

On the Complexity Gap Between POMDPs and Multi-Environment POMDPs

Eline M. Bovy*
Radboud University
Nijmegen, The Netherlands

Caleb Probine*
The University of Texas at Austin
Austin, USA

Marnix Suilen
University of Antwerp
Antwerp, Belgium

Ufuk Topcu
The University of Texas at Austin
Austin, USA

Nils Jansen
Ruhr-University Bochum
Bochum, Germany
Radboud University
Nijmegen, The Netherlands

Abstract

Multi-environment POMDPs (ME-POMDPs) generalize POMDPs by encoding a finite set of possible environments, capturing model uncertainty. The finite-horizon robust reward maximization problem for a ME-POMDP asks for a policy that maximizes reward in the worst environment, and recent work has shown this problem to be PSPACE-complete. We expand the complexity landscape of ME-POMDPs by identifying which subclasses retain this PSPACE-completeness. In particular, we take *subclasses* of POMDPs that are tractable in the single-environment case, and investigate the computational complexity of these classes when extended to the multi-environment setting. We first prove the NP-completeness of multi-environment *blind* MDPs and *deterministic* POMDPs using a certain closure-under-mixture condition. We then prove PSPACE-completeness for a range of subclasses that do not satisfy that closure condition, yielding new complexity results for multi-observation POMDPs. Finally, we show *existential theory of the reals* ($\exists\mathbb{R}$)-completeness for numerous ME-POMDP problems under time-dependent Markovian policies or finite-state controllers that are NP-complete for standard POMDPs. These results reveal a subtle relationship between POMDP and ME-POMDP complexity that depends heavily on the subclasses involved.

Keywords

POMDPs, Multi-environment, Robustness

1 Introduction

Partially observable Markov decision processes (POMDPs) are a canonical model for sequential decision-making problems under partial information [32]. POMDPs capture problems in which a decision-maker must reason about an unknown state of the world, with applications ranging from robotics to healthcare [29, 37].

POMDP planning algorithms, however, rely on the restrictive assumption that a single exact model of the environment exists. Multi-environment POMDPs (ME-POMDPs) remove this assumption and instead assume a finite set of possible environment models [15]. A ME-POMDP comprises a discrete set of possible environments with shared state, observation, and action spaces. The robust reward-maximization problem for ME-POMDPs then asks for a policy that maximizes

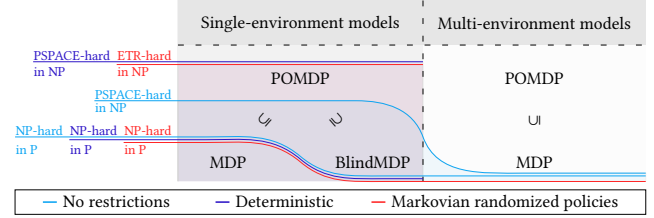


Figure 1: Known complexity results for finite-horizon ME-POMDPs. Dark blue and red lines correspond to problems with deterministic dynamics and Markovian policies, respectively, while bright blue lines correspond to problems with no dynamics or policy restrictions. An annotation “X-hard” atop a line encodes that problems above that line are hard for the complexity class X. An annotation “in X” below a line encodes that problems below the line are decidable in X.

expected reward in the *worst-case* environment, chosen by an adversary. Like POMDPs, ME-POMDPs admit value-iteration-based [15] and gradient-ascent-based [25] algorithms for reward maximization, as well as reinforcement learning algorithms for reach-reward objectives [30]. Meanwhile, in terms of complexity, finite-horizon reward maximization is PSPACE-complete in ME-POMDPs [16], just as in standard POMDPs [46], while the discounted infinite-horizon setting is already undecidable for the latter [41].

Significant effort has been poured into finding more tractable subclasses of POMDPs. These classes rely on additional assumptions to reduce the computational complexity of finite-horizon reward maximization or to achieve decidability for many objectives, including infinite-horizon discounted rewards and logic objectives such as reachability and parity [4, 8]. For finite-horizon rewards on single-environment models, only complexity results for general POMDPs, fully observable MDPs, and blind MDPs are known [46]. In the multi-environment setting, results are only known for multi-environment MDPs [49] and general ME-POMDPs [16], as Figure 1 illustrates, leaving a large gap between single-environment subclasses of POMDPs and their extension to multiple environments.

Contributions

We investigate the complexity of finite-horizon reward maximization in multi-environment POMDPs across a large set of subclasses.

*Shared first authorship, ordered alphabetically.

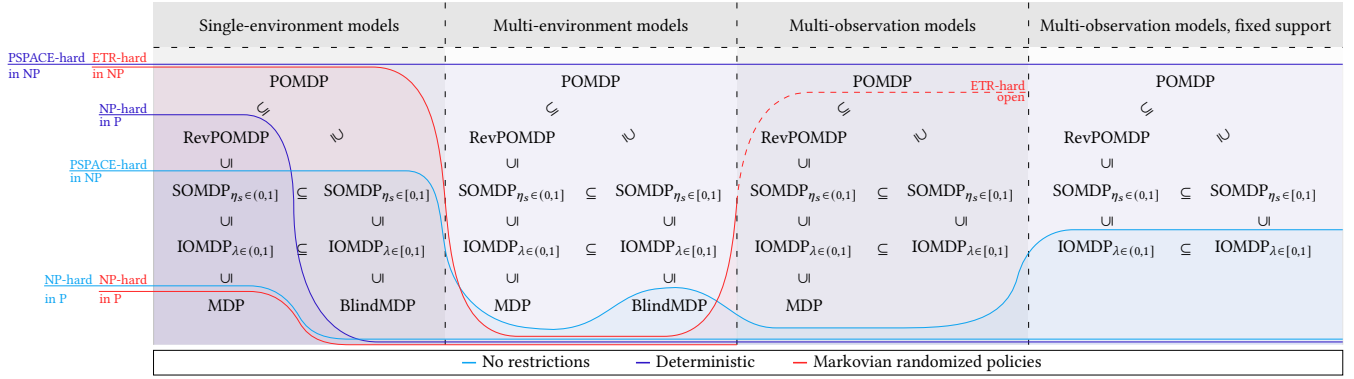


Figure 2: Overview of ME-POMDP complexity including results we present. Dark blue and red lines correspond to problems with deterministic dynamics and Markovian policies, respectively, while bright blue lines correspond to problems with no dynamics or policy restrictions. An annotation “X-hard” atop a line encodes that problems above that line are hard for the complexity class X. An annotation “in X” below a line encodes that problems below the line are decidable in X.

We consider these subclasses in full generality without further restrictions, and also examine special cases with deterministic transition and observation dynamics, as well as restricted policy classes such as finite-memory policies. Our results establish a significantly expanded complexity landscape for ME-POMDPs in Figure 2 compared with Figure 1. Our specific contributions are as follows.

- (1) We show that POMDPs where all states are hidden through the same observation (Blind MDPs) and POMDPs with deterministic transition and observation functions (deterministic POMDPs) remain NP-complete when extended to the multi-environment setting. These results rely on both subclasses respecting a certain *closure under mixture* condition: Allowing the adversary to randomly choose an environment from a fixed set of POMDPs in one of these subclasses is equivalent to the agent acting in a new POMDP in that subclass.
- (2) We analyze a range of multi-environment models that lack the closure property above, yielding hardness results for a range of *multi-observation* POMDP problems. We show that simple observation uncertainty, in particular uncertainty about state labelings or the probability that the true state is revealed, leads to PSPACE-completeness, as in general ME-POMDPs. We also obtain novel complexity results for many single-environment POMDP subclasses.
- (3) We characterize the complexity of ME-POMDPs with restricted policies. We show that numerous problems involving time-dependent Markovian policies or finite-state controllers that are NP-complete for POMDPs become *existential theory of the reals* ($\exists\mathbb{R}$)-complete for ME-POMDPs.

Outline. After related work and preliminaries, Section 4 details all involved subclasses. Section 5 presents the complexity results for ME-POMDPs using hidden-mixture reasoning. Hardness results for ME-POMDPs built from subclasses that are not closed under hidden mixtures are in Section 6. Results on restricted policy classes for ME-POMDPs are in Section 7.

2 Related Work

A range of works provide algorithmic and complexity results for Multi-Environment MDPs (ME-MDPs). In particular, for the robust setting, in which we assume no distribution on the environments, previous work characterizes the complexity of various qualitative problems [13, 20, 49, 53, 54], and provides algorithmic approaches for the reward-maximization problem [3, 52]. For finite-horizon reward maximization, the problem is PSPACE-complete for history-dependent policies [16, 38] and NP-hard when policies are restricted to be time-dependent Markovian [38, 39]. While we study ME-POMDPs, which strictly generalize ME-MDPs, we also tighten existing ME-MDP results by showing that the reward-maximization problem with time-dependent Markovian policies is $\exists\mathbb{R}$ -complete, where $\exists\mathbb{R}$ lies between NP and PSPACE. Single-environment POMDP complexity results provide trivial lower bounds on ME-POMDP complexity. The PSPACE-hardness of POMDPs [46], along with the NP-hardness of subclasses such as Blind MDPs [46] and deterministic POMDPs [40], thus also holds for ME-POMDPs. The closest related work is [16], which shows that finite-horizon reward maximization in ME-POMDPs is PSPACE-complete, but it considers only general POMDPs, not subclasses.

The interaction between the agent and adversary in a ME-POMDP can be encoded as a one-sided zero-sum partially observable stochastic game (POSG) [15]. As ME-POMDPs have a highly restricted adversary structure, complexity lower bounds for POSGs do not immediately transfer to ME-POMDPs. While upper bounds for POSGs naturally transfer to ME-POMDPs, existing results already separate the two, with finite-horizon reward-maximization being PSPACE-complete for deterministic POSGs¹ but NP-complete for deterministic ME-POMDPs, as we show. Like POSGs, *extensive-form games* also involve partial information and have well-known complexity results [35]. However, they are strictly more general than ME-POMDPs in the structure of the adversary’s actions, and naive encodings of ME-POMDPs as imperfect-information games lead to an exponential blow-up in representation size.

¹This fact follows from the proof of [46].

3 Preliminaries

The sets of real numbers and natural numbers are $\mathbb{R}$ and $\mathbb{N}$, respectively. We denote the set of integers $\{1, \dots, n\}$ as $[n]$. We use $\Delta(B)$ to denote the set of distributions on elements of a set B . For a distribution $q \in \Delta(X)$, $\text{supp}(q)$ is the set of elements $x \in X$ such that $q(x) > 0$. For $x \in X$, $\delta_x \in \Delta(X)$ is the Dirac distribution which places all probability mass on x . For a vector $\mathbf{u}$ we use $\mathbf{u}[i]$ or $\mathbf{u}(i)$ to denote the i^{th} element, and $\mathbf{u}^\top$ denotes its transpose.

We assume basic familiarity with standard complexity classes P, NP, and PSPACE, as well as the classical NP-complete problem 3SAT, its PSPACE-complete variant QSAT [45], and their *conjunctive normal form* (CNF). We also consider the complexity class $\exists\mathbb{R}$. The first-order theory of the reals over $(\mathbb{R}, 0, 1, +, \cdot, <)$ is the logic used to express whether a polynomial system of (in)equalities has a solution in $\mathbb{R}$. The *existential* theory of the reals is the fragment that asks whether $\exists x_1, \dots, x_n. \phi(x_1, \dots, x_n)$ has a solution for a given quantifier-free formula ϕ . The truth problem for this fragment defines the complexity class $\exists\mathbb{R}$, with $\text{NP} \subseteq \exists\mathbb{R} \subseteq \text{PSPACE}$ [50].

3.1 POMDPs

A partially observable Markov decision process (POMDP) models an environment in which an agent takes actions to maximize reward under partial information on the true state of the world.

Definition 3.1 (POMDP). A partially observable Markov decision process (POMDP) is a tuple (S, A, Z, T, O, R, b, H) , where S is a finite set of states, A is a finite set of actions, Z is a finite set of observations, $T: S \times A \rightarrow \Delta(S)$ is a transition function, $O: S \rightarrow \Delta(Z)$ is an observation function, $R: S \times A \rightarrow \mathbb{R}$ is a reward function, $b \in \Delta(S)$ is an initial state distribution, and $H \in \mathbb{N}$ is the horizon.

We write $T(s, a, s') = T(s, a)(s')$ and similarly, $O(s, z) = O(s)(z)$. The state s_0 at time 0 is drawn according to b . For every $t \geq 0$, the observation z_t is drawn from $O(s_t)$. The agent then observes z_t , takes an action a_t , and the next state is drawn from $T(s_t, a_t)$.

An agent acts in a POMDP using a policy that maps observation histories to distributions over actions: $\pi: \cup_{t=0}^{H-1} Z(AZ)^t \rightarrow \Delta(A)$. We denote by $\Pi^{Z,A,H}$ the set of all such history-dependent randomized policies for a given observation space, action space, and horizon. When the POMDP is clear from context, we will simply write Π . A policy π is deterministic if, for all histories h in $\cup_{t=0}^{H-1} Z(AZ)^t$, $\pi(h)$ is a Dirac distribution, and we use Π_{det} to denote the set of deterministic policies. A *mixed policy* is a distribution over deterministic policies, and the agent randomly draws one such policy before playing. In the finite-horizon setting, any mixed policy π' has an equivalent behavioral policy π that induces the same path distribution in all POMDPs [36].

The finite-horizon expected reward of a policy π in a POMDP $\mathcal{M}$ is $V_\pi^\pi = \mathbb{E}_\pi^\pi \left[\sum_{t=0}^{H-1} R(s_t, a_t) \right]$, where s_t and a_t are random variables representing the state and action respectively at time t , for t in the set $\{0, 1, \dots, H-1\}$, and we use $\mathbb{E}_\pi^\pi$ to denote the expectation under the measure induced by the policy π and the POMDP $\mathcal{M}$.

3.2 Multi-Environment POMDPs

A multi-environment POMDP (ME-POMDP) comprises a finite set of POMDPs with common state, action, and observation spaces, and a common horizon, but potentially different dynamics.

Definition 3.2 (ME-POMDP). A multi-environment POMDP (ME-POMDP) is a tuple $(S, A, Z, (T_i)_{i \in [n]}, (O_i)_{i \in [n]}, (R_i)_{i \in [n]}, (b_i)_{i \in [n]}, H)$. Each index $i \in [n]$ defines a POMDP $\mathcal{M}_i = (S, A, Z, T_i, O_i, R_i, b_i, H)$.

We emphasize that we define ME-POMDPs *without* a distribution over environments; thus, ME-POMDPs are not a subclass of POMDPs. We define the set of history-dependent policies in the same manner as we do for POMDPs. We evaluate policies in ME-POMDPs in a *worst-case* fashion. For a policy π in a ME-POMDP $(\mathcal{M}_i)_{i \in [n]}$, we define its value as

$$V_{(\mathcal{M}_i)_{i \in [n]}}^\pi = \min_{i \in [n]} V_{\mathcal{M}_i}^\pi = \min_{i \in [n]} \mathbb{E}_{\mathcal{M}_i}^\pi \left[\sum_{t=0}^{H-1} R(s_t, a_t) \right]. \quad (1)$$

A *multi-observation POMDP* (MO-POMDP) is a ME-POMDP in which the environments differ only in their observation functions. That is, for all i, j in $[n]$, we have $T_i = T_j$, $R_i = R_j$ and $b_i = b_j$. A MO-POMDP additionally satisfies the *fixed support* condition if, for all states $s \in S$ and environments $i, j \in [n]$, the set of possible observations is fixed, i.e., $\text{supp}(O_i(s)) = \text{supp}(O_j(s))$.

3.3 Decision Problems and Encodings

We study the complexity of the following decision problem.

Problem 1 (Value problem). Given a ME-POMDP $(\mathcal{M}_i)_{i \in [n]}$, and a rational number B , the value problem asks whether there exists a policy π such that $V_{(\mathcal{M}_i)_{i \in [n]}}^\pi \geq B$.

The value problem for restricted policy classes is defined analogously. Problem 1 considers history-dependent randomized policies $\pi \in \Pi$, unless noted otherwise. The value problem for single-environment models follows from Equation (1) with $n = 1$.

We encode each ME-POMDP following the standard conventions for single-environment POMDPs [46]. All transition, observation, and reward functions, as well as the initial distributions, are given as tables, where each entry is a rational number encoded as a ratio of two integers given in binary. The horizon is encoded in unary.

4 Single-Environment Models

In this section, we introduce POMDP subclasses we consider and the complexities of their reward-maximization problems. Figure 3 summarizes the subclasses we consider. We give complete proofs for all novel single-environment complexity results in Appendix D.

4.1 Subclasses of POMDPs

Recall a POMDP is a tuple (S, A, Z, T, O, R, b, H) . We consider the following subclasses of POMDPs, ordered by increasing restriction.

Revealing POMDPs (RevPOMDPs). In a RevPOMDP, each state has a non-zero probability of revealing the true state through a unique observation [4, 8]. Formally, a RevPOMDP is a POMDP where Z contains a subset $\{\tau_s \mid s \in S\}^2$ of observations such that $\forall s \in S, O(s, \tau_s) > 0 \wedge \forall s' \in S, s' \neq s \implies O(s', \tau_s) = 0$. This definition encodes a variant of the strongly revealing POMDPs of [8] or revealing POMDPs of [4] with state-based observations.

²The subset of Z can be any $|S|$ observations as long as the observation function maps a unique state to each observation.

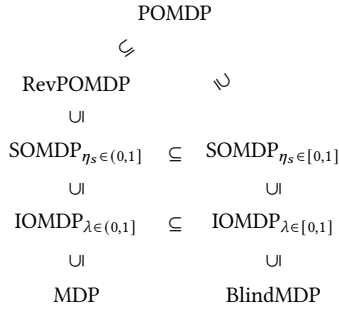


Figure 3: Overview of the models considered in this paper.

RevPOMDPs are decidable for almost-sure, limit-sure, and quantitative reachability and parity objectives [4, 8], in contrast to general POMDPs where most objectives are undecidable [19, 41].

Semi-Observable MDPs (SOMDPs). In a SOMDP, each state has a revealing observation and is otherwise unobservable [7]. The revelation probability may vary across states, and is encoded by a likelihood function $\eta: S \rightarrow [0, 1]$. Formally, a SOMDP is a POMDP where the observation set comprises revealing observations for each state and a blind observation, i.e., $Z = \{\tau_s \mid s \in S\} \cup \{\perp\}$. The observation function then must satisfy $O(s) = \{\tau_s \mapsto \eta(s), \perp \mapsto 1 - \eta(s)\}$ for some function $\eta: S \rightarrow [0, 1]$. This definition corresponds to a variant of SOMDPs with state-based observations [7]. In SOMDPs, one can track the action-observation history by tracking the last observed state and all actions taken since then [7].

Intermittently-Observable MDPs (IOMDPs). In an IOMDP, each state has a revealing observation and is otherwise unobservable [21], as in a SOMDP. However, IOMDPs additionally require the revelation probability to be constant across states. Formally, an IOMDP is a POMDP where $Z = \{\tau_s \mid s \in S\} \cup \{\perp\}$, and the observation function satisfies $O(s) = \{\tau_s \mapsto \lambda, \perp \mapsto 1 - \lambda\}$ for some $\lambda \in [0, 1]$. In an IOMDP, action-observation histories are tracked by the last observed state and all actions since then.

Blind MDPs (BlindMDPs). In a BlindMDP, there is only a single observation [41, 44, 46]. Formally, it is a POMDP where $Z = \{\perp\}$, and for each state s , $O(s, \perp) = 1$.

MDPs. MDPs are a fully observable subclass of POMDPs, in which each state has a single unique observation [48]. Formally, an MDP is a POMDP where $Z = \{\tau_s \mid s \in S\}$ consists of a revealing observation for each state τ_s , also called a *state labeling*, and the observation function is deterministic with $O(s, \tau_s) = 1$.

Partial orders on subclasses. In Figure 3, we arrange the above POMDP subclasses by the inclusion relation. Note that the subset inclusion of Blind MDPs depends on whether the three middle models, i.e., IOMDPs, SOMDPs, and RevPOMDPs, allow revelation probabilities of 0. Also note that allowing RevPOMDPs to include zero revelation probabilities recovers the class of all POMDPs.

Deterministic dynamics. For each subclass introduced above, we also consider a variant with deterministic transition and observation functions. Note that, when the observation function is deterministic,

RevPOMDPs, SOMDPs with $\eta_s \in (0, 1]$, and IOMDPs with $\lambda \in (0, 1]$ all collapse to MDPs as each revelation probability becomes 1. Also, IOMDPs with $\lambda \in [0, 1]$ collapse to either MDPs or Blind MDPs, as the revelation probability is constant.

Subclasses not considered. We focus on the subclasses above as they prove sufficient to uncover interesting complexity behavior. However, we also note that there are numerous other interesting subclasses of POMDPs which we do not characterize, such as posterior-deterministic POMDPs [24], quasi-deterministic POMDPs [11] or observable POMDPs [26]. We leave exploration of multi-environment versions of these subclasses to future work.

4.2 Complexity of Single-Environment Models

We discuss the complexity of finite-horizon reward maximization in single-environment models. The first column of Figure 2 summarizes the results. First, recall that this problem for MDPs without any restrictions is in P [46], and for Blind MDPs it is NP-complete [44, 46]. We now show that for all models between IOMDPs and SOMDPs the problem is NP-complete.

PROPOSITION 4.1. *The value problem for IOMDPs with $\lambda \in (0, 1]$ is NP-hard.*

PROOF SKETCH. We reduce 3SAT to the value problem for IOMDPs. For each 3SAT instance, one can build a BlindMDP with non-positive reward such that the 3SAT instance is satisfiable if and only if the BlindMDP has zero expected reward [44]. To prove NP-hardness for IOMDPs, we use the same transition structure, but allow the agent to see the true state with probability $1/2$. With positive probability, the agent never observes the state and, in effect, plays a sequence of actions in a BlindMDP, and we can argue that this sequence of actions also defines a satisfying assignment for the original 3SAT instance. $\square$

Proposition 4.1 implies that finite-horizon reward maximization for all models in the class $\text{IOMDP}_{\lambda \in (0,1]}$ is NP-hard. We next provide an NP upper bound for all models below $\text{SOMDP}_{\eta_s \in [0,1]}$.

PROPOSITION 4.2. *The value problem for SOMDPs with $\eta_s \in [0, 1]$ is in NP.*

PROOF SKETCH. We show that the optimal policy in a SOMDP can be guessed and evaluated in polynomial time. This policy is belief-based and deterministic, tracking the last observed state, the time of the last observation, and the current time. The belief in an SOMDP collapses every time a state is revealed. Until the next state revelation, all observations are blind, so the decision-maker has only the actions and number of time steps that have passed as available information. Since the actions are chosen deterministically, tracking the last revelation time and the current time step suffices to track the belief. $\square$

We complete the complexity results for single-environment models without deterministic dynamics by proving that reward maximization in RevPOMDPs is PSPACE-hard.

PROPOSITION 4.3. *The value problem for revealing POMDPs is PSPACE-hard.*

PROOF SKETCH. We reduce from QSAT. For each QSAT instance, one can build a POMDP with non-positive reward such that the QSAT instance is satisfiable if and only if the POMDP has zero expected reward [44]. To prove PSPACE-hardness for RevPOMDPs, we use the same transition structure, but allow the agent to see the true state with probability $1/2$. With positive probability, the agent never observes the state and, in effect, plays a sequence of actions in a POMDP, and we can argue that this sequence of actions also defines a satisfying assignment for the original QSAT instance. $\square$

Deterministic Single-Environment Models. Next, we discuss subclasses with deterministic dynamics. Deterministic MDPs are a subset of MDPs, and are thus decidable in P. Deterministic RevPOMDPs collapse to MDPs because they must have a positive probability of revealing each state. Therefore, every model that is a subset of RevPOMDPs lies in P for deterministic dynamics. For BlindMDPs, the reduction used to prove NP-hardness in [44] uses deterministic transitions. As such, BlindMDPs, and any superset model, remain NP-hard in the deterministic case. Finally, the general deterministic POMDP (DetPOMDP) problem lies in NP [12, 40].

4.3 Complexity of Policy Restrictions

Restricting policy classes also reduces POMDP complexity, and we next discuss the complexity of the associated decision problems. Under a time-dependent Markovian policy, an agent chooses an action using only their last observation and the current time, i.e., $\pi: \{0, \dots, H-1\} \times Z \rightarrow \Delta(A)$. We write Π_{MR} for the set of time-dependent Markovian randomized policies. A policy is *stationary* if it only depends on the observation, and not the time step.

The value problem under time-dependent Markovian policies is NP-complete for all POMDP subclasses besides MDPs. By [44], the value problem for time-dependent Markovian policies lies in NP when restricted to deterministic policies, and these deterministic policies are optimal among time-dependent Markovian policies [5]. Meanwhile, NP-hardness for Blind MDPs follows by observing that history-dependent and time-dependent Markovian policies coincide in this subclass. For IOMDPs with $\lambda \in (0, 1]$, we can also prove NP-hardness using similar arguments to those used in proving NP-hardness for Blind MDPs.

PROPOSITION 4.4. *The value problem under time-dependent Markovian policies for IOMDPs is NP-hard, even when $\lambda \in (0, 1]$.*

PROOF SKETCH. For both time-dependent Markovian and history-dependent policies, the agent’s policy on the event that it never observes the state is a sequence of actions. Analogously to the proof of Proposition 4.1, we can establish a correspondence between these action sequences and satisfying assignments of a 3SAT instance, which we in turn use to establish NP-hardness. $\square$

While not shown in Figure 2, we also explore finite-state controllers [43, 47] for POMDPs and their multi-environment counterparts. A generalization of a time-dependent Markovian policy is a time-dependent FSC, in which an agent can use a set of memory states to store information about past actions and observations.

Definition 4.5 (Time-dependent FSC). For a (ME-)POMDP with observations Z , actions A , and horizon H , a time-dependent FSC is a tuple $(Q, \iota, (\gamma_t)_{t \in \{0, \dots, H-1\}}, (\eta_t)_{t \in \{0, \dots, H-1\}})$, where Q is a set of

memory states, $\iota \in \Delta(Q)$ is an initial memory-state distribution, $\gamma_t: Q \times Z \rightarrow \Delta(A)$ is a time-dependent action mapping, and $\eta_t: Q \times Z \times A \rightarrow \Delta(Q)$ is a time-dependent memory update mapping.

A time-dependent FSC is equivalent to a standard time-independent FSC with memory nodes $Q' = Q \times \{0, \dots, H-1\}$ where one always updates (q, t) to $(q', t+1)$. The value problem for FSCs mirrors Problem 1, but takes an additional integer $|Q|$ as input to encode the memory size the agent may use.

Problem 2 (FSC value problem). *Given a ME-POMDP $(M_i)_{i \in [n]}$, a rational number B , and a memory bound $|Q|$ polynomial in the ME-POMDP size, the FSC value problem asks whether there exists a time-dependent FSC π such that $V_{(M_i)_{i \in [n]}}^\pi \geq B$.*

LEMMA 4.6. *For any POMDP M , and fixed $|Q|$, there exists a deterministic optimal time-dependent FSC for M . Consequently, the time-dependent FSC value problem for POMDPs is NP-complete.*

PROOF SKETCH. We can construct a product POMDP M' with state space $S \times Q$ [31]. Time-dependent FSCs in the original POMDP then correspond to time-dependent Markovian policies in this product, and *deterministic* time-dependent Markovian policies are already optimal [5]. Thus, we can simply guess and evaluate such policies in M' to decide the FSC value problem. NP-hardness follows from the time-dependent Markovian case, which we can encode by restricting $|Q|$ to be one. $\square$

5 Upper Bounds via Hidden Mixtures

The complexity results for subclasses of POMDPs naturally raise the question of whether they extend to the multi-environment setting. We next provide a general method for characterizing when this extension is possible by characterizing subclass behavior under hidden mixtures. Full proofs reside in Appendix E.

Many positive POMDP subclass complexity results follow due to optimal policy structure. Indeed, if one can guess a policy quickly, then a natural NP algorithm exists for determining the value problem. BlindMDPs and DetPOMDPs exhibit this structure. Indeed, any deterministic policy in a BlindMDP is simply an action sequence, while optimal policies in DetPOMDPs have concise tree representations [40].

We show that BlindMDPs and DetPOMDPs share another favorable property: they are *closed under hidden mixtures*, and this property lets us transfer complexity results to ME-POMDPs. Informally, a subclass C is closed under hidden mixtures if allowing the adversary to randomly choose one of n POMDPs from C yields a new POMDP in C . As we show in this section, ME-POMDPs built from subclasses that (1) satisfy this hidden-mixture condition, and (2) admit structured optimal policies, also admit efficient algorithms.

We first abstract the notion of a policy class. A policy class defines a set of policies for each POMDP depending on its observation, state, and action space and horizon. Let $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{Z}$ be sets encoding all possible state spaces, action spaces, and observation spaces.³

³We remark that while the definition of the set of all state sets seems unrigorous, we give a more formal justification in Appendix E.4.

Definition 5.1 (Policy class). A policy class is a family of policy sets $\Pi_C = (\Pi_C^{S,Z,A,H})_{S \in \mathcal{S}, A \in \mathcal{A}, Z \in \mathcal{Z}, H \in \mathbb{N}}$ such that

$$\Pi_C^{S,Z,A,H} \subset \Pi^{Z,A,H} \quad \forall Z \in \mathcal{Z}, A \in \mathcal{A}, H \in \mathbb{N}. \quad (2)$$

While state-space-dependence may seem unnecessary, it handles optimal policies for DetPOMDPs, whose representation depends on $|S|$. We next need to tie optimal policy classes for POMDP subclasses to their multi-environment counterparts. We bridge this gap by introducing *hidden-mixture* POMDPs.

A *hidden mixture of a finite set of POMDPs* is a POMDP in which the environment is randomly drawn before the agent executes their policy. Fix a set $(\mathcal{M}_i)_{i \in [n]}$ of POMDPs and a distribution $\mathbf{p} \in \Delta([n])$. In the resulting hidden mixture POMDP, the adversary uses $\mathbf{p}$ to draw an environment, and the dynamics unfold according to environment i . We use mixtures to define new POMDP subclasses.

Definition 5.2 (Class of hidden mixture POMDPs). Given a subclass C of POMDPs, $\text{Mix}(C)$ is the set of POMDPs formed by mixing over a finite number of POMDPs from C .

Hidden mixture POMDPs are structurally distinct from ME-POMDPs, as they assume a distribution over environments and, in the terminology of [13], have *prior semantics*. Hidden mixture POMDPs nonetheless serve as a powerful tool for reasoning about ME-POMDP complexity, as our next result shows.

The optimal policy in a ME-POMDP built from a subclass C randomizes over a small set of policies that are optimal in $\text{Mix}(C)$.

PROPOSITION 5.3. Fix a class C of POMDPs, and suppose Π_C is a policy class that contains a deterministic optimal policy for every POMDP in $\text{Mix}(C)$. If $(\mathcal{M}_i)_{i \in [n]}$ is a ME-POMDP with state, action and observation sets S, A, Z such that each POMDP lies in C , then there exists an optimal policy that mixes over n elements of $\Pi_C^{(S \times [n]), Z, A, H}$.

PROOF SKETCH. Let $(\mathcal{M}_i)_{i \in [n]}$ be a ME-POMDP with POMDPs in C where each POMDP has state, action and observation spaces S, A , and Z . The value satisfies the following chain of equalities.

$$\begin{aligned} \max_{\pi \in \Pi} \min_{i \in [n]} V_{\mathcal{M}_i}^\pi &= \min_{\mathbf{p} \in \Delta([n])} \max_{\pi \in \Pi_{\det}^{Z,A,H}} \sum_{i \in [n]} \mathbf{p}[i] V_{\mathcal{M}_i}^\pi \\ &= \min_{\mathbf{p} \in \Delta([n])} \max_{\pi \in \Pi_C^{S \times [n], Z, A, H}} \sum_{i \in [n]} \mathbf{p}[i] V_{\mathcal{M}_i}^\pi = \max_{\pi \in \text{Mix}(\Pi_C^{S \times [n], Z, A, H})} \min_{i \in [n]} V_{\mathcal{M}_i}^\pi. \end{aligned}$$

The first and last equalities apply the min-max theorem for zero-sum matrix games [6], which holds because $\Pi_{\det}$ is finite and the number of environments is finite. The middle inequality uses the optimality of policies in Π_C for mixture POMDPs, along with the fact that each mixture POMDP has state space $S \times [n]$. To deduce that the optimal policy mixes over at most n policies in Π_C , we can then either use a linear programming result or appeal to Carathéodory's theorem, as in [16]. $\square$

Using Proposition 5.3, we can show that the value problem for both deterministic ME-POMDPs and ME-BlindMDPs lies in NP. Critically, these classes satisfy $C = \text{Mix}(C)$. That is, they are *closed* under mixtures. Thus, by Proposition 5.3, we can solve the ME-POMDP by guessing n policies for the original subclass and verifying with a linear program (LP) that we can mix over these policies.

COROLLARY 5.4. The value problem for ME-DetPOMDPs is in NP.

PROOF SKETCH. When we mix over a set of DetPOMDPs $(\mathcal{M}_i)_{i \in [n]}$, we get another DetPOMDP $\tilde{\mathcal{M}}$ with state space $S \times [n]$. By [40], this POMDP $\tilde{\mathcal{M}}$ will have an optimal policy with representation-size polynomial in $n|S|, |A|, |Z|$ and H . These polynomially sized policies form the set Π_C of policies for the class C of deterministic POMDPs. Thus, to determine if a ME-POMDP with deterministic dynamics has a policy with value at least B , we can guess n policies from Π_C and solve an LP, all in polynomial time. $\square$

COROLLARY 5.5. The value problem for ME-BlindMDPs is in NP.

PROOF SKETCH. Any optimal policy for a BlindMDP comprises a sequence of actions. Also, mixing over a set of BlindMDPs $(\mathcal{M}_i)_{i \in [n]}$ results in another BlindMDP $\tilde{\mathcal{M}}$. Thus, by the same argument as Corollary 5.4, we can guess and check a randomized policy in polynomial time. $\square$

The above corollaries settle the complexity of all deterministic ME-POMDP subclasses we consider, and we additionally obtain our first instance of an NP-complete ME-POMDP problem with stochastic dynamics. However, the above mixture reasoning fails for subclasses such as SOMDPs, as they are not closed under mixtures. Similarly, Proposition 5.3 does not apply to restricted policy classes, as a mixture over deterministic time-dependent Markovian policies may not be a time-dependent Markovian policy.

6 Multi-Observation POMDPs

We next present hardness results for subclasses where the mixture reasoning of Section 5 fails. We identify subclasses that reside in P or NP in the single-environment case, but become PSPACE-complete when we examine the relevant *multi-observation* problem. In identifying these subclasses, we expand the range of PSPACE-hard multi-environment problems beyond known examples such as ME-MDPs [38]. Full proofs reside in Appendix F.

First, we observe that multi-observation MDPs (MO-MDPs) are PSPACE-hard despite the significant structure of their observation functions. A MO-MDP comprises a set of MDPs with the same dynamics but potentially different state labelings. We demonstrate that shuffling this state labeling is sufficient to simulate partial observability and thus recover PSPACE-hardness.

PROPOSITION 6.1. The value problem for MO-MDPs is PSPACE-hard.

PROOF SKETCH. In the classical proof of PSPACE-hardness for POMDPs [46], one transforms a QSAT instance to a POMDP in which nature randomly selects one of m clauses in the conjunctive normal form (CNF) formula at the initial time step, and then uses randomness to assign values to universal variables [46]. The agent cannot observe the chosen clause and must therefore use a clause-independent policy to choose existential variables so the formula remains true for any universal variable assignment. We simulate this partial observability in MO-MDPs using multiple state labelings. We first design a reference environment with the same transition structure as the above POMDP, where states corresponding to clause l emit observations in some set Z_l , for $l \in [m]$. We then

define $m - 1$ new environments, where, in environment k , states corresponding to clause l emit observations in the set Z_{l+k} . We can then extract a clause-independent assignment strategy by examining the agents' behavior on observations in the set Z_1 , since for any clause l , there exists an environment in which the corresponding states emit observations in Z_1 . $\square$

This shuffling trick fails if we assume the observation function's support is fixed between environments. However, as we next demonstrate, fixed-support models can still exhibit PSPACE-hardness.

Even with fixed supports, multi-observation semi-observable MDPs remain PSPACE-complete. We again prove this result by adapting the proof of [46], by letting the agent and the environment alternately choose binary variables in a CNF formula. We design an environment for each clause, and we design environment i so that satisfying clause i reveals information that helps the agent reach an accepting state.

PROPOSITION 6.2. *The value problem for MO-SOMDPs with fixed support is PSPACE-hard.*

PROOF SKETCH. We design an environment corresponding to each clause in a CNF formula with n variables. In particular, fix a clause i . The POMDP for clause i has $2n$ stages: odd stages correspond to variable assignment, and even stages correspond to a guessing game in which the agent can reach an accepting state. At variable assignment stages, either the environment or the agent chooses a variable assignment, as in [46]. At the guessing game stage, an unbiased coin is flipped, which the agent must identify to reach an accepting state. If the last variable assignment satisfied clause i , the environment reveals the state, and thus the coin value, with high probability. Conversely, if the last variable assignment did not set the clause to true, the state remains hidden with high probability. In this MO-POMDP, policies that eventually reveal the coin's value with high probability correspond to strategies that satisfy the given QSAT instance. Conversely, given a QSAT solution, we can build a POMDP policy that reveals the information with sufficiently high probability. This multi-environment problem is also a MO-SOMDP with fixed support. At the variable assignment stages, we can reveal the true state in all environments. In the guessing game states, two observations are possible in all environments: the blind observation and the state's unique observation. $\square$

Unlike SOMDPs, however, IOMDPs have sufficient structure to avoid PSPACE-hardness. In an IOMDP, every state has an equal revelation probability. This structure in turn gives MO-IOMDPs certain monotonicity properties [21], which allow us to solve the MO-IOMDP by identifying and solving the worst environment, i.e., the environment with the lowest revelation probability.

PROPOSITION 6.3. *The value problem for MO-IOMDPs with fixed support is in NP.*

PROOF SKETCH. Increasing the revelation probability in an MO-IOMDP cannot decrease the reward. In particular, fix a transition function, and let π_λ be the optimal policy for a given revelation probability λ . By Corollary 1 in [21], increasing the revelation probability to λ' increases the expected reward for π_λ . Thus, to solve the

MO-IOMDP problem, we merely need to guess and check an optimal policy for an IOMDP, which we can do efficiently by following the reasoning used to prove Proposition 4.2. $\square$

This last result provides an instance of a MO-POMDP problem with stochastic dynamics and a non-trivial observation function that avoids PSPACE-hardness. However, the associated monotonicity property is a stringent condition on the MO-POMDP.

7 Restricted Policy Classes

We next demonstrate that a range of restricted policy class problems that are NP-complete for POMDPs are $\exists\mathbb{R}$ -complete when we move to the ME-POMDP setting. Full proofs reside in Appendix G.

7.1 Markovian Policies

The difficulty of finding time-dependent Markovian policies for ME-POMDPs lies in reasoning about decoupled randomness. When finding time-dependent Markovian policies in POMDPs, we can assume the policy is deterministic, i.e., in Π_{det} , and guess a policy in polynomial time. However, optimal ME-POMDP policies generally require randomness. One might try to solve the ME-POMDP problem by reasoning about mixtures over deterministic Markovian policies, as in Section 5. However, a mixture of deterministic Markovian policies may not be a Markovian policy, i.e., $\Pi_{\text{MR}} \not\subseteq \Delta(\Pi_{\text{det}} \cap \Pi_{\text{MR}})$. In particular, the class of time-dependent Markovian policies requires that the randomness be decoupled across time steps, creating problems that are known across game theory and distributed systems; see, e.g. [14, 17, 18, 28].

For all the subclasses we consider, the time-dependent Markovian policy problem for ME-POMDPs is $\exists\mathbb{R}$ -hard. In a ME-BlindMDP, the agent's strategy comprises a sequence of action distributions, one for each time step. We can establish hardness results by viewing each time step as a player on a team. That is, finding a time-dependent Markovian strategy in a ME-BlindMDP means finding a set of mixed strategies for a team to use against an adversary, where team members can not correlate their randomness. Finding optimal team strategies in such games is $\exists\mathbb{R}$ -hard.

PROPOSITION 7.1. *The time-dependent Markovian policy problem for deterministic ME-BlindMDPs is $\exists\mathbb{R}$ -hard.*

PROOF SKETCH. We reduce 2-vs.-1 team matrix games with two players, known to be $\exists\mathbb{R}$ -hard [9, 17, 27], to ME-BlindMDPs. $\square$

In the proof for ME-BlindMDPs, we store the agent's first action in the state, which prevents the result from generalizing to ME-MDPs because the agent can then use this information at the next time step, even with a Markovian policy. However, we can resolve this issue using a state-relabeling construction.

PROPOSITION 7.2. *The time-dependent Markovian policy problem for deterministic multi-environment MDPs is $\exists\mathbb{R}$ -hard.*

PROOF SKETCH. Similar to the proof of Proposition 7.1, we encode team matrix games in ME-MDPs. However, we add additional environments that differ in state-space labeling, preventing the agent from using state observations at the second time step to create correlations between actions at different time steps. $\square$

Propositions 7.1 and 7.2 establish $\exists\mathbb{R}$ -hardness for all the multi-environment problems we consider. We can also prove $\exists\mathbb{R}$ -hardness for the multi-observation problem by changing the observation function to control the agent’s actions and simulate an adversary.

PROPOSITION 7.3. *The time-dependent Markovian policy problem for multi-observation POMDPs is $\exists\mathbb{R}$ -hard.*

PROOF SKETCH. The proof begins with the same reduction as in the proof of Proposition 7.1. However, the environment can no longer modify the transition or reward functions. To address this difficulty, we add a component to the problem in which certain observations are revealed *conditional* on the agent taking a certain action in each environment. Adding this component then allows the adversary to control the environment dynamics by incentivizing the agent to take certain actions. $\square$

We can also easily express the time-dependent Markovian policy problem in $\exists\mathbb{R}$. Indeed, we parameterize the policy with $|Z||A|H$ variables and encode the dynamics as bilinear constraints linking $\mathbb{P}[s_{t+1} = s]$ to $(\mathbb{P}[s_t = s, a_t = a])_{a \in A, s \in S}$.

LEMMA 7.4. *The time-dependent Markovian policy problem for ME-POMDPs is in $\exists\mathbb{R}$.*

PROOF SKETCH. This result follows from standard encodings of robust POMDP problems as non-linear constrained optimization problems, for example as in [22]. Critically, because the ME-POMDP comprises a finite set of POMDPs, the $\exists\mathbb{R}$ -encoding is finite. $\square$

7.2 Policies with Finite Memory

Next, we show a counter-intuitive relationship between memory and complexity in ME-POMDPs. For any memory size, the value problem for time-dependent FSCs in POMDPs lies in NP. In contrast, we show that ME-POMDPs exhibit a cutoff phenomenon, in which the complexity of the FSC problem depends on the ratio of memory size to the number of environments. We adapt the FSC value problem (Problem 2) with a fixed rational parameter α .

Problem 3 (α -FSC value problem). *Given a ME-POMDP $(\mathcal{M}_i)_{i \in [n]}$, a rational number B , and a memory bound $|Q|$ polynomial in the ME-POMDP size that satisfies $|Q| \geq \alpha n$, the α -FSC value problem asks whether there exists a time-dependent FSC π such that $V_{(\mathcal{M}_i)_{i \in [n]}}^\pi \geq B$.*

Setting $\alpha = 1$ gives the set of all FSC-problems where the memory bound is at least the number of environments. For certain subclasses of ME-POMDPs, increasing α past a certain threshold drops the complexity of the α -parameterized problem from $\exists\mathbb{R}$ -complete to NP-complete, as we show next.

Blind MDPs. For ME-BlindMDPs, FSC synthesis is NP-complete when we restrict the FSC size to exceed the number of environments, as we recover the problem with no policy constraints.

LEMMA 7.5. *For $\alpha = 1$, the α -FSC value problem for ME-BlindMDPs is NP-complete.*

PROOF. Let n be the number of environments in the ME-BlindMDP. By Proposition 5.3, the ME-BlindMDP has an optimal policy that randomizes over n action sequences of length H . We can encode this policy in an FSC with $|Q|$ states, since by assumption $|Q| \geq n$,

so we can use the memory to index the sequence and use γ_i to encode the n action sequences. Thus, solving the memory-constrained ME-BlindMDP problem is equivalent to solving the ME-BlindMDP problem without policy constraints. The latter problem is in NP by Corollary 5.5, and NP-hard as Blind MDPs are NP-hard [46]. $\square$

Lemma 7.5 raises the problem of identifying the α value at which complexity drops from $\exists\mathbb{R}$ to NP. As we show below, $\alpha = 1$ is tight, in the sense that the FSC policy existence problem is $\exists\mathbb{R}$ -complete for any smaller value of α . This result follows from the fact that, when $\alpha < 1$, we can encode $\exists\mathbb{R}$ -hard low-non-negative-rank feasibility problems in the time-dependent FSC problem.

THEOREM 7.6. *For any fixed rational $\alpha \in (0, 1)$, the α -FSC value problem for ME-BlindMDPs is $\exists\mathbb{R}$ -complete.*

PROOF SKETCH. We prove this result by linking the existence of time-dependent finite-state controllers to solving certain non-negative-rank-constrained feasibility problems. We first show a one-to-one correspondence between time-dependent FSC controllers with $|Q|$ states and inter-time-step joint action distributions with non-negative rank less than $|Q|$ [55]. We can then build a ME-POMDP that has a non-negative value if and only if a joint action distribution exists satisfying a set of linear constraints, where each action distribution corresponds to a matrix with low non-negative rank. Finally, we adapt existing results on the complexity of non-negative rank-constrained feasibility [42] to show that deciding whether such a rank-constrained distribution exists is $\exists\mathbb{R}$ -hard. Critically, in adapting this result, we must carefully control the relationship between the memory size $|Q|$ and the number n of environments in the instance we build. $\square$

The above results demonstrate a tradeoff between computation at planning and deployment time, whereby we can increase the memory size to reduce the complexity of the planning problem.

General POMDPs. For general POMDPs, the α -FSC value problem is $\exists\mathbb{R}$ -hard even for $\alpha = 1$, and thus generally harder than for BlindMDPs. In effect, introducing non-trivial observation functions allows one to construct ME-POMDPs where the observations consume all the agent’s memory, so the agent cannot couple its actions across time steps at all.

PROPOSITION 7.7. *For $\alpha = 1$, the α -FSC value problem for ME-POMDPs is $\exists\mathbb{R}$ -complete.*

PROOF SKETCH. We use the same techniques as in Proposition 7.1. However, we add a randomly drawn state component to the environment, and we set the rewards so agent must remember this state component to avoid negative reward. This component exhausts the agent’s memory so that it must draw actions at other timesteps independently, allowing us to simulate the 2-vs.-1 matrix game. $\square$

It remains an open problem whether there are lower bounds on $|Q|$ in terms of n , or other ME-POMDP parameters, such that the α -FSC value problem becomes NP-complete.

8 Conclusion

We established a range of new complexity results for finite-horizon ME-POMDPs across subclasses of POMDPs. Our results show that some POMDP subclasses remain tractable when extended to the multi-environment setting. Future work will extend complexity characterizations to more subclasses and examine infinite-horizon and logic-based objectives.

Acknowledgments

Nils Jansen was supported by the ERC Starting Grant DEUCE (101077178). Marnix Suilen was supported by Research Foundation Flanders (FWO) Junior Postdoctoral Fellowship grant 12ALH27N (IDEAS in POMDPs). Caleb Probine and Ufuk Topcu were funded in part by the Air Force Office of Scientific Research (AFOSR) under grant number FA9550-22-1-0403, and by the Office of Naval Research (ONR) under grant number N00014-24-1-2797.

References

- [1] Mikkel Abrahamsen, Anna Adamaszek, and Tillmann Miltzow. 2022. The Art Gallery Problem is $\exists\mathbb{R}$ -complete. *J. ACM* 69, 1, 4:1–4:70.
- [2] Mikkel Abrahamsen and Tillmann Miltzow. 2019. Dynamic Toolbox for ETRINV. *CoRR* abs/1912.08674 (2019).
- [3] Vinayak S. Ahluwalia, Lauren N. Steinle, and Brian T. Denton. 2021. Policy-based branch-and-bound for infinite-horizon Multi-model Markov decision processes. *Comput. Oper. Res.* 126 (2021), 105108.
- [4] Ali Asadi, Krishnendu Chatterjee, David Lurie, and Raimundo Saona. 2026. Revealing POMDPs: Qualitative and Quantitative Analysis for Parity Objectives. In *AAAI*. AAAI Press, 36146–36154.
- [5] J. Andrew Bagnell, Sham M. Kakade, Andrew Y. Ng, and Jeff G. Schneider. 2003. Policy Search by Dynamic Programming. In *NIPS*. MIT Press, 831–838.
- [6] Tamer Başar and Geert Jan Olsder. 1998. *Dynamic noncooperative game theory*. SIAM.
- [7] Connor Basich, John R. Peterson, and Shlomo Zilberstein. 2022. Planning with Intermittent State Observability: Knowing When to Act Blind. In *IROS*. IEEE, 11657–11664.
- [8] Marius Belly, Nathanaël Fijalkow, Hugo Gimbert, Florian Horn, Guillermo A. Pérez, and Pierre Vandenheve. 2025. Revelations: A Decidable Class of POMDPs with Omega-Regular Objectives. In *AAAI*. AAAI Press, 26454–26462.
- [9] Marie Louisa Tølbøll Berthelsen and Kristoffer Arnsfelt Hansen. 2022. On the Computational Complexity of Decision Problems About Multi-player Nash Equilibria. *Theory Comput. Syst.* 66, 3 (2022), 519–545.
- [10] Dimitri Bertsekas, Angelia Nedic, and Asuman Ozdaglar. 2003. *Convex analysis and optimization*. Vol. 1. Athena Scientific.
- [11] Camille Besse and Brahim Chaib-draa. 2009. Quasi-Deterministic Partially Observable Markov Decision Processes. In *ICONIP (1) (Lecture Notes in Computer Science, Vol. 5863)*. Springer, 237–246.
- [12] Blai Bonet. 2009. Deterministic POMDPs Revisited. In *UAI*. AUAI Press, 59–66.
- [13] Benjamin Bordaïs and Jean-François Raskin. 2026. Multi-Environment MDPs with Prior and Universal Semantics. In *ICALP (LIPIcs, Vol. 374)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 171:1–171:15.
- [14] Christian Borgs, Jennifer Chayes, Nicole Immorlica, Adam Tauman Kalai, Vahab Mirrokni, and Christos Papadimitriou. 2008. The myth of the folk theorem. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*. 365–372.
- [15] Eline M. Bovy, Caleb Probine, Marnix Suilen, Ufuk Topcu, and Nils Jansen. 2025. Multi-Environment POMDPs: Discrete Model Uncertainty Under Partial Observability. In *NeurIPS*.
- [16] Léonard Brice, Filip Cano, Krishnendu Chatterjee, Thomas A. Henzinger, and Stefanie Muroya. 2026. Multi-Environment POMDPs with Finite-Horizon Objectives. *CoRR* abs/2605.07537 (2026).
- [17] Léonard Brice, Thomas A. Henzinger, and K. S. Thejaswini. 2026. Dicey Games: Shared Sources of Randomness in Distributed Systems. In *LICS (LIPIcs, Vol. 380)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 23:1–23:26.
- [18] Andrea Celli and Nicola Gatti. 2018. Computational Results for Extensive-Form Adversarial Team Games. In *AAAI*. AAAI Press, 965–972.
- [19] Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. 2010. Qualitative Analysis of Partially-Observable Markov Decision Processes. In *MFCS (Lecture Notes in Computer Science, Vol. 6281)*. Springer, 258–269.
- [20] Krishnendu Chatterjee, Laurent Doyen, Jean-François Raskin, and Ocan Sankur. 2025. The Value Problem for Multiple-Environment MDPs with Parity Objective. In *ICALP (LIPIcs, Vol. 334)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 150:1–150:17.
- [21] Gongpu Chen and Soung Chang Liew. 2026. Intermittently Observable Markov Decision Processes. *IEEE Trans. Autom. Control*. 71, 2 (2026), 1054–1069.
- [22] Murat Cubuktepe, Nils Jansen, Sebastian Junges, Ahmadreza Marandi, Marnix Suilen, and Ufuk Topcu. 2021. Robust Finite-State Controllers for Uncertain POMDPs. In *AAAI*. AAAI Press, 11792–11800.
- [23] Jiu Ding and Noah H Rhee. 2014. When a matrix and its inverse are nonnegative. *Missouri Journal of Mathematical Sciences* 26, 1 (2014), 98–103.
- [24] Nathanaël Fijalkow, Arka Ghosh, Roman Kniazev, Guillermo A. Pérez, and Pierre Vandenheve. 2026. Computing the Reachability Value of Posterior-Deterministic POMDPs. *CoRR* abs/2602.07473 (2026).
- [25] Maris F. L. Galesloot, Roman Andriushchenko, Milan Ceska, Sebastian Junges, and Nils Jansen. 2025. Robust Finite-Memory Policy Gradients for Hidden-Model POMDPs. In *IJCAI*. ijcai.org, 8518–8526.
- [26] Noah Golowich, Ankur Moitra, and Dhruv Rohatgi. 2022. Planning in Observable POMDPs in Quasipolynomial Time. *CoRR* abs/2201.04735 (2022).
- [27] Kristoffer Arnsfelt Hansen. 2019. The Real Computational Complexity of Minmax Value and Equilibrium Refinements in Multi-player Games. *Theory Comput. Syst.* 63, 7 (2019), 1554–1571.
- [28] Kristoffer Arnsfelt Hansen, Thomas Dueholm Hansen, Peter Bro Miltersen, and Troels Bjerre Sørensen. 2008. Approximability and Parameterized Complexity of Minmax Values. In *WINE (Lecture Notes in Computer Science, Vol. 5385)*. Springer, 684–695.
- [29] Milos Hauskrecht and Hamish S. F. Fraser. 2000. Planning treatment of ischemic heart disease with partially observable Markov decision processes. *Artif. Intell. Medicine* 18, 3 (2000), 221–244.
- [30] David Hudák, Maris F. L. Galesloot, Martin Tappler, Martin Kurečka, Nils Jansen, and Milan Ceska. 2026. Finite-State Controllers for (Hidden-Model) POMDPs using Deep Reinforcement Learning. In *AAMAS*. ACM, 1201–1210.
- [31] Sebastian Junges, Nils Jansen, Ralf Wimmer, Tim Quatmann, Leonore Winterer, Joost-Pieter Katoen, and Bernd Becker. 2018. Finite-State Controllers of POMDPs using Parameter Synthesis. In *UAI*. AUAI Press, 519–529.
- [32] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and Acting in Partially Observable Stochastic Domains. *Artif. Intell.* 101, 1-2 (1998), 99–134.
- [33] Narendra Karmarkar. 1984. A New Polynomial-Time Algorithm for Linear Programming. In *STOC*. ACM, 302–311.
- [34] Leonid G. Khachiyan. 1980. Polynomial algorithms in linear programming. *U. S. S. R. Comput. Math. and Math. Phys.* 20, 1 (1980), 53–72.
- [35] Daphne Koller and Nimrod Megiddo. 1992. The complexity of two-person zero-sum games in extensive form. *Games and economic behavior* 4, 4 (1992), 528–552.
- [36] Harold W Kuhn. 1953. Extensive games and the problem of information. *Contributions to the Theory of Games* 2, 28 (1953), 193–216.
- [37] Hanna Kurniawati. 2022. Partially Observable Markov Decision Processes and Robotics. *Annu. Rev. Control. Robotics Auton. Syst.* 5 (2022), 253–277.
- [38] Yann Le Tallec. 2007. *Robust, risk-sensitive, and data-driven control of Markov decision processes*. Ph.D. Dissertation. Massachusetts Institute of Technology.
- [39] Yan Li. 2026. Computational hardness of static distributionally robust Markov decision processes. *Oper. Res. Lett.* 67 (2026), 107465.
- [40] Michael L. Littman. 1996. *Algorithms for sequential decision-making*. Brown University.
- [41] Omid Madani, Steve Hanks, and Anne Condon. 2003. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artif. Intell.* 147, 1-2 (2003), 5–34.
- [42] Angshul Majumdar. 2026. Constrained Nonnegative Gram Feasibility is $\exists\mathbb{R}$ -Complete. *CoRR* abs/2603.19976v1 (2026).
- [43] Nicolas Meuleau, Kee-Eung Kim, Leslie Pack Kaelbling, and Anthony R. Cassandra. 1999. Solving POMDPs by Searching the Space of Finite Policies. In *UAI*. Morgan Kaufmann, 417–426.
- [44] Martin Mundhenk, Judy Goldsmith, Christopher Lusena, and Eric Allender. 2000. Complexity of finite-horizon Markov decision process problems. *J. ACM* 47, 4 (2000), 681–720.
- [45] Christos H. Papadimitriou. 1994. *Computational complexity*. Addison-Wesley.
- [46] Christos H. Papadimitriou and John N. Tsitsiklis. 1987. The Complexity of Markov Decision Processes. *Math. Oper. Res.* 12, 3 (1987), 441–450.
- [47] Pascal Poupart and Craig Boutilier. 2003. Bounded Finite State Controllers. In *NIPS*. MIT Press, 823–830.
- [48] Martin L. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley.
- [49] Jean-François Raskin and Ocan Sankur. 2014. Multiple-Environment Markov Decision Processes. In *FSTTCS (LIPIcs, Vol. 29)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 531–543.
- [50] Marcus Schaefer, Jean Cardinal, and Tillmann Miltzow. 2024. The Existential Theory of the Reals as a Complexity Class: A Compendium. *CoRR* abs/2407.18006 (2024).
- [51] Yaroslav Shitov. 2016. A universality theorem for nonnegative matrix factorizations. *arXiv preprint arXiv:1606.09068* (2016).

- [52] Lauren N. Steimle, Vinayak S. Ahluwalia, Charmee Kamdar, and Brian T. Denton. 2021. Decomposition methods for solving Markov decision processes with multiple models of the parameters. *IJSE Trans.* 53, 12 (2021), 1295–1310.
- [53] Marnix Suilen, Marck van der Vegt, and Sebastian Junges. 2024. A PSPACE Algorithm for Almost-Sure Rabin Objectives in Multi-Environment MDPs. In *CONCUR (LIPIcs, Vol. 311)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 40:1–40:17.
- [54] Marck van der Vegt, Nils Jansen, and Sebastian Junges. 2023. Robust Almost-Sure Reachability in Multi-Environment MDPs. In *TACAS (1) (Lecture Notes in Computer Science, Vol. 13993)*. Springer, 508–526.
- [55] Stephen A. Vavasis. 2009. On the Complexity of Nonnegative Matrix Factorization. *SIAM J. Optim.* 20, 3 (2009), 1364–1377.

A Statement on the use of Artificial Intelligence

A.1 Expanding literature searches

We used large language models to supplement search keywords for literature review. In particular, when searching for literature in areas where search terms were unclear, we used generative models to suggest keywords and search strategies that could be useful for finding papers. We then used these keywords in Google Scholar searches to see whether they yielded new papers that added context to our related work. Examples of prompts provided to ChatGPT are as follows.

- (1) “What are terms for optimization subject to constraints that take the form of “A must have a non-negative factorization with dimension k” ”
- (2) “Consider the following setup for a partially observable stochastic game 1. P1 has full observability 2. P1 moves only at the first time step 3. P2 then plays the rest of the game with some partial observability. Such games are naturally a subclass of extensive-form games, partially observable stochastic games (POSGs), and games on graphs. What theoretical complexity results exist for such games, and what might they have been called? ”

LLM-generated text from keyword generation does not directly appear in the paper. Instead, we used the generated keywords in Google Scholar searches to expand the set of papers we examined when contextualizing our work. When generative models suggested papers, we first sourced them via Google Scholar and read them to identify the results they claimed.

A.2 Identifying folklore results

We also used language models to identify canonical references for folklore results. For example, it is well-known that many different formalisms for POMDPs that differ slightly are actually equivalent. For example, it is well-known that one can transform a POMDP with stochastic observations into one with deterministic observations. Literature on such folklore results, however, is difficult to find via standard search techniques. While one can easily prove these results, we used ChatGPT to try to identify proofs of such folklore results in the literature so they could be referenced. An example of a prompt used for this purpose when developing the paper is as follows.

- “Can you find a POMDP paper that gives the polytime reduction from POMDPs with state-action-based observations to state-based observation functions”

When ChatGPT suggested such papers, we again confirmed their existence and read them to determine what results they actually contained.

The particular models used to generate the responses include GPT-5.6-LUNA, GPT-5.5, and GPT-5.5-MINI.

B Additional preliminaries

For a vector $\mathbf{u}$ we use $\mathbf{u}[i]$ or $\mathbf{u}(i)$ to denote the i^{th} element, and $\mathbf{u}^\top$ denotes its transpose. The entry-wise 1 norm of a matrix $P \in \mathbb{R}^{m \times n}$ is $\|P\|_{1,1} = \sum_{i \in [m], j \in [n]} |P[i, j]|$, and similarly for a vector $\|\mathbf{u}\|_1 = \sum_{i \in [n]} |\mathbf{u}[i]|$. We use $\mathbf{e}_i \in \mathbb{R}^n$ to denote the standard basis vector with a one at entry i and zeros elsewhere, and $\mathbf{1}$ is the all-ones vector. For a pair of matrices $P, Q \in \mathbb{R}^{m,n}$, $\langle P, Q \rangle$ is the element-wise product $\sum_{i \in [m], j \in [n]} P[i, j]Q[i, j]$.

Paths of length H . Given a horizon H , we evaluate the reward on paths of the shape $\rho = s_0, a_0, \dots, s_{H-1}, a_{H-1}$. Although the final state-action pair (s_{H-1}, a_{H-1}) still induces a distribution on the successor state s_H , we ignore this state, as it does not contribute anything to the finite-horizon reward $\sum_{t=0}^{H-1} R(s_t, a_t)$.

Definition B.1 (Belief update). Given a belief b , an action a , and observation z , we compute the successor belief b' using the belief update function $b_{upd}: \Delta(S) \times A \times Z \rightarrow \Delta(S)$, which we define as follows.

$$\begin{aligned} b'(s') &= b_{upd}(b, a, z)(s'), \\ &= \frac{O(s', z) \sum_{s \in S} b(s) \cdot T(s, a, s')}{\sum_{s' \in S} O(s', z) \sum_{s \in S} b(s) \cdot T(s, a, s')}. \end{aligned}$$

Belief update at $t = 0$. Although we are given an initial belief, we also receive an observation of the initial state. Upon receiving this observation, we perform a belief update without action:

$$b_{upd}(b_0, \cdot, z_0)(s) = \frac{O(s, z_0)b_0(s)}{\sum_{s \in S} O(s, z_0)b_0(s)}.$$

Additional policy types. We next list some additional types of policies that we use in proofs in this appendix.

Definition B.2 (Belief-based policies). A time-dependent belief-based policy maps tuples of distributions over states and time to distributions over actions. Formally, a belief-based policy is a function

$$\pi: \Delta(S) \times \{0, \dots, H-1\} \rightarrow \Delta(A).$$

Definition B.3 (Observation-based policies). An observation-based policy maps sequences of observations to distributions over actions, i.e.

$$\pi: \bigcup_{t=1}^H Z^t \rightarrow \Delta(A).$$

B.1 Supporting lemmas

We will repeatedly need to evaluate time-dependent Markovian policies as a subroutine in the algorithms we provide, and as such, we provide the following lemma on the evaluation of deterministic time-dependent Markovian policies in POMDPs. In particular, for an arbitrary POMDP $\mathcal{M}$, we can evaluate a deterministic time-dependent Markovian policy π in polynomial time, and furthermore, the representation of the expected reward of this policy is polynomial in the size of the POMDP. We remark that existing work details polynomial-time procedures for policy evaluation in POMDPs when probabilities are represented with binary fractions, and observation functions are deterministic [44]. For completeness, we provide the following lemma, which handles stochastic observation functions and explicitly bounds the reward's representation size.

LEMMA B.4. *There exists an algorithm that takes a POMDP $\mathcal{M} = (S, A, Z, T, O, R, b, H)$ and a deterministic time-dependent Markovian policy $(\pi_t)_{t \in \{0, \dots, H-1\}}$ and evaluates the expected reward $\mathbb{E}_{\mathcal{M}}^{\pi} \left[\sum_{t=0}^{H-1} R(s_t, a_t) \right]$ in polynomial time in the representation size $\mathcal{N}$ of the POMDP $\mathcal{M}$.*

Furthermore, the bit lengths of the integers required to represent $\mathbb{E}_{\mathcal{M}}^{\pi} \left[\sum_{t=0}^{H-1} R(s_t, a_t) \right]$ are bounded by a polynomial in $\mathcal{N}$.

PROOF. Before proceeding, we argue that we can process any POMDP instance so that all rationals share a common denominator. We also argue that this procedure runs in polynomial time in the POMDP instance size. In particular, we can argue in a similar manner to the proof of Lemma 2 in [16], and following the notation in [16], we use $\mathcal{N}$ to denote the representation size of the ME-POMDP problem.

We enumerate all rational numbers $(q_k)_{k \in [K]}$ in the POMDP's representation,

$$\underbrace{x_1/y_1, \dots, x_K/y_K}_{q_1 \quad q_K}, \quad (3)$$

where $K \leq \mathcal{N}$ is the number of rational numbers in the POMDP's encoding which are used to represent transition functions, observation functions, rewards and initial state distributions. Clearly, we can rewrite the rational number q_l as

$$q_l = \frac{x_l \prod_{l' \in [K]: l' \neq l} y_{l'}}{\prod_{l' \in [K]} y_{l'}} = \frac{X_l}{C}, \quad (4)$$

where $X_l := x_l \prod_{l' \in [K]: l' \neq l} y_{l'}$, and $C := \prod_{l' \in [K]} y_{l'}$. Furthermore, if the POMDP has an $\mathcal{N}$ -bit representation, then we have $|x_l| \leq 2^{\mathcal{N}}$ and $|y_l| \leq 2^{\mathcal{N}}$ for all $l \in [K]$, which implies that $|X_l| \leq 2^{N^2}$ and $|C| \leq 2^{N^2}$. Computing each X_l along with the constant C comprises multiplying $K \leq \mathcal{N}$ integers each with bit length bounded above by $\mathcal{N}$. Thus, we can compute this common-denominator representation in polynomial time in the POMDP size.

To evaluate the memoryless policy $(\pi_t)_{t \in \{0, \dots, H-1\}}$, we first evaluate the following expressions which encode the probability of being at each state at each time.

$$p_0(s) = b(s) \quad \forall s \in S$$

$$p_{t+1}(s) = \sum_{s' \in S} p_t(s') \sum_{z \in Z} T(s', \pi_t(z))(s) O(s')(z) \quad \forall s \in S, t \in \{0, \dots, H-2\}.$$

We can then argue, by induction, that the probabilities $(p_t(s))_{s \in S}$ have C^{2t+1} as a denominator. Trivially, we can represent the probability $p_0(s)$ as a rational with denominator C for each $s \in S$. Then, assume that $p_t(s) = \frac{w_s^{(t)}}{C^{2t+1}}$ for all $s \in S$, where $w_s^{(t)} \in \{0, \dots, C^{2t+1}\}$ for each s . Then, for each s , we have

$$p_{t+1}(s) = \sum_{s' \in S} \frac{w_{s'}^{(t)}}{C^{2t+1}} \sum_{z \in Z} \frac{u_{s' \pi_t(z)s}}{C} \frac{v_{s' z}}{C} = \frac{w_s^{(t+1)}}{C^{2(t+1)+1}} \quad (5)$$

for some $w_s^{(t+1)} \in \{0, \dots, C^{2(t+1)+1}\}$, where $T(s', a, s) = u_{s' as}/C$ for each $s', s \in S$, and $a \in A$, and $O(s', z) = v_{s' z}/C$ for each $s' \in S, z \in Z$.

We can also evaluate $(p_t(s))_{t \in \{0, \dots, H-1\}}$ in polynomial time. First, note that $C^{2(t+1)+1} \leq 2^{N^2(2H+1)} \leq 2^{3N^3}$, which implies that each integer $w_s^{(t)}$ has at most $3N^3$ bits. To compute each of the $|S|(H-1)$ values $w_s^{(t+1)}$, we need to compute $\sum_{s' \in S, z \in Z} w_{s'}^{(t)} u_{s' \pi_t(a)s} v_{s' z}$, which can clearly be done in polynomial time, since each $w_{s'}^{(t)}$ has at most $3N^3$ bits, and each of the u and v values has at most N^2 bits.

The expected reward for the memoryless policy is

$$\mathbb{E}_{\mathcal{M}}^{\pi} \left[\sum_{t=0}^{H-1} R(s_t, a_t) \right] = \sum_{t=0}^{H-1} \sum_{s \in S, z \in Z} p_t(s) O(s)(z) R(s, \pi_t(z)) = \sum_{t=0}^{H-1} \sum_{s \in S, z \in Z} \frac{C^{2(H-t)} w_s^{(t)}}{C^{2H+1}} \frac{v_{sz}}{C} \frac{\kappa_{s \pi_t(z)}}{C}, \quad (6)$$

where we assume $R(s, a) = \kappa(s, a)/C$ for each $s \in S$ and $a \in A$. Again, we can compute the sum in polynomial time, as all of the integers in the above expression have representation length bounded by a fixed polynomial in $\mathcal{N}$.

We can also observe that the expected reward is equal to $\xi/(C^{2H+3})$ for some integer ξ , which we now bound. In particular, since the expected reward is bounded above by $H \max_{s,a} |R(s, a)|$, we have

$$\left| \frac{\xi}{(C^{2H+3})} \right| \leq H \max_{s \in S, a \in A} |R(s, a)| \leq H \frac{\max_{l \in K} |X_l|}{C}, \quad (7)$$

which implies that

$$\xi \leq HC^{2H+2} \max_{l \in [K]} |X_l| \leq H2^{N^2(2H+2)} 2^{N^2} \leq H2^{5HN^2} \leq 2^{6N^3}.$$

Thus, from the bounds on C and ξ , we deduce the polynomial bit-length condition for the policy's expected reward. $\square$

Next, we discuss a correspondence between FSCs on POMDPs and Markovian policies on a product POMDP. This correspondence is well known, and mirrors the transformation from *time-independent* FSC-synthesis to memoryless policy synthesis in a product POMDP [31]. We give a proof here for completeness in handling the time-dependent details.

LEMMA B.5. *Fix a POMDP $\mathcal{M} = (S, A, Z, T, O, R, b, H)$, and a memory size $|Q|$. In deterministic polynomial time, we can construct an extended POMDP $\mathcal{M}'$ such that there exists a correspondence between*

- (1) *the set of time-dependent FSCs with $|Q|$ states in the POMDP $\mathcal{M}$, and*
- (2) *the set of time-dependent Markovian policies in an expanded POMDP $\mathcal{M}'$,*

such that for every time-dependent FSC with $|Q|$ states in the POMDP $\mathcal{M}$, there exists a time-dependent Markovian policy in an expanded POMDP $\mathcal{M}'$ that achieves the same reward, and vice versa. Furthermore, this policy translation procedure can be done in polynomial time in the policy's representation size.

Furthermore, if the time-dependent Markovian policy π' in the expanded POMDP $\mathcal{M}'$ is deterministic, then there exists a deterministic time-dependent FSC with $|Q|$ states in $\mathcal{M}$ with the same reward.

PROOF. Define a POMDP $\mathcal{M}'$ with

$$\mathcal{M}' = (\underbrace{\{s_\perp\} \cup S \times Q}_{S'}, \underbrace{A \times Q}_{A'}, \underbrace{Z \times Q}_{Z'}, T', O', R', b', H + 1), \quad (8)$$

where we denote the augmented state space as S' , the augmented action space as A' and the augmented observation space as Z' . The state $s_\perp$ is a dummy state, included to allow a random initial FSC state. The transition function encodes the original POMDP's transition function and the memory update.

$$\begin{aligned} T'(s_\perp, (a, q)) &= b \times \delta_q & \forall (a, q) \in A \times Q \\ T'((s, q), (a, q')) &= T(s, a) \times \delta_{q'} & \forall (a, q') \in A \times Q, (s, q) \in S \times Q. \end{aligned}$$

The observation function reveals the memory state and encodes the original observation function.

$$\begin{aligned} O'(s_\perp) &= \delta_{(z^{(*)}, q^{(*)})} \\ O'((s, q)) &= O(s) \times \delta_q \quad \forall s \in S, q \in Q. \end{aligned}$$

In this observation function, $z^{(*)}$ and $q^{(*)}$ are an arbitrary observation-memory-state pair. The reward function encodes the original reward function outside the dummy state, where the reward is zero.

$$\begin{aligned} R'(s_\perp, (a, q)) &= 0 & \forall (a, q) \in A \times Q \\ R'((s, q), (a, q')) &= R(s, a) & \forall (a, q') \in A \times Q, (s, q) \in S \times Q. \end{aligned}$$

Finally, the initial state distribution places all mass on the dummy state, i.e., $b' = \delta_{s_\perp}$.

Clearly, we can construct $\mathcal{M}'$ from $\mathcal{M}$ in polynomial time.

Any FSC $(Q, \iota, (\gamma_t)_{t \in \{0, \dots, H-1\}}, (\eta_t)_{t \in \{0, \dots, H-1\}})$ for the original POMDP corresponds to a time-dependent Markovian policy in $\mathcal{M}'$ which attains the same reward. Indeed, for each $t \in \{0, \dots, H-1\}$, let $\xi_t : Q \times Z \rightarrow \Delta(A \times Q)$ denote the function such that

$$\xi_t((q, z))((a, q')) = \gamma_t(q, z)(a) \eta_t(q, z, a)(q') \quad \forall q, q' \in Q, z \in Z, a \in A. \quad (9)$$

That is, ξ_t encodes the joint distribution on actions and successor memory states given an original memory state and an observation.

Clearly, we can implement ξ_t as a time-dependent Markovian policy π' in the expanded POMDP $\mathcal{M}'$ by setting

$$\begin{aligned} \pi'_0((z^{(*)}, q^{(*)}))((a^{(*)}, q)) &= \iota(q) & \forall q \in Q, \\ \pi'_t((z, q))((a, q')) &= \xi_{t-1}((q, z))((a, q')) & \forall q, q' \in Q, z \in Z, a \in A, t \in \{1, \dots, H\}, \end{aligned}$$

where $a^{(*)} \in A$ is an arbitrary fixed action. Now, if $h = q_0, s_0, z_0, a_0, \dots, q_{H-1}, s_{H-1}, z_{H-1}, a_{H-1} \in (Q \times S \times Z \times A)^H$ is some trajectory in the original POMDP, then the set of trajectories

$$\mathcal{H}_h := \{s_\perp, (z^{(*)}, q^{(*)}), (a^{(*)}, q_0), (s_0, q_0), (z_0, q_0), (a_0, q_1), \dots, (s_{H-1}, q_{H-1}), (z_{H-1}, q_{H-1}), (a_{H-1}, q_H) | q_{H+1} \in Q\} \subset (S' \times Z' \times A')^{H+1} \quad (10)$$

satisfies $\mathbb{P}_{\mathcal{M}}^{\pi}[h] = \mathbb{P}_{\mathcal{M}'}^{\pi'}[\mathcal{H}_h]$. As the two POMDPs give the same reward for trajectories in h and $\mathcal{H}_h$ respectively, we can deduce that π' obtains the same reward in $\mathcal{M}'$ as π does in $\mathcal{M}$.

Conversely, let π' be some time-dependent Markovian policy in $\mathcal{M}'$. We first set ι such that $\iota(q) = \sum_{a \in A} \pi'_0((z^{(*)}, q^{(*)}))((q, a))$ for all $q \in Q$. Then, we set, for each $t \in \{0, \dots, H-1\}$, $q \in Q$, $z \in Z$, and $a \in A$,

$$\gamma_t(q, z)(a) = \sum_{q' \in Q} \pi'_{t+1}((z, q))((q', a)). \quad (11)$$

Next, for each $t \in \{0, \dots, H-1\}$, $q, q' \in Q$, $z \in Z$, and $a \in A$, we set

$$\eta_t(q, z, a)(q') = \begin{cases} \frac{\pi'_{t+1}((z, q))((q', a))}{\gamma_t(q, z)(a)} & \text{if } \gamma_t(q, z)(a) > 0 \\ \delta_{q^{(*)}} & \text{otherwise.} \end{cases} \quad (12)$$

In the above expression $q^{(*)}$ is some arbitrary fixed memory-state in Q . Clearly, we have

$$\pi'_{t+1}((z, q))((q', a)) = \eta_t(q, z, a)(q') \gamma_t(q, z)(a) \quad \forall q, q' \in Q, a \in A, z \in Z, t \in \{0, \dots, H-1\}, \quad (13)$$

and thus, by the same logic as above, we can deduce that if $h = q_0, s_0, z_0, a_0, \dots, q_H, s_H, z_H, a_H \in (Q \times S \times Z \times A)^H$ is some trajectory in the *original* POMDP $\mathcal{M}$, then the following set of trajectories in the expanded POMDP $\mathcal{M}'$ has the same probability and reward under π' .

$$\mathcal{H}_h := \{s_\perp, (z^{(*)}, q^{(*)}), (\tilde{a}, q_0), (s_0, q_0), (z_0, q_0), (a_0, q_1), \dots, (s_H, q_H), (z_H, q_H), (a_H, q_{H+1}) | q_{H+1} \in Q, \tilde{a} \in A\} \subset (S' \times Z' \times A')^{H+1}. \quad (14)$$

We can then deduce that the policy π defined by $(\iota, (\gamma_t)_{t \in \{0, \dots, H-1\}}, (\eta_t)_{t \in \{0, \dots, H-1\}})$ attains the same reward as π' .

We have now established the correspondence between the two policy classes, and we remark that each direction of the translation between the two policy classes can be done in polynomial time in the size of the policy.

Furthermore, examining the proof, we see that if a time-dependent Markovian policy π' in $\mathcal{M}'$ is deterministic, we can construct an FSC-based policy in $\mathcal{M}$ with the same reward that is also deterministic. $\square$

C Figures

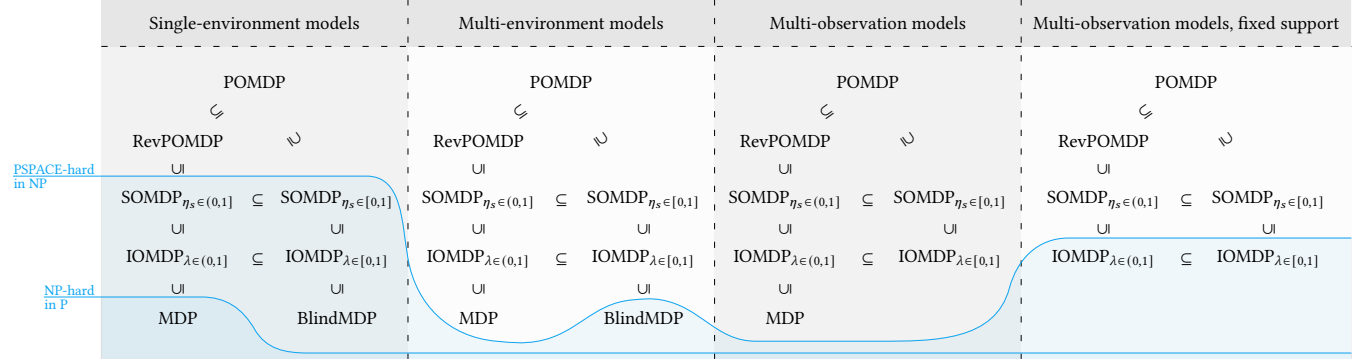


Figure 4: Overview of the complexity results no restrictions

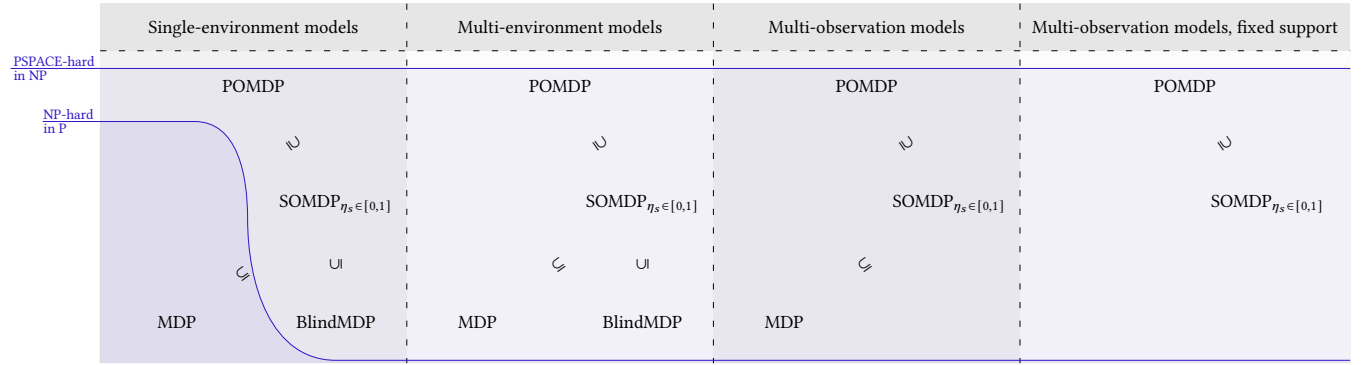


Figure 5: Overview of the complexity results deterministic models

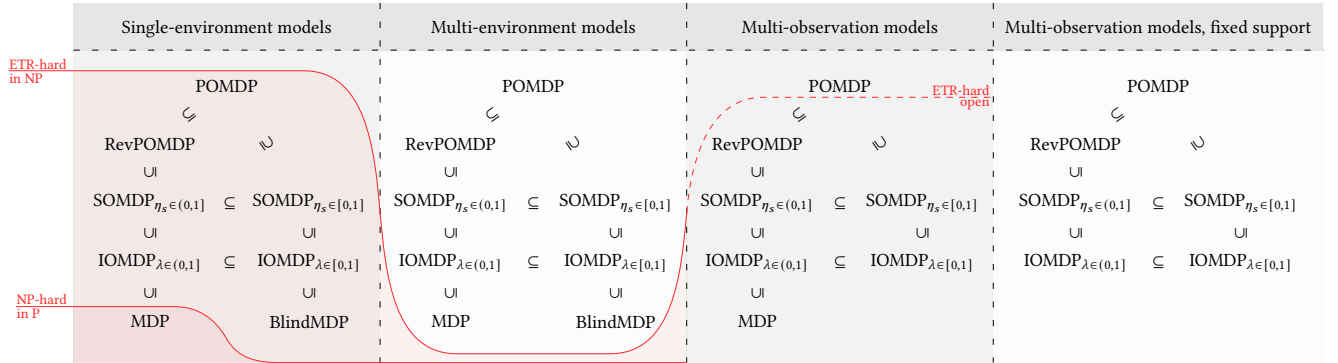


Figure 6: Overview of the complexity results memoryless policies

D Proofs for Section 4.2

D.1 Intermittently-Observable MDPs

PROPOSITION 4.1. *The value problem for IOMDPs with $\lambda \in (0, 1]$ is NP-hard.*

PROOF. We reduce 3SAT to this problem. The reduction is very similar to the reduction from 3SAT to Blind MDPs as presented in [46]. The difference lies in the way we need to deal with state revelations, which do not occur in BlindMDPs.

Fix a 3SAT instance φ , and let $\mathcal{M}' = (S, A, \{\perp\}, T, O, R, b, H)$ be a BlindMDP such that the following conditions hold.

(Assumption 1) The rewards in the POMDP $\mathcal{M}'$ are less than or equal to zero.

(Assumption 2) There exists a policy in $\mathcal{M}'$ with zero expected reward if and only if φ has a solution.

Such a POMDP exists for each 3SAT instance, as one can argue by following the reasoning in [46].

Next, let $\mathcal{M}$ be an IOMDP with all components equal to $\mathcal{M}'$ except for the observation set and function, which we modify such that the true state is revealed with probability $\frac{1}{2}$ at each time step.

We claim that there exists a policy in $\mathcal{M}$ with non-negative reward if and only if the 3SAT instance φ is satisfiable.

($\Rightarrow$) By assumption, the IOMDP $\mathcal{M}$ has a history-dependent π with zero reward, and we can assume this policy to be deterministic, as such policies are optimal among POMDP policies in the finite-horizon case.

We now claim that there exists a policy for the POMDP $\mathcal{M}'$ that also achieves zero reward. Let $h_t^\perp$ be the t step observation history where the state is never revealed, consisting of $t + 1$ $\perp$ -observations. Note that we can, without loss of generality, assume that the optimal policy is observation-based since it is deterministic. Then, let π' be a policy for the BlindMDP $\mathcal{M}'$ where

$$\pi'(h_t^\perp) = \pi(h_t^\perp) \quad \forall t \in 0, \dots, H-1. \quad (15)$$

Evaluating the reward of π , we have

$$\begin{aligned} \mathbb{E}_{\mathcal{M}}^{\pi} \left[\sum_{t=0}^{H-1} R(s_t, a_t) \right] &\leq \sum_{s_0, \dots, s_{H-1} \in S^H} \mathbb{P}_{\mathcal{M}}[s_0] \prod_{t=0}^{H-2} \mathbb{P}_{\mathcal{M}}[s_{t+1}|s_t, \pi(h_t^\perp)] \prod_{t=0}^{H-1} \mathbb{P}_{\mathcal{M}}[\perp|s_t] \left(\sum_{t=0}^{H-1} R(s_t, \pi(h_t^\perp)) \right) \\ &= \left(\frac{1}{2} \right)^H \sum_{s_0, \dots, s_{H-1} \in S^H} \mathbb{P}_{\mathcal{M}}[s_0] \prod_{t=0}^{H-2} \mathbb{P}_{\mathcal{M}}[s_{t+1}|s_t, \pi(h_t^\perp)] \left(\sum_{t=0}^{H-1} R(s_t, \pi(h_t^\perp)) \right) = \left(\frac{1}{2} \right)^H \mathbb{E}_{\mathcal{M}'}^{\pi'} \left[\sum_{t=0}^{H-1} R(s_t, a_t) \right] \end{aligned}$$

In the first inequality, we use the fact that we can split the trajectories where the agent observes the state at least once from the trajectories where the agent never observes the state. Since all rewards are non-positive, the reward contribution from all trajectories where the agent observes the state at least once is upper bounded by zero. Therefore, we can consider the reward contribution of the trajectories where the agent never observes the state as an upper bound on the total reward. The first *equality* follows from the fact that the IOMDP $\mathcal{M}$ has a revelation probability of $\frac{1}{2}$. The last equality follows from the fact that the POMDPs $\mathcal{M}$ and $\mathcal{M}'$ share a transition function, along with the definition of the policy π' . As $\mathbb{E}_{\mathcal{M}}^{\pi} [\sum_{t=0}^{H-1} R(s_t, a_t)]$ is non-negative, we deduce the existence of a policy in $\mathcal{M}'$ with non-negative reward. Then, using (Assumption 2), we deduce that φ is satisfiable.

($\Leftarrow$) By (Assumption 2), there exists a policy in the BlindMDP $\mathcal{M}'$ which attains zero expected reward. We can deploy this policy in $\mathcal{M}$, playing blind and ignoring any revealing observation, and get zero reward. $\square$

D.2 Semi-Observable MDPs

PROPOSITION 4.2. *The value problem for SOMDPs with $\eta_s \in [0, 1]$ is in NP.*

PROOF. Fix a SOMDP. We show that there exists an optimal policy π where the action a for each history $h = z_0, a_0, \dots, z_{t-1}, a_{t-1}, z_t$ is determined by the following data.

- (1) $t' = \sup\{u : z_u \neq \perp\}$, i.e., the last revelation time⁴.
- (2) $s = \begin{cases} q : z_{t'} = \top_q & \text{if } t' \neq -\infty, \\ \text{blind} & \text{if } t' = -\infty \end{cases}$, i.e., the last revealed state (if any).
- (3) t , i.e., the current time.

Let $(\pi_t^*)_{t=0}^{H-1}$ be a deterministic time-dependent belief-based optimal policy for the SOMDP. We will build a policy π which attains the same value while tracking the smaller information set described above. In particular, we will define a policy

$$\pi : (S \cup \{\text{blind}\}) \times (\{0, \dots, H-1\} \cup \{-\infty\}) \times \{0, \dots, H-1\} \rightarrow A. \quad (16)$$

For $s \in S$, $t' \in (\{0, \dots, H-1\} \cup \{-\infty\})$ and $t \in \{0, \dots, H-1\}$, $\pi(s, t', t)$ will encode the action to take when

- The state was last revealed at t' ,
- The current time is t ,
- The state was last revealed to be s .

⁴Using the convention that $\sup(\emptyset) = -\infty$.

Meanwhile, $\pi(\text{blind}, -\infty, t)$ will encode the action to take in the event when, at time t , the state has never been revealed.

We will build the new policy π by inductively computing beliefs and the resulting action. Let $\mathbf{b}_{(s,t',u)}^{\text{seen}}$ denote the belief on the SOMDP's state when

- The state was last revealed at t' ,
- The current time is u ,
- The state was last revealed to be s ,
- We take actions using π^* .

We can inductively define $\mathbf{b}_{(s,t',u)}^{\text{seen}}$. First, we have

$$\mathbf{b}_{(s,t',t')}^{\text{seen}} = \delta_s. \quad (17)$$

That is, if the state was revealed at the current timestep, the belief collapses. Then, we can iteratively update beliefs as follows.

$$a_{(s,t',u)}^{\text{seen}} = \pi_u^*(\mathbf{b}_{(s,t',u)}^{\text{seen}}) \quad \mathbf{b}_{(s,t',u+1)}^{\text{seen}}(s') = \frac{(1 - \eta(s')) \cdot \sum_{s_u \in S} T(s_u, a_{(s,t',u)}^{\text{seen}})(s') \cdot \mathbf{b}_{(s,t',u)}^{\text{seen}}(s_u)}{\sum_{s_{u+1} \in S} (1 - \eta(s_{u+1})) \cdot \sum_{s_u \in S} T(s_u, a_{(s,t',u)}^{\text{seen}})(s_{u+1}) \cdot \mathbf{b}_{(s,t',u)}^{\text{seen}}(s_u)} \quad (18)$$

In the above equation, $a_{(s,t',u)}^{\text{seen}}$ encodes the action the optimal policy will take at time u if the state was last observed at t' to be s . With this recursion, we can then set

$$\pi(s, t', t) = a_{(s,t',t)}^{\text{seen}} \quad (19)$$

To compute $\pi(\text{blind}, -\infty, t, 0)$, we simply propagate the initial belief as appropriate. In particular, let b be the initial state distribution. We define a_t^{blind} inductively. We set $\mathbf{b}_0^{\text{blind}} = b$, and then we use the following recursion.

$$a_t^{\text{blind}} = \pi_t^*(\mathbf{b}_t^{\text{blind}}) \quad \mathbf{b}_{t+1}^{\text{blind}}(s') = \frac{(1 - \eta(s')) \cdot \sum_{s_t \in S} T(s_t, a_t^{\text{blind}})(s') \cdot \mathbf{b}_t^{\text{blind}}(s_t)}{\sum_{s_{t+1} \in S} (1 - \eta(s_{t+1})) \cdot \sum_{s_t \in S} T(s_t, a_t^{\text{blind}})(s_{t+1}) \cdot \mathbf{b}_t^{\text{blind}}(s_t)} \quad (20)$$

We then set $\pi(\text{blind}, -\infty, t)$ to a_t^{blind} .

We now claim that π is optimal. We first argue that, for any observable history h that is possible⁵ when the policy π^* is used, π^* and π choose the same action. Fix an observable history h under π^* , where $h = z_0, a_0, \dots, z_{t-1}, a_{t-1}, z_t$, and let a_t be the action π^* chooses upon observing this history. Since h is observable and π^* is deterministic, we know $\forall i \in \{0, \dots, t-1\}, \pi_i^*(b_i) = a_i$ where b_i is the belief after observing history $h_i = z_0, a_0, \dots, z_i$. Additionally, let t' be the latest time u such that $z_u \neq \perp$, and let s be the state such that $z_{t'} = \top_s$.

We argue by induction that, for each $l \in \{0, \dots, t-t'\}$, we have $a_{t'+l} = a_{(s,t',t'+l)}^{\text{seen}} = \pi(s, t', t'+l)$ and $b_{t'+l} = \mathbf{b}_{(s,t',t'+l)}^{\text{seen}}$.

For the base case, when $l = 0$, the state has been revealed and $b_{t'} = \delta_s = \mathbf{b}_{(s,t',t')}^{\text{seen}}$. Additionally, observe that

$$a_{t'} = \pi_{t'}^*(\delta_s) = a_{(s,t',t')}^{\text{seen}}. \quad (21)$$

As stated above, the first equality holds as h is an observable history. Meanwhile, the second equality simply holds from the definition of $a_{(s,t',t')}^{\text{seen}}$.

Consider $l \leq t - t' - 1$. For the inductive step, assume $a_{t'+l} = a_{(s,t',t'+l)}^{\text{seen}}$ and $b_{t'+l} = \mathbf{b}_{(s,t',t'+l)}^{\text{seen}}$. We show $a_{t'+l+1} = a_{(s,t',t'+l+1)}^{\text{seen}}$:

$$\begin{aligned} a_{t'+l+1} &= \pi_{t'+l+1}^*(b_{t'+l+1}) \\ &= \pi_{t'+l+1}^*(b_{\text{upd}}(b_{t'+l}, a_{t'+l}, z_{t'+l+1})) \\ &= \pi_{t'+l+1}^*(b_{\text{upd}}(\mathbf{b}_{(s,t',t'+l)}^{\text{seen}}, a_{(s,t',t'+l)}^{\text{seen}}, \perp)) \\ &= \pi_{t'+l+1}^*(\mathbf{b}_{(s,t',t'+l+1)}^{\text{seen}}) \\ &= a_{(s,t',t'+l+1)}^{\text{seen}} \end{aligned}$$

Additionally, since we update the same belief $b_{t'+l} = \mathbf{b}_{(s,t',t'+l)}^{\text{seen}}$ with the same action and receive the same blind observation, we may also conclude $b_{t'+l+1} = \mathbf{b}_{(s,t',t'+l+1)}^{\text{seen}}$.

By the above induction, $a_t = a_{(s,t',t)}^{\text{seen}}$. By definition π will take the action $a_{(s,t',t)}^{\text{seen}}$ on seeing history h . Thus, on any possible history where the state is revealed at some point, π and π^* give the same action.

For the case of a history $h = z_0, a_0, \dots, z_{t-1}, a_{t-1}, z_t$ where $z_k = \perp$ for all k , we can use similar arguments. In this case, we must show that $a_i = a_i^{\text{blind}}$ and $b_i = \mathbf{b}_i^{\text{blind}}$ for all $i \in \{0, \dots, t-1\}$. For the base case, when $i = 0$, we have by definition $b_0 = b = \mathbf{b}_0^{\text{blind}}$ and:

$$\begin{aligned} a_0 &= \pi_0^*(b) \\ &= \pi_0^*(\mathbf{b}_0^{\text{blind}}) \\ &= a_0^{\text{blind}} \end{aligned}$$

⁵I.e., occurring with probability greater than zero.

For the inductive step, assume $a_i = a_i^{\text{blind}}$ and $b_i = \mathbf{b}_i^{\text{blind}}$. Then we have:

$$\begin{aligned}
 a_{i+1} &= \pi_{i+1}^*(b_{i+1}) \\
 &= \pi_{i+1}^*(b_{\text{upd}}(b_i, a_i, \perp)) \\
 &= \pi_{i+1}^*(b_{\text{upd}}(\mathbf{b}_i^{\text{blind}}, a_i, \perp)) \\
 &= \pi_{i+1}^*(b_{\text{upd}}(\mathbf{b}_i^{\text{blind}}, a_i^{\text{blind}}, \perp)) \\
 &= \pi_{i+1}^*(\mathbf{b}_{i+1}^{\text{blind}}) \\
 &= a_{i+1}^{\text{blind}}
 \end{aligned}$$

Additionally, since we update the same belief $b_i = \mathbf{b}_i^{\text{blind}}$ with the same action and receive the same blind observation, we may also conclude $b_{i+1} = \mathbf{b}_{i+1}^{\text{blind}}$.

By the above induction, we can conclude that $a_t = a_t^{\text{blind}}$, but by definition π will take the action a_t^{blind} on seeing history h . Thus, on any possible history where the state is revealed at some point π and π^* give the same action.

We can easily deduce now that π is optimal. Indeed, let Paths^{π^*} denote the set of possible H -length paths under the policy π^* , where a path ρ takes the form:

$$\rho = s_0, z_0, a_0, \dots, s_{H-1}, z_{H-1}, a_{H-1}. \quad (22)$$

Note that $\sum_{\rho \in \text{Paths}^{\pi^*}} \mathbb{P}^{\pi^*}[\rho] = 1$, and $V^{\pi^*} = \sum_{\rho} \mathbb{P}^{\pi^*}[\rho] R(\rho)$, where $R(\rho)$ is the reward accumulated along a path. It is thus sufficient to show that, for all $\rho \in \text{Paths}^{\pi^*}$, we have $\mathbb{P}^{\pi^*}[\rho] = \mathbb{P}^{\pi}[\rho]$, where π is our newly constructed policy. We can easily observe this as follows

$$\begin{aligned}
 \mathbb{P}^{\pi^*}[s_0, z_0, a_0, \dots, s_{H-1}, z_{H-1}, a_{H-1}] &= \mathbb{P}[s_0] \prod_{t=0}^{H-1} \mathbb{P}[z_t | s_t] \prod_{t=0}^{H-2} \mathbb{P}[s_{t+1} | s_t, a_t] \prod_{t=0}^{H-1} \mathbb{P}^{\pi^*}[a_t | z_0, a_0, \dots, z_{t-1}, a_{t-1}, z_t] \\
 &= \mathbb{P}[s_0] \prod_{t=0}^{H-1} \mathbb{P}[z_t | s_t] \prod_{t=0}^{H-2} \mathbb{P}[s_{t+1} | s_t, a_t] \prod_{t=0}^{H-1} \mathbb{P}^{\pi}[a_t | z_0, a_0, \dots, z_{t-1}, a_{t-1}, z_t] \\
 &= \mathbb{P}^{\pi}[s_0, z_0, a_0, \dots, s_{H-1}, z_{H-1}, a_{H-1}]
 \end{aligned}$$

In the first and last equality, we simply apply conditional probability rules. Note that we drop superscripts in the probability measure when they do not depend on the deployed policy. In the second equality, we use the following facts

- As the path ρ is possible under π^* , $\mathbb{P}^{\pi^*}[a_t | z_0, a_0, \dots, z_{t-1}, a_{t-1}, z_t] = 1$ for each t .
- Since π takes the same action as π^* for any possible history, $\mathbb{P}^{\pi}[a_t | z_0, a_0, \dots, z_{t-1}, a_{t-1}, z_t] = 1$ for each t .

We now conclude that π^* and π give the same reward.

Next, we show that we can evaluate a policy of the proposed structure in polynomial time. First, note that we can represent the policy class using time-dependent FSCs with memory state space $Q = (S \cup \{s_{\perp}\}) \times (\{0, \dots, H-1\} \cup \{-\infty\})$. Given a policy $\pi: (S \cup \{\text{blind}\}) \times (\{0, \dots, H-1\} \cup \{-\infty\}) \times \{0, \dots, H-1\} \rightarrow A$, we construct time-dependent FSC policy $(\iota, (\gamma_t)_{t \in \{0, \dots, H-1\}}, (\eta_t)_{t \in \{0, \dots, H-1\}})$ such that

$$\begin{aligned}
 \iota &= \{(s_{\perp}, -\infty) \mapsto 1\}, \\
 \gamma_t((s, t'), z) &= \begin{cases} \pi(s', t, t) & \text{if } z = \top_{s'}, \\ \pi(s, t', t) & \text{if } z = \perp, \end{cases} \\
 \eta_t((s, t'), z, a) &= \begin{cases} (s', t) & \text{if } z = \top_{s'}, \\ (s, t') & \text{if } z = \perp, \end{cases},
 \end{aligned}$$

where we write $\top_s \in Z$ for the unique revealing observation of state s and $\perp \in Z$ for the blind observation. Note that this FSC is deterministic.

We then use Lemma B.5 and Lemma B.4 to conclude that we can evaluate a deterministic time-dependent FSC in polynomial time.

We can conclude that there exists a non-deterministic algorithm to solve this problem. In particular, we

- (1) Guess a policy by guessing an action for each combination of current time, last revelation time, and revealed state.
- (2) Evaluate the guessed policy via the FSC construction.

□

D.3 Revealing POMDPs

PROPOSITION 4.3. *The value problem for revealing POMDPs is PSPACE-hard.*

PROOF. We reduce QSAT to this problem. The reduction is similar very similar to the reduction from QSAT to POMDPs as presented in [46]. The difference lies in the way we need to deal with state revelations, which do not occur in the POMDPs used in the QSAT reduction.

Fix a QSAT instance φ , and let $\mathcal{M}' = (S, A, Z, T, O, R, b, H)$ be a POMDP such that the following conditions hold.

(Assumption 1) The rewards in the POMDP $\mathcal{M}'$ are less than or equal to zero.

(Assumption 2) There exists a policy in $\mathcal{M}'$ with zero expected reward if and only if φ has a solution.

Such a POMDP exists for each QSAT instance [46].

Next, let $\mathcal{M}$ be an RevPOMDP with all components equal to $\mathcal{M}'$ except for the observation set and function, which we modify such that the true state is revealed with probability $\frac{1}{2}$.

We claim that there exists a policy in $\mathcal{M}$ with non-negative reward if and only if the QSAT instance φ is satisfiable.

($\Rightarrow$) By assumption, the RevPOMDP $\mathcal{M}$ has a policy π with zero reward, and we can assume this policy to be deterministic, as such policies are optimal among POMDP policies in the finite-horizon case.

We now claim that there exists a policy for the POMDP $\mathcal{M}'$ that also achieves zero reward. Since the RevPOMDP is the POMDP extended with state revelations, the set of every non-revealing observation history in the RevPOMDP and the set of all observation histories in the POMDP are the same. Let $h_{t,\perp}^\pi$ be a history obtained by playing π for t steps where the state is never revealed. Then, let π' be a policy for the BlindMDP $\mathcal{M}'$ where

$$\pi'(h_{t,\perp}^\pi) = \pi(h_{t,\perp}^\pi) \quad \forall t \in \{0, \dots, H-1\}. \quad (23)$$

Evaluating the reward of π , we have

$$\begin{aligned} \mathbb{E}_{\mathcal{M}}^\pi \left[\sum_{t=0}^{H-1} R(s_t, a_t) \right] &\leq \sum_{s_0, \dots, s_{H-1} \in S^H} \mathbb{P}_{\mathcal{M}}[s_0] \prod_{t=0}^{H-2} \mathbb{P}_{\mathcal{M}}[s_{t+1}|s_t, \pi(h_{t,\perp}^\pi)] \prod_{t=0}^{H-1} \mathbb{P}_{\mathcal{M}}[\perp|s_t] \left(\sum_{t=0}^{H-1} R(s_t, \pi(h_{t,\perp}^\pi)) \right) \\ &= \left(\frac{1}{2} \right)^H \sum_{s_0, \dots, s_{H-1} \in S^H} \mathbb{P}_{\mathcal{M}}[s_0] \prod_{t=0}^{H-2} \mathbb{P}_{\mathcal{M}}[s_{t+1}|s_t, \pi(h_{t,\perp}^\pi)] \left(\sum_{t=0}^{H-1} R(s_t, \pi(h_{t,\perp}^\pi)) \right) = \left(\frac{1}{2} \right)^H \mathbb{E}_{\mathcal{M}'}^{\pi'} \left[\sum_{t=0}^{H-1} R(s_t, a_t) \right] \end{aligned}$$

In the first inequality, we use the fact that we can split the trajectories where the agent observes the state at least once from the trajectories where the agent never observes the state. Since all rewards are non-positive, the reward contribution from all trajectories where the agent observes the state at least once is upper bounded by zero. Therefore, we can consider the reward contribution of the trajectories where the agent never observes the state as an upper bound on the total reward. The first equality follows from the fact that the RevPOMDP $\mathcal{M}$ has a revelation probability of $\frac{1}{2}$. The last equality follows from the fact that the POMDPs $\mathcal{M}$ and $\mathcal{M}'$ share transition dynamics, along with the definition of the policy π' . As $\mathbb{E}_{\mathcal{M}}^\pi [\sum_{t=0}^{H-1} R(s_t, a_t)]$ is non-negative, we deduce the existence of a policy in $\mathcal{M}'$ with non-negative reward. Then, using (Assumption 2), we know φ is satisfiable.

($\Leftarrow$) By (Assumption 2), there exists a policy in the BlindMDP $\mathcal{M}'$ which attains zero expected reward. We can deploy this policy in $\mathcal{M}$, playing blind when we receive a revealing observation, and get zero reward. $\square$

D.4 Time-dependent Markovian policies in IOMDPs with non-zero revelation probabilities

PROPOSITION 4.4. *The value problem under time-dependent Markovian policies for IOMDPs is NP-hard, even when $\lambda \in (0, 1]$.*

PROOF. We can prove this result by adapting the argument from Proposition 4.1 to time-dependent policies. In particular, fix a 3SAT instance φ , and let $\mathcal{M}' = (S, A, \{\perp\}, T, O, R, b, H)$ be the same BlindMDP satisfying Assumptions 1 and 2 from the proof of Proposition 4.1.

(Assumption 1) The rewards in the POMDP $\mathcal{M}'$ are less than or equal to zero.

(Assumption 2) There exists a policy in $\mathcal{M}'$ with zero expected reward if and only if φ has a solution.

Also, again, let $\mathcal{M}$ be the IOMDP built from $\mathcal{M}'$ where the true state is revealed with probability $\frac{1}{2}$.

We claim that there exists a time-dependent Markovian policy in $\mathcal{M}$ with non-negative reward if and only if the SAT instance φ is satisfiable.

($\Rightarrow$) By assumption, the IOMDP $\mathcal{M}$ has a time-dependent Markovian policy $(\pi_t)_{t=0}^{H-1}$ with non-negative reward, and we can assume this policy to be deterministic, as such policies are optimal among time-dependent Markovian policies in the finite-horizon case [5].

We now claim that there exists a policy for the POMDP $\mathcal{M}$ that also achieves zero reward. Indeed, let π' be a time-dependent policy for the BlindMDP $\mathcal{M}'$ where

$$\pi'_t(\perp) = \pi_t(\perp) \quad \forall t \in 0, \dots, H-1. \quad (24)$$

By the same reasoning as in the proof of Proposition 4.1, we can argue that

$$\mathbb{E}_{\mathcal{M}}^\pi \left[\sum_{t=0}^{H-1} R(s_t, a_t) \right] \leq \left(\frac{1}{2} \right)^H \mathbb{E}_{\mathcal{M}'}^{\pi'} \left[\sum_{t=0}^{H-1} R(s_t, a_t) \right], \quad (25)$$

which implies that π' attains non-negative reward in $\mathcal{M}'$, and thus φ is satisfiable.

($\Leftarrow$) By (Assumption 2), there exists a history-dependent policy π' in the BlindMDP $\mathcal{M}'$ which attains non-negative expected reward. Without loss of generality, we can assume this policy to be deterministic, and as such, the policy π' is equivalent to a policy that plays a sequence of actions deterministically. Thus, there exists a time-dependent Markovian policy in the BlindMDP $\mathcal{M}'$ which attains non-negative expected reward. We can deploy this time-dependent Markovian policy in $\mathcal{M}$ as well and get non-negative reward. $\square$

D.5 Time-dependent FSC-based policies

LEMMA 4.6. *For any POMDP $\mathcal{M}$, and fixed $|Q|$, there exists a deterministic optimal time-dependent FSC for $\mathcal{M}$. Consequently, the time-dependent FSC value problem for POMDPs is NP-complete.*

PROOF. Fix a POMDP $\mathcal{M} = (S, A, Z, T, O, R, b, H)$. By Lemma B.5, we can, in polynomial time, build a new POMDP

$$\mathcal{M}' = (S', A', Z', T', O', R', b', H + 1) \quad (26)$$

such that any $|Q|$ -state time-dependent FSC in $\mathcal{M}$ has a corresponding time-dependent Markovian policy in $\mathcal{M}'$ with the same reward, and vice versa. From this fact, it follows that

$$\max_{\pi \in \Pi_{\text{FSC}(Q)}} \mathbb{E}_{\mathcal{M}}^{\pi} \left[\sum_{t=0}^{H-1} R(s_t, a_t) \right] = \max_{\pi' \in \Pi_{\text{MR}}} \mathbb{E}_{\mathcal{M}'}^{\pi'} \left[\sum_{t=0}^H R(s'_t, a'_t) \right] \quad (27)$$

where $\Pi_{\text{FSC}(Q)}$ is the set of time-dependent-FSC-based policies with Q states.

Additionally, note that by Lemma B.5, we can take any deterministic time-dependent Markovian policy in $\mathcal{M}'$ and convert it to a deterministic time-dependent FSC policy in $\mathcal{M}$. Since there exists an optimal time-dependent Markovian policy in $\mathcal{M}'$ that is deterministic, there also exists an optimal time-dependent FSC-based policy in $\mathcal{M}$ that is deterministic.

To decide whether the POMDP admits an FSC-based policy with $|Q|$ states and reward at least B , we can

- (1) Build the expanded POMDP $\mathcal{M}'$.
- (2) Guess a deterministic time-dependent Markovian policy in $\mathcal{M}'$.
- (3) Evaluate the policy and check whether it attains reward at least B .

The first two steps can clearly be done in non-deterministic polynomial time, while the last stage can be done in polynomial time by Lemma B.4. We note that the assumption of determinism for the time-dependent Markovian policy is without loss of generality, a fact that holds due to the optimality of deterministic policies among time-dependent Markovian policies.

NP-hardness follows from the time-dependent Markovian case, which we can encode by restricting $|Q|$ to be one. $\square$

E Proofs for Section 5

E.1 Optimal ME-POMDP Policies via Hidden-Mixture POMDPs

PROPOSITION 5.3. Fix a class C of POMDPs, and suppose Π_C is a policy class that contains a deterministic optimal policy for every POMDP in $\text{Mix}(C)$. If $(\mathcal{M}_i)_{i \in [n]}$ is a ME-POMDP with state, action and observation sets S, A, Z such that each POMDP lies in C , then there exists an optimal policy that mixes over n elements of $\Pi_C^{(S \times [n]), Z, A, H}$.

PROOF. Let $(\mathcal{M}_i)_{i \in [n]}$ be a ME-POMDP with POMDPs in C where each POMDP has state, action and observation spaces S, A , and Z . The value satisfies the following equality chain.

$$\max_{\pi \in \Pi} \min_{i \in [n]} V_{\mathcal{M}_i}^\pi = \min_{p \in \Delta([n])} \max_{\pi \in \Pi_{\det}^{Z, A, H}} \sum_{i \in [n]} p[i] V_{\mathcal{M}_i}^\pi = \min_{p \in \Delta([n])} \max_{\pi \in \Pi_C^{S \times [n], Z, A, H}} \sum_{i \in [n]} p[i] V_{\mathcal{M}_i}^\pi = \max_{\pi \in \text{Mix}(\Pi_C^{S \times [n], Z, A, H})} \min_{i \in [n]} V_{\mathcal{M}_i}^\pi. \quad (28)$$

The first and last equalities apply the min-max theorem for zero-sum matrix games [6], which holds due to the finiteness of $\Pi_{\det}$ and the finite number of environments. The middle inequality applies the optimality of policies in Π_C for mixture POMDPs, along with the fact that each mixture POMDP is a POMDP with state space $S \times [n]$.

Similar to the proof in [16], we can then argue that there exists an optimal policy mixing over at most n policies.

To align with later algorithmic results, we give a proof based on linear programming (LP). In particular, observe that the final expression in (28) is equal to the optimal value of the following LP.

$$\max_{\substack{\mathbf{q} \in \mathbb{R}^{\Pi_C^{S \times [n], Z, A, H}} \\ B \in \mathbb{R}}} B \quad (29)$$

$$\text{s.t.} \quad \sum_{\pi \in \Pi_C^{S \times [n], Z, A, H}} \mathbf{q}[\pi] = 1, \quad (30)$$

$$\mathbf{q}[\pi] \geq 0 \quad \forall \pi \in \Pi_C^{S \times [n], Z, A, H}, \quad (31)$$

$$\sum_{\pi \in \Pi_C^{S \times [n], Z, A, H}} \mathbf{q}[\pi] V_i^\pi \geq B \quad \forall i \in [n]. \quad (32)$$

The variable B encodes the value of the ME-POMDP, while the vector $\mathbf{q}$ encodes a distribution over deterministic history-dependent policies. The constraint (32) encodes that the policy must attain value B in all environments, while the constraints (30) and (31) encode the validity of the distribution $\mathbf{q}$. Note that we identify the element of the vector $\mathbf{q}$ corresponding to the deterministic policy π with $\mathbf{q}[\pi]$. This LP has an optimal solution since the objective is bounded above by the greatest expected reward obtained under any combination of a POMDP $\mathcal{M}_i$ and a deterministic policy π .

We now show that the feasible set has an extreme point. For any deterministic policy π' in the policy class, the point $\mathbf{q}' := \mathbf{e}_{\pi'}, B' := \min_{i \in [n]} V_i^{\pi'}$ is an extreme point, as the matrix defined by the active constraints at this point is of full rank.

Because the feasible set has an extreme point, by convex optimization theory, this LP has an optimal solution at an extreme point [10, Proposition 3.4.2]. We now show that any such extreme point has at most n non-zero elements in the corresponding probability vector $\mathbf{q}$. Given an extreme point $(\mathbf{q}, B)$, suppose there are L environments for which (32) is tight. Let $\{\iota_1, \dots, \iota_L\} \subseteq [n]$ be the environments i for which $\sum_{\pi \in \Pi_C^{S \times [n], Z, A, H}} \mathbf{q}[\pi] V_i^\pi = B$. Similarly, suppose $\{\kappa_1, \dots, \kappa_R\} \subset \Pi_C^{S \times [n], Z, A, H}$ contains the policies corresponding to the zero elements of $\mathbf{q}$. Let V_i denote the vector in $\mathbb{R}^{\Pi_C^{S \times [n], Z, A, H}}$ such that $V_i[\pi] = V_i^\pi$ for all policies π in $\Pi_C^{S \times [n], Z, A, H}$. We must have

$$\text{rank} \begin{pmatrix} -V_{\iota_1}^\top & 1 \\ \vdots & \vdots \\ -V_{\iota_L}^\top & 1 \\ \mathbf{1}^\top & 0 \\ \mathbf{e}_{\kappa_1}^\top & 0 \\ \vdots & \vdots \\ \mathbf{e}_{\kappa_R}^\top & 0 \end{pmatrix} = 1 + |\Pi_C^{S \times [n], Z, A, H}|, \quad (33)$$

as $(\mathbf{q}, B)$ is an extreme point [10, Proposition 3.3.3], where $\mathbf{e}_j$ is the j^{th} unit vector in the appropriate dimension. However, as the number of rows trivially bounds the rank, and $L \leq n$, we then have

$$1 + |\Pi_C^{S \times [n], Z, A, H}| \leq n + 1 + R, \quad (34)$$

which implies that $R \geq |\Pi_C^{S \times [n], Z, A, H}| - n$. Thus, at an extreme point solution, at most n coordinates of $\mathbf{q}$ are non-zero, and we conclude that the ME-POMDP has an optimal policy that mixes over at most n elements of $\Pi_C^{S \times [n], Z, A, H}$. $\square$

E.2 Deterministic POMDPs

COROLLARY 5.4. *The value problem for ME-DetPOMDPs is in NP.*

PROOF. When we mix over a set of deterministic POMDPs $(\mathcal{M}_i)_{i \in [n]}$, we get another deterministic POMDP $\tilde{\mathcal{M}}$. Indeed, let POMDP i be

$$\mathcal{M}_i = (S, A, Z, T_i, O_i, R_i, b_i, H). \quad (35)$$

Given a mixture distribution $\mathbf{p}$, we can then construct a new deterministic POMDP

$$\tilde{\mathcal{M}} = (S \times [n], A, Z, \tilde{T}, \tilde{O}, \tilde{R}, \tilde{b}, H), \quad (36)$$

where, for all $s \in S, i \in [n]$, we have

$$\begin{aligned} \tilde{T}((s, i), a) &= T_i(s, a) \times \delta_i \\ \tilde{O}((s, i)) &= O_i(s) \\ \tilde{R}((s, i), a) &= R_i(s, a) \\ \tilde{b}(s, i) &= \mathbf{p}[i] b_i(s). \end{aligned}$$

Any policy π in the original ME-POMDP is a valid policy in $\tilde{\mathcal{M}}$ where

$$\mathbb{E}_{\tilde{\mathcal{M}}}^{\pi} \left[\sum_{t=0}^{H-1} \tilde{R}(\tilde{s}_t, \tilde{a}_t) \right] = \sum_{i \in [n]} \mathbf{p}[i] \mathbb{E}_{\mathcal{M}_i}^{\pi} \left[\sum_{t=0}^{H-1} R_t(s_t, a_t) \right]. \quad (37)$$

Critically, this new POMDP $\tilde{\mathcal{M}}$ is also deterministic, as the transition functions $(T_i)_{i \in [n]}$ and observation functions $(O_i)_{i \in [n]}$ are deterministic.

By Lemma 6.3 in [40], deterministic POMDPs have optimal policies with concise representations. In particular, [40] introduces a notion of a *policy tree* to reason about optimal policies in deterministic POMDPs. A policy tree is a tree where each node is labeled with an action, and the descendant relation encodes which subtree to follow for each possible observation [40]. To ease exposition, we introduce a slightly more general notion in the form of a policy-directed-acyclic-graph (policy-DAG). In particular, a policy-DAG comprises H node sets $(U_t)_{t=0}^{H-1}$, along with two sequences of functions

$$\begin{aligned} \text{Act}_t : U_t &\rightarrow A && \text{for each } t \in 0, \dots, H-1, \text{ and} \\ \text{NodeUpd}_t : U_t \times Z &\rightarrow U_{t+1} && \text{for each } t \in 0, \dots, H-2, \end{aligned}$$

along with a mapping $\text{Init} : Z \rightarrow U_0$ specifying an initial node for each initial observation. By Lemma 6.3 in [40], a deterministic POMDP with m states has an optimal policy in the form of a policy-DAG where the number $|U_t|$ of nodes at each stage is bounded by m .

We can apply Proposition 5.3 to deduce that there exists an optimal policy for the original ME-POMDP that mixes over n policy DAGs with at most $n|S|$ nodes per time step, and this fact suggests the following non-deterministic algorithm to decide if the ME-POMDP has a policy with reward at least B .

- (1) Guess n policies $(\pi_j)_{j \in [n]}$ where each policy is a policy DAG with at most $n|S|$ nodes in each set U_t .
- (2) Evaluate each policy π_j in each POMDP $\mathcal{M}_i$.
- (3) Solve an LP to determine if there exists a distribution on these policy DAGs such that the induced mixed policy attains reward at least B in the ME-POMDP

The first stage of the above procedure clearly takes polynomial time, and we now argue that the last two stages do as well. We first fix notation, using $\mathcal{N}$ to refer to the representation size of the ME-POMDP instance. Now, fix a POMDP-index $i \in [n]$, an initial state $s_0 \in \text{supp}(b_i)$, and a policy tree π_j with associated update functions $\text{Init}^{(j)}$, $\text{NodeUpd}^{(j)}$ and $\text{Act}^{(j)}$. As the POMDP $\mathcal{M}_i$ and policy π_j are deterministic, there exists a unique trajectory for each initial state, and we can, in polynomial time, compute the resulting trajectory by applying the following equalities.

$$\begin{aligned} z_t &= O_i(s_t) \quad \forall t \in \{0, \dots, H-1\}, \\ u_0 &= \text{Init}^{(j)}(z_0), \\ u_t &= \text{NodeUpd}_t^{(j)}(u_{t-1}, z_t) \quad \forall t \in \{0, \dots, H-2\}, \\ a_t &= \text{Act}_t^{(j)}(u_t) \quad \forall t \in \{0, \dots, H-1\}, \\ s_{t+1} &= T_i(s_t, a_t) \quad \forall t \in \{0, \dots, H-2\}. \end{aligned}$$

We can evaluate the reward of this trajectory in polynomial time, and we can represent the reward as a rational $r_{is}^{(j)} := p_{is}^{(j)} / q_{is}^{(j)}$ where $p_{is}^{(j)} \leq H2^{NH}$ and $q_{is}^{(j)} \leq 2^{NH}$ and $\mathcal{N}$ is the representation-size. To evaluate the reward of the policy π_j in POMDP $\mathcal{M}_i$, we simply need to compute the expectation $\sum_{s \in S} r_{is}^{(j)} b_i(s)$. By similar reasoning to Lemma B.4, we can deduce that evaluating the reward in environment i for policy j can be done in polynomial time, and furthermore, this expected reward has a polynomially-sized representation with respect to the

original ME-POMDP instance size. Repeating this procedure for each environment i and policy j defines a table of values $(V_i^{(j)})_{i \in [n], j \in [n]}$ where $V_i^{(j)}$ encodes the reward of policy-DAG j in POMDP i and $V_i^{(j)}$ is a rational number with size bounded polynomially in $\mathcal{N}$.

Finally, to determine whether the requisite mixture exists, we need to determine whether a distribution $\mathbf{q}$ exists satisfying the following linear inequalities

$$\sum_{j \in [n]} \mathbf{q}[j] V_i^{(j)} \geq B \quad \forall i \in [n].$$

We can solve this problem by solving an LP. While the corresponding LP has rational coefficients, we can convert this LP to an LP with integer coefficients that is polynomially sized in terms of the original ME-POMDP instance. We can then solve this LP in polynomial time [33, 34]. □

E.3 Blind MDPs

COROLLARY 5.5. *The value problem for ME-BlindMDPs is in NP.*

PROOF. When we mix over a set of BlindMDPs $(\mathcal{M}_i)_{i \in [n]}$, we obviously obtain another BlindMDP with an expanded state space. The optimal policy for this BlindMDP is then a sequence of H actions, and thus by Proposition 5.3, there exists an optimal policy for the ME-BlindMDP which is a mixture of n such sequences.

As such, there exists a non-deterministic algorithm to solve this problem. In particular, we

- (1) Guess n sequences of actions, each of length H , where H is the horizon of the ME-BlindMDP.
- (2) Evaluate each of the n sequences of actions in each BlindMDP.
- (3) Solve a linear program to determine if some mixture of these sequences results in a policy with reward at least B .

The first step clearly runs in polynomial time, and we next argue the same for the last two steps.

Each sequence of actions is equivalent to some time-dependent Markovian policy, and so we can evaluate it in polynomial time with the resulting reward having a polynomially-sized representation. Indeed, let $(\pi_j)_{j \in [n]}$ be the n policies. Lemma B.4 implies that, in polynomial time, we can evaluate each policy in each POMDP. From this procedure we obtain a set of reward values $(V_i^{(j)})_{i \in [n], j \in [n]}$ where $V_i^{(j)}$ encodes the reward of the j^{th} policy in the i^{th} BlindMDP, and each value $V_i^{(j)}$ has a representation $V_i^{(j)} = u_i^{(j)} / v_i^{(j)}$ such that

$$\max_{i \in [n], j \in [n]} \max(|u_i^{(j)}|, v_i^{(j)}) \leq 2^{f(\mathcal{N})} \quad (38)$$

for some fixed instance-independent polynomial $f(\mathcal{N})$ of the instance size $\mathcal{N}$.

To deduce that the third stage runs in polynomial time, we can then use the same argument as in the proof of Corollary 5.4. That is, deciding the third stage is equivalent to solving a linear program with integral coefficients, where the coefficients have bit-size bounded by a polynomial of the size of the original ME-BlindMDP representation. □

E.4 On the definition of policy classes

In Section 5, we defined the notion of policy classes by invoking the notion of the set of all sets. We can make this notion rigorous by assuming that all states, actions, and observations come from some universal set. For example, we could assume that each action, state, and observation is a natural number, i.e.,

$$S \subset \mathbb{N}, A \subset \mathbb{N}, Z \subset \mathbb{N}. \quad (39)$$

In this case, we can identify the set of all state sets with $2^{\mathbb{N}}$, and we can use a similar argument for the sets of all action sets and observation sets.

F Proofs for Section 6

F.1 Multi-Observation MDPs

PROPOSITION 6.1. *The value problem for MO-MDPs is PSPACE-hard.*

PROOF. Let $\exists x_1 \forall x_2 \dots \exists x_{n-1} \forall x_n, F(x_1, \dots, x_n)$ be a quantified formula in conjunctive normal form with n variables and m clauses. We construct a MO-POMDP which has the following properties.

- Each environment is an MDP.
- Each environment only differs in how we label the states.
- There exists a policy with value at least 0 if and only if the QSAT instance is true.

Intuitively, each environment has the same transition structure as the POMDP constructed in [46] to prove PSPACE-hardness, with one branch per clause. We then use the multiple observation functions to mimic partial observability. In particular, we construct an MO-POMDP with environments $(\mathcal{M}_i)_{i \in [m]}$ where

$$\mathcal{M}_i = (S, \underbrace{\{\text{setTrue}, \text{setFalse}\}}_A, Z, T, O_i, R, b, \underbrace{2n+1}_H). \quad (40)$$

The state space comprises an acceptance state, a dead state, and three states for each clause-variable pair. In particular, we have

$$S = \{s_{\text{fail}}, s_{\text{acc}}\} \cup \bigcup_{l \in [m], j \in [n]} \{s_{T_{l,j}}, s_{F_{l,j}}, s_{A_{l,j}}\}. \quad (41)$$

The action space comprises two actions to encode variable assignments

$$A = \{\text{setTrue}, \text{setFalse}\}. \quad (42)$$

Finally, we have a unique observation for each state, i.e.

$$Z = \{z_{\text{fail}}, z_{\text{acc}}\} \cup \bigcup_{l \in [m], j \in [n]} \{z_{T_{l,j}}, z_{F_{l,j}}, z_{A_{l,j}}\}. \quad (43)$$

The transition dynamics then encode the QSAT instance in a similar manner to [46]. We set the initial distribution b so that the initial state of the POMDP is uniformly chosen from $\{s_{A_{1,1}}, \dots, s_{A_{m,1}}\}$, and this random initial state choice corresponds to a random choice of a clause. If x_j is an *existential variable*, the next state after $s_{A_{l,j}}$ follows the agent's action. That is, for each $l \in [m]$, $j \in \{1, 3, \dots, n-1\}$, we have

$$\begin{aligned} T(s_{A_{l,j}}, \text{setTrue}) &= s_{T_{l,j}} \\ T(s_{A_{l,j}}, \text{setFalse}) &= s_{F_{l,j}}. \end{aligned}$$

Meanwhile, if x_j is a *universal variable*, the next state is randomly chosen from $\{s_{T_{l,j}}, s_{F_{l,j}}\}$, i.e., for each $l \in [m]$, $j \in \{2, 4, \dots, n\}$, we have

$$T(s_{A_{l,j}}, a) = \left\{ s_{T_{l,j}} \mapsto \frac{1}{2}, s_{F_{l,j}} \mapsto \frac{1}{2} \right\} \quad \forall a \in A.$$

The transitions from $s_{F_{l,j}}$ and $s_{T_{l,j}}$ then encode the QSAT instance. In particular, for each clause l , the variables $\cup_{j \in [n]} \{s_{A_{l,j}}, s_{F_{l,j}}, s_{T_{l,j}}\}$ form a branch where a clause-satisfying variable assignment sends the process to the accepting state. More precisely, for each $j \leq n-1$, $l \in [m]$, we set

$$\begin{aligned} T(s_{T_{l,j}}, a) &= \begin{cases} s_{\text{acc}} & \text{if } x