}, s_0) = 1,$$

    and

$$v_t^P(s_i, a_{\text{pos}}) = 0 \quad \text{and} \quad v_t^P(s_i, a_{\text{neg}}) = 1.$$

– otherwise,

$$P_t(s_i, a_{\text{pos}}, s_i) = P_t(s_i, a_{\text{neg}}, s_i) = 1,$$

and

$$v_t^P(s_i, a_{\text{pos}}) = v_t^P(s_i, a_{\text{neg}}) = 0.$$

- The principal and the agent are in a zero-sum situation, i.e., for any $t \in [T]$, $s \in \mathcal{S}$, $a \in \mathcal{A}$,

$$v_t^A(s, a) = 1 - v_t^P(s, a).$$

Now we show that the maximum utility is precisely the fraction of clauses that can be simultaneously satisfied. First observe that without loss of generality, an optimal mechanism depends only on time (and not on the reported states). This is because of the zero-sum situation: if the mechanism depends on the reports, then the agent can always choose the worst sequence of actions, which can only make the principal's utility smaller. Moreover, given the above observation, without loss of generality, an optimal mechanism is deterministic. This is because the overall utility of the principal is linear in the action at any time t, so one can always round a randomized mechanism into a deterministic one with at least the same overall utility.

Given the above observations, an optimal mechanism corresponds precisely to a way of assigning values to variables in the MAX-SAT instance: for each $t \in [T]$, the action at time t is a_{pos} iff the variable $x_t = 1$ (i.e., the literal x_t^+ is chosen). Moreover, when the initial state is s_i, the onward utility is 1 if the clause c_i is satisfied by the above assignment, and 0 otherwise. Since the initial state is uniformly at random among $\{s_i\}_{i \in [m]}$, the maximum utility is precisely the maximum fraction of clauses that are satisfiable by some assignment. The theorem then follows from the fact that MAX-SAT is hard to approximate within a factor of $7/8 + \varepsilon$ for any $\varepsilon > 0$ [32]. □

Proof of Lemma 1. For brevity, let obj denote the objective, i.e.,

$$\mathrm{obj} = \sum_{h \in \mathcal{H}, s \in \mathcal{S}: (h,s) \text{ is feasible}} \left(\sum_{a \in \mathcal{A}} v_{|h|+1}^P(s, a) \cdot x(h, s, a) + y(h, s) \right).$$

Moreover, for each $h \in \mathcal{H}$, $s \in \mathcal{S}$, let

$$\mathrm{obj}(h, s) = \sum_{h' \in \mathcal{H}, s' \in \mathcal{S}: (h,s) \subseteq (h',s')} \left(\sum_{a \in \mathcal{A}} v_{|h'|+1}^P(s', a) \cdot x(h', s', a) + y(h', s') \right).$$

Observe that

$$\mathrm{obj} = \sum_{s \in \mathcal{S}} \mathrm{obj}(\emptyset, s).$$

We first prove inductively that for each $h \in \mathcal{H}$, $s \in \mathcal{S}$,

$$\mathrm{obj}(h, s) = z(h, s) \cdot u_P^M(h, s).$$

When $|h| = T - 1$, by the definition of feasible extensions and the construction of the mechanism,

$$\begin{aligned}
\mathrm{obj}(h, s) &= \sum_{a \in \mathcal{A}} v_T^P(s, a) \cdot x(h, s, a) + y(h, s) \\
&= z(h, s) \cdot \left(\sum_{a \in \mathcal{A}} v_T^P(s, a) \cdot \pi(h, s, a) + p(h, s) \right) \\
&= z(h, s) \cdot u_P^M(h, s).
\end{aligned}$$

When $|h| < T-1$, for similar reasons,

$$\begin{aligned}
\mathrm{obj}(h,s) &= \sum_{h',s':(h,s)\subseteq(h',s')} \left(\sum_{a\in\mathcal{A}} v^P_{|h'|+1}(s',a)\cdot x(h',s',a) + y(h',s') \right) \\
&= \sum_{a\in\mathcal{A}} v^P_{|h|+1}(s,a)\cdot x(h,s,a) + y(h,s) \\
&\quad + \sum_{h',s':(h,s)\subseteq(h',s'),|h'|>|h|} \left(\sum_{a\in\mathcal{A}} v^P_{|h'|+1}(s',a)\cdot x(h',s',a) + y(h',s') \right) \\
&= z(h,s)\cdot \left(\sum_{a\in\mathcal{A}} v^P_{|h|+1}(s,a)\cdot \pi(h,s,a) + p(h,s) \right) \\
&\quad + \sum_{a',s'':P_{|h|+1}(s,a',s'')>0} \sum_{h',s':(h+(s,a'),s'')\subseteq(h',s')} \left(\sum_{a\in\mathcal{A}} v^P_{|h'|+1}(s',a)\cdot x(h',s',a) + y(h',s') \right)
\end{aligned}$$

By the induction hypothesis, the second sum above is equal to

$$\begin{aligned}
&\sum_{a',s'':P_{|h|+1}(s,a',s'')>0} \mathrm{obj}(h+(s,a'),s'') \\
&= \sum_{a',s'':P_{|h|+1}(s,a',s'')>0} z(h+(s,a'),s'')\cdot u^M_P(h+(s,a'),s'') \\
&= \sum_{a',s'':P_{|h|+1}(s,a',s'')>0} x(h,s,a')\cdot P^E_{|h|+1}(s,a',s'')\cdot u^M_P(h+(s,a'),s'') \\
&= \sum_{a\in\mathcal{A},s'\in\mathcal{S}} x(h,s,a)\cdot P_{|h|+1}(s,a,s')\cdot u^M_P(h+(s,a),s') \\
&= z(h,s)\cdot \sum_{a\in\mathcal{A}} \left(\pi(h,s,a)\cdot \sum_{s'\in\mathcal{S}} P_{|h|+1}(s,a,s')\cdot u^M_P(h+(s,a),s') \right).
\end{aligned}$$

Putting this back into the above expression for $\mathrm{obj}(h,s)$, we get

$$\begin{aligned}
&\mathrm{obj}(h,s) \\
&= z(h,s)\cdot \left(\sum_{a\in\mathcal{A}} v^P_{|h|+1}(s,a)\cdot \pi(h,s,a) + p(h,s) \right) \\
&\quad + z(h,s)\cdot \sum_{a\in\mathcal{A}} \left(\pi(h,s,a)\cdot \sum_{s'\in\mathcal{S}} P_{|h|+1}(s,a,s')\cdot u^M_P(h+(s,a),s') \right) \\
&= z(h,s)\cdot \left(\sum_{a\in\mathcal{A}} \cdot \pi_{|h|+1}(h,s,a)\cdot \left(v^P_{|h|+1}(s,a) + \sum_{s'\in\mathcal{S}} P_{|h|+1}(s,a,s')\cdot u^M_P(h+(s,a),s') \right) + p(h,s) \right) \\
&= z(h,s)\cdot u^M_P(h,s).
\end{aligned}$$

So for any $h\in\mathcal{H}$, $s\in\mathcal{S}$, $\mathrm{obj}(h,s) = z(h,s)\cdot u^M_P(h,s)$. Then we immediately have

$$u^M_P(\emptyset) = \sum_{s\in\mathcal{S}} P_0(s)\cdot u^M_P(\emptyset,s) = \sum_{s\in\mathcal{S}} z(\emptyset,s)\cdot u^M_P(\emptyset,s) = \sum_{s\in\mathcal{S}} \mathrm{obj}(\emptyset,s) = \mathrm{obj}. \qquad \square$$

Proof of Lemma 3. By Eq. (4) and Lemma 2, for all h, s, s',

$$
\begin{aligned}
& u(h,s,s') \\
&= \sum_{a\in\mathcal{A}} v^A_{|h|+1}(s,a)\cdot x(h,s',a) - y(h,s') \\
&\quad + \sum_{a\in\mathcal{A},s''\in\mathcal{S}} \frac{P_{|h|+1}(s,a,s'')}{P^E_{|h|+1}(s',a,s'')}\cdot z(h+(s',a),s'')\cdot u^M_A(h+(s',a),s'') && \text{(Lemma 2)} \\
&= \sum_{a\in\mathcal{A}} v^A_{|h|+1}(s,a)\cdot x(h,s',a) - y(h,s') + \sum_{a\in\mathcal{A},s''\in\mathcal{S}} P_{|h|+1}(s,a,s'')\cdot x(h,s',a)\cdot u^M_A(h+(s',a),s'') \\
& && \text{(Eq. (4))}
\end{aligned}
$$

Now by rearranging the above expression and applying the construction of the mechanism M and the single-step reporting strategy $r_{h,s,s'}$, we have

$$
\begin{aligned}
& u(h,s,s') \\
&= \sum_{a\in\mathcal{A}} x(h,s',a)\left(v^A_{|h|+1}(s,a) + \sum_{s''\in\mathcal{S}} P_{|h|+1}(s,a,s'')\cdot u^M_A(h+(s',a),s'')\right) - y(h,s') \\
& && \text{(rearranging)} \\
&= z(h,s')\cdot\left(\sum_a \pi(h,s',a)\cdot\left(v^A_{|h|+1}(s,a) + \sum_{s''} P_{|h|+1}(s,a,s'')\cdot u^M_A(h+(s',a),s'')\right) - p(h,s')\right) \\
& && \text{(construction of mechanism)} \\
&= z(h,s')\cdot\left(\sum_a \pi(h,s',a)\cdot\left(v^A_{|h|+1}(s,a) + \sum_{s''} P_{|h|+1}(s,a,s'')\cdot u_A^{M,r_{h,s,s'}}(h+(s',a),s'')\right) - p(h,s')\right) \\
& && \text{(construction of } r_{h,s,s'}\text{)} \\
&= z(h,s')\cdot u_A^{M,r_{h,s,s'}}(h,s), && \text{(definition of } u_A^{M,r_{h,s,s'}}\text{)}
\end{aligned}
$$

as desired. □

Proof of Lemma 4. Fix $h\in\mathcal{H}$, $s,s'\in\mathcal{S}$, and let $(s_p,a_p)=\text{last}(h)$. When $h=\emptyset$, by Lemmas 2 and 3 and Eq. (3),

$$
\begin{aligned}
& u(h,s) \geq \frac{P^E_{|h|}(s_p,a_p,s)}{P^E_{|h|}(s_p,a_p,s')}\cdot u(h,s,s') \\
\iff\ & z(\emptyset,s)\cdot u^M_A(\emptyset,s) \geq \frac{P^E_0(s_p,a_p,s)}{P^E_0(s_p,a_p,s')}\cdot z(\emptyset,s')\cdot u_A^{M,r_{\emptyset,s,s'}}(\emptyset,s) \\
\iff\ & z(\emptyset,s)\cdot u^M_A(\emptyset,s) \geq \frac{P^E_0(s)}{P^E_0(s')}\cdot z(\emptyset,s')\cdot u_A^{M,r_{\emptyset,s,s'}}(\emptyset,s) \\
\iff\ & u^M_A(\emptyset,s) \geq u_A^{M,r_{\emptyset,s,s'}}(\emptyset,s).
\end{aligned}
$$

When $|h|>0$, suppose $h=(s_1,a_1,\ldots,s_t,a_t)$, and let $h_p=(s_1,a_1,\ldots,s_{t-1},a_{t-1})$. By Lemmas 2 and 3 and Eq. (2),

$$
\begin{aligned}
& u(h,s) \geq \frac{P^E_{|h|}(s_p,a_p,s)}{P^E_{|h|}(s_p,a_p,s')}\cdot u(h,s,s') \\
\iff\ & z(h,s)\cdot u^M_A(h,s) \geq \frac{P^E_{|h|}(s_p,a_p,s)}{P^E_{|h|}(s_p,a_p,s')}\cdot z(h,s')\cdot u_A^{M,r_{h,s,s'}}(h,s) \\
\iff\ & x(h_p,s_p,a_p)\cdot u^M_A(h,s) \geq x(h_p,s_p,a_p)\cdot u_A^{M,r_{h,s,s'}}(h,s).
\end{aligned}
$$

Note that when $x(h_p,s_p,a_p)=0$, (h,s) cannot be reached, because (1) if $z(h_p,s_p)>0$, then when the (reported) history-state pair is (h_p,s_p), the mechanism never takes action a_p, and (2) if $z(h_p,s_p)=0$, then such an impossible action exists somewhere in h_p. In such cases, $\pi(h,s)$ and

$p(h, s)$ will never be accessed, since it is impossible for the (reported) history to be h. In other words, when (h, s) is reachable, we must have $x(h_p, s_p, a_p) > 0$, in which case the last inequality is equivalent to $u_A^M(h, s) \geq u_A^{M, r_{h,s,s'}}(h, s)$. □

Proof of Lemma 5. We only need to show that IC is equivalent to robustness against single-step misreporting. We prove this inductively, aiming to eliminate misreporting one step at a time. To be more specific, consider the following partial reporting strategy. For a reporting strategy r, $t \in [T]$, let $r|_{\geq t}$ denote the reporting strategy restricted to time $t, t+1, \ldots, T$, i.e., for any $h' \in \mathcal{H}$, $s' \in \mathcal{S}$,

$$r|_{\geq t}(h', s') = \begin{cases} s', \text{if } |h'| + 1 < t \\ r(h', s'), \text{otherwise} \end{cases}.$$

Similarly, let $r|_{<t}$ denote r restricted to time $1, 2, \ldots, t-1$, and $r|_{=t}$ denote r restricted to time t. We show inductively that for any reachable history-state pair (h, s), and any reporting strategy r,

$$u_A^{M,(r|_{<|h|+1})}(h, s) \geq u_A^{M,r}(h, s).$$

Without loss of generality, we assume that for any unreachable pair (h', s'), r simply reports truthfully, i.e., $r(h', s') = s'$.

Recall that $r(h)$ is the reported history given by r when the true history is h. When $|h| = T - 1$, the above claim is implied by Lemma 4, because

$$u_A^{M,r}(h, s) = u_A^{M,(r|_{\geq T})}(r(h), s) \geq u_A^M(r(h), s) = u_A^{M,(r|_{<T})}(h, s).$$

Now suppose $|h| < T - 1$. By the induction hypothesis, we have

$$u_A^{M,r}(h, s) = u_A^{M,(r|_{\geq |h|+1})}(r(h), s) \leq u_A^{M,((r|_{\geq |h|+1})|_{<|h|+2})}(r(h), s) = u_A^{M,(r|_{=|h|+1})}(r(h), s).$$

Now again by Lemma 4, we have

$$u_A^{M,r}(h, s) \leq u_A^{M,(r|_{=|h|+1})}(r(h), s) \leq u_A^M(r(h), s) = u_A^{M,(r|_{<|h|+1})}(h, s),$$

which establishes the above claim.

Now observe that as a special case of the claim, for any $s \in \mathcal{S}$,

$$u_A^{M,r}(\emptyset, s) \leq u_A^{M,(r|_{<1})}(\emptyset, s) = u_A^M(\emptyset, s).$$

Now summing over s, this implies that for any reporting strategy r,

$$u_A^{M,r}(\emptyset) = \sum_{s \in \mathcal{S}} P_0(s) \cdot u_A^{M,r}(\emptyset, s) \leq \sum_{s \in \mathcal{S}} P_0(s) \cdot u_A^M(\emptyset, s) = u_A^M(\emptyset).$$ □

Proof of Theorem 2. Given the correspondence between mechanisms and LP variables, by Lemma 5, it is easy to see that (modulo the unreachable parts) every IC and IR mechanism corresponds bijectively to a feasible solution to the LP in Figure 1. Moreover, by Lemma 1, the objective value of this solution is precisely the principal's overall utility, which directly implies that an optimal solution to the LP corresponds to an IC and IR mechanism which maximizes the principal's overall utility.

Now observe that the number of variables and the number of constraints in the LP are both $O(|\mathcal{S}|^{T+1}|\mathcal{A}|^T)$. Moreover, all relevant coefficients in the LP can be encoded using $O(L)$ bits. It is well-known that such an LP can be solved in time $\text{poly}(|\mathcal{S}|^T, |\mathcal{A}|^T, L)$. □

H Omitted Proofs from Section D

Proof of Lemma 7. We construct M' explicitly based on M. Let $\pi'(t, s_p, a_p, s, a)$ be the probability that M' chooses action a at time t in state s when the previous state-action pair is (s_p, a_p). Similarly, let $p'(t, s_p, a_p, s)$ be the payment specified by M' at time t in state s when the previous state-action

pair is (s_p, a_p). We construct M' from M inductively as follows. For each $t \in [T]$, $s_p \in \mathcal{S}$ and $a_p \in \mathcal{A}$, let $h^*(t, s_p, a_p) \in \mathcal{H}_{t-1}$ be any history such that

$$h^*(t, s_p, a_p) \in \operatorname*{argmax}_{h \in \mathcal{H}_{t-1}:(s_p,a_p)=\text{last}(h)} \sum_{s \in \mathcal{S}} P_{t-1}(s_p, a_p, s) \cdot \Bigg(p(h, s) + \sum_{a \in \mathcal{A}} \pi(h, s, a) \cdot \Bigg(v^P_{|h|+1}(s, a) + \sum_{s' \in \mathcal{S}} P_t(s, a, s') \cdot u^M_P(h + (s, a), s') \Bigg) \Bigg).$$

Then, for all $s \in \mathcal{S}$, let

$$\pi'(t, s_p, a_p, s) = \pi(h^*(t, s_p, a_p), s) \quad \text{and} \quad p(t, s_p, a_p, s) = p(h^*(t, s_p, a_p), s).$$

This finishes the construction of M'.

We first show that $u^{M'}_P(\emptyset) \geq u^M_P(\emptyset)$, by inductively showing a stronger claim: for all $h \in \mathcal{H}$,

$$\sum_s P_{|h|}(s_p, a_p, s) \cdot u^{M'}_P(h, s) \geq \sum_s P_{|h|}(s_p, a_p, s) \cdot u^M_P(h, s),$$

where $(s_p, a_p) = \text{last}(h)$. For all $h \in \mathcal{H}_{T-1}$, letting $(s_p, a_p) = \text{last}(h)$, by the construction of M', we have

$$\begin{aligned} \sum_s P_{T-1}(s_p, a_p, s) \cdot u^{M'}_P(h, s) &= \sum_s P_{T-1}(s_p, a_p, s) \cdot u^M_P(h^*(T, s_p, a_p), s) \\ &\geq \sum_s P_{T-1}(s_p, a_p, s) \cdot u^M_P(h, s). \end{aligned}$$

Now for all $h \in \mathcal{H}$ where $|h| < T - 1$, letting $(s_p, a_p) = \text{last}(h)$ and $h^* = h^*(|h| + 1, s_p, a_p)$, we have

$$\begin{aligned} &\sum_s P_{|h|}(s_p, a_p, s) \cdot u^{M'}_P(h, s) \\ &= \sum_s P_{|h|}(s_p, a_p, s) \cdot \left(p(h^*, s) + \sum_{a \in \mathcal{A}} \pi(h^*, s, a) \cdot \left(v^P_{|h|+1}(s, a) + \sum_{s' \in \mathcal{S}} P_t(s, a, s') \cdot u^{M'}_P(h + (s, a), s') \right) \right) \\ &= \sum_s P_{|h|}(s_p, a_p, s) \cdot \left(p(h^*, s) + \sum_{a \in \mathcal{A}} \pi(h^*, s, a) \cdot \left(v^P_{|h|+1}(s, a) + \sum_{s' \in \mathcal{S}} P_t(s, a, s') \cdot u^{M'}_P(h^* + (s, a), s') \right) \right) \\ &\qquad \text{(property of } M') \\ &\geq \sum_s P_{|h|}(s_p, a_p, s) \cdot \left(p(h^*, s) + \sum_{a \in \mathcal{A}} \pi(h^*, s, a) \cdot \left(v^P_{|h|+1}(s, a) + \sum_{s' \in \mathcal{S}} P_t(s, a, s') \cdot u^M_P(h^* + (s, a), s') \right) \right) \\ &\qquad \text{(induction hypothesis)} \\ &\geq \sum_s P_{|h|}(s_p, a_p, s) \cdot \left(p(h, s) + \sum_{a \in \mathcal{A}} \pi(h, s, a) \cdot \left(v^P_{|h|+1}(s, a) + \sum_{s' \in \mathcal{S}} P_t(s, a, s') \cdot u^M_P(h + (s, a), s') \right) \right) \\ &\qquad \text{(choice of } h^*) \\ &= \sum_s P_{|h|}(s_p, a_p, s) \cdot u^M_P(h, s). \end{aligned}$$

Then in particular, we have

$$u^{M'}_P(\emptyset) = \sum_s P_0(s) \cdot u^{M'}_P(\emptyset, s) \geq \sum_s P_0(s) \cdot u^M_P(\emptyset, s) = u^M_P(\emptyset).$$

Finally we prove that M' is IC. By the proof of Lemma 5, we only need to show that M' is robust against any single-step reporting strategy $r_{h,s,s'}$. In fact, letting $(s_p, a_p) = \text{last}(h)$ and $h^* = h^*(|h| + 1, s_p, a_p)$,

$$u^{M'}_A(h, s) = \sum_a \pi(h^*, s, a) \cdot v^A_{|h|+1}(s, a) + p(h^*, s) = u^M_A(h^*, s).$$

Moreover,

$$u_A^{M',r_{h,s,s'}}(h,s) = \sum_a \pi(h^*,s',a)\cdot v^A_{|h|+1}(s,a) + p(h^*,s) = u_A^{M,r_{h,s,s'}}(h^*,s).$$

Since M is IC, we have

$$u_A^{M'}(h,s) = u_A^M(h^*,s) \geq u_A^{M,r_{h,s,s'}}(h^*,s) = u_A^{M',r_{h,s,s'}}(h,s).$$

Now by the argument in the proof of Lemma 5, we know that for all reporting strategy r, $h \in \mathcal{H}$, $s \in \mathcal{S}$,

$$u_A^{M',r}(h,s) \leq u_A^{M',(r|_{<|h|+1})}(h,s),$$

so

$$u_A^{M',(r|_{\geq|h|+1})}(h,s) \leq u_A^{M',((r|_{\geq|h|+1})|_{<|h|+1})}(h,s) = u_A^{M'}(h,s),$$

which is precisely the IC requirement for myopic agents. Similar arguments guarantee that M' has the same IR property as M. □

Proof of Theorem 4. We first argue the easy part, i.e., the time complexity. Observe that calls to OptStatMech dominates the time complexity. Moreover, the algorithm makes $T|\mathcal{S}||\mathcal{A}|$ calls to OptStatMech, so the overall time complexity is as stated.

Now we show the optimality of the computed mechanism M. We prove inductively a stronger claim, i.e., for any $t \in [T]$, $s_p \in \mathcal{S}$, $a_p \in \mathcal{A}$,

$$\sum_s P_0(s_p,a_p,s)\cdot u_P^M(t,s_p,a_p,s) = \max_{M'} P_0(s_p,a_p,s)\cdot u_P^{M'}(t,s_p,a_p,s),$$

where the maximum is over all succinct mechanisms M' that are IC and (optionally) IR. First observe that for all $s \in \mathcal{S}$, $a \in \mathcal{A}$,

$$u(T,s,a) = v_T^P(s,a).$$

So, for all $s_p \in \mathcal{S}$, $a_p \in \mathcal{A}$,

$$\begin{aligned}
&\sum_s P_0(s_p,a_p,s)\cdot u_P^M(T,s_p,a_p,s)\\
&= \sum_s P_0(s_p,a_p,s)\cdot\left(p(T,s_p,a_p,s) + \sum_a \pi(T,s_p,a_p,s,a)\cdot v_T^P(s,a)\right)\\
&= \max_{M'=(\pi',p')} \sum_s P_0(s_p,a_p,s)\cdot\left(p'(T,s_p,a_p,s) + \sum_a \pi'(T,s_p,a_p,s,a)\cdot v_T^P(s,a)\right)\\
&\qquad\qquad \text{(optimality of } M \text{ at time } T \text{ as a static mechanism)}\\
&= \max_{M'} \sum_s P_0(s_p,a_p,s)\cdot u_P^M(T,s_p,a_p,s).
\end{aligned}$$

Again, the maximum is over all succinct mechanisms M' that are IC and (optionally) IR.

Now for $t \in [T-1]$, by the construction of M,

$$\begin{aligned}
&\sum_s P_0(s_p,a_p,s)\cdot u_P^M(t,s_p,a_p,s)\\
&= \sum_s P_0(s_p,a_p,s)\cdot\left(p(t,s_p,a_p,s) + \sum_a \pi(t,s_p,a_p,s,a)\cdot\left(v_t^P(s,a)\right.\right.\\
&\quad \left.\left. + \sum_{s'} P_t(s,a,s')\cdot u_P^M(t+1,s,a,s')\right)\right)\\
&= \max_{M'=(\pi',p')} \sum_s P_0(s_p,a_p,s)\cdot\left(p'(t,s_p,a_p,s) + \sum_a \pi'(t,s_p,a_p,s,a)\cdot\left(v_t^P(s,a)\right.\right.\\
&\quad \left.\left. + \sum_{s'} P_t(s,a,s')\cdot u_P^M(t+1,s,a,s')\right)\right). \quad \text{(optimality of } M \text{ at time } t \text{ as a static mechanism)}
\end{aligned}$$

By the induction hypothesis and the fact that M' is succinct,

$$\begin{aligned}
&\sum_s P_0(s_p, a_p, s) \cdot u_P^M(t, s_p, a_p, s) \\
&= \max_{M'=(\pi',p')} \sum_s P_0(s_p, a_p, s) \cdot \left(p'(t, s_p, a_p, s) + \sum_a \pi'(t, s_p, a_p, s, a) \cdot \left(v_t^P(s, a) \right.\right. \\
&\quad \left.\left. + \max_{M''} \sum_{s'} P_t(s, a, s') \cdot u_P^{M''}(t+1, s, a, s') \right) \right) && \text{(induction hypothesis)} \\
&= \max_{M'=(\pi',p')} \sum_s P_0(s_p, a_p, s) \cdot \left(p'(t, s_p, a_p, s) + \sum_a \pi'(t, s_p, a_p, s, a) \cdot \left(v_t^P(s, a) \right.\right. \\
&\quad \left.\left. + \sum_{s'} P_t(s, a, s') \cdot u_P^{M'}(t+1, s, a, s') \right) \right) && (M' \text{ is succinct}) \\
&= \max_{M'} \sum_s P_0(s_p, a_p, s) \cdot u_P^{M'}(t, s_p, a_p, s).
\end{aligned}$$

All maxima are over all succinct mechanisms that are IC and (optionally) IR. As a result, we have

$$u_P^M(\emptyset) = \sum_s P_0(s) \cdot u_P^M(\emptyset, s) = \max_{M'} \sum_s P_0(s) \cdot u_P^{M'}(\emptyset, s) = \max_{M'} u_P^{M'}(\emptyset). \qquad \square$$

I Omitted Proofs from Section E

Proof of Theorem 5. First suppose the agent is patient and without loss of generality has a discount factor of 1. Let $T = 2$ and $\mathcal{S} = \mathcal{A} = [n]$ where $n \geq \varepsilon^{-1}$. The initial distribution is uniform over $[n]$, i.e., $P_0(i) = 1/n$ for all $i \in [n]$, i.e., no matter what action is played, all states always transition to state 1. The transition operator is such that $P_1(i, j, 1) = 1$ for all $i, j \in [n]$. At time $T = 2$, the principal's valuations are $v_T^P(i, j) = 0$ for all $i, j \in [n]$. At time 1, the principal's valuation function is such that for all $i, j \in [n]$, $v_1^P(i, j) = 1$ if $i = j$, and $v_1^P(i, j) = 0$ if $i \neq j$. For $t \in [T]$, the agent's valuation function is such that for all $i, j \in [n]$, $v_t^A(i, j) = 0$ if $i = j$, and $v_t^A(i, j) = 1$ if $i \neq j$.

Consider the principal's optimal utility, which is clearly upper bounded by 1 (1 at time 1 and 0 at time 2). The following mechanism is IC and achieves this upper bound:

- At time 1, play action i for each state $i \in [n]$.
- At time $T = 2$, play action $(i \bmod n) + 1$ iff the state at time 1 is i.

The mechanism is IC because regardless of the (reported) initial state, the agent achieves overall utility 1. It is easy to check this mechanism achieves utility 1.

On the other hand, any memoryless IC mechanism can achieve utility at most $1/n \leq \varepsilon$. This is because at time $T = 2$, the current state provides absolutely no information, so the mechanism has to perform the same (randomized) action regardless of the initial state. As a result, in order to be IC, the mechanism has to satisfy the following condition at time 1: for all $i, j \in [n]$, $\pi(i, i) \leq \pi(j, i)$, where $\pi(a, b)$ is the probability that action b is played in state a at time 1. So the principal's utility can be bounded as follows:

$$\frac{1}{n} \sum_i \pi(i, i) \leq \frac{1}{n} \sum_i \left(\frac{1}{n} \sum_j \pi(j, i) \right) = \frac{1}{n^2} \sum_{i,j} \pi(j, i) = \frac{1}{n}.$$

This concludes the proof when the agent is patient.

Now consider the case with a myopic agent. Again, let $T = 2$ and $\mathcal{S} = \mathcal{A} = [n]$ where $n \geq \varepsilon^{-1}$. The initial distribution is again uniform over $[n]$, i.e., $P_0(i) = 1/n$ for all $i \in [n]$. The transition operator is such that $P_1(i, j, i) = 1$ for all $i, j \in [n]$, i.e., no matter what action is played, state i always transitions to state i. At time 1, the principal's and the agent's valuations are $v_1^P(i, j) = v_1^A(i, j) = 0$

for all $i, j \in [n]$. At time $T = 2$, the principal's valuation function is such that for all $i, j \in [n]$, $v_T^P(i, j) = 1$ if $i = j$, and $v_T^P(i, j) = 0$ if $i \neq j$. And the agent's valuation function is such that for all $i, j \in [n]$, $v_T^A(i, j) = 0$ if $i = j$, and $v_T^A(i, j) = 1$ if $i \neq j$.

The principal's optimal utility, 1, is achieved by the following succinct (but not memoryless) IC mechanism:

- At time 1, play action 1 for all states.
- At time 2, play action i iff the state at time 1 is i.

The mechanism is IC in particular because the agent is myopic and cannot change the past. It is easy to check this mechanism achieves utility 1.

On the other hand, any memoryless IC mechanism can achieve utility at most $1/n \leq \varepsilon$. This is because at time $T = 2$, the mechanism cannot memorize anything before, so it has to be IC based only on the current state, which puts the mechanism in a situation that is essentially the same as at time 1 in the hard instance for patient agents. Similar arguments then guarantee that the principal's utility is at most $1/n$, which concludes the proof for myopic agents. Finally, we note that the above constructions work even if payments are allowed. □

Proof of Theorem 6. We use reductions from MAX-SAT similar to that in Theorem 1 for both myopic and patient agents. First consider the case where the agent is patient with a discount factor of 1. In this case, the reduction in Theorem 1 applies without any modification. In particular, since the principal and the agent are in a zero-sum situation, without loss of generality, any optimal memoryless mechanism does not depend on the reported states. And again, since the principal's utility is multilinear in the actions at each time, there is a deterministic mechanism which is optimal. As argued in the proof of Theorem 1, such a mechanism corresponds precisely to an optimal assignment of variables in the MAX-SAT instance, which implies the $7/8 + \varepsilon$ inapproximability.

Now consider the case where the agent is myopic. Here we slightly modify the reduction, and in particular, the agent's valuation functions. That is, for each $t \in [T]$ and $i \in [m]$, we let

$$v_t^A(s, a_{\text{pos}}) = c \quad \text{and} \quad v_t^A(s, a_{\text{neg}}) = 0,$$

for all $s \in \mathcal{S}$, where $c > 0$ is an arbitrarily small constant. This guarantees that at any time t, in order to be IC, the (randomized) actions for all states have to be exactly the same. Then since the principal's utility is multilinear, again it is without loss of generality to consider deterministic mechanisms, which correspond to assignments of variables. The ratio of $7/8 + \varepsilon$ follows immediately. Finally, we remark that the above reductions still work when payments are allowed. □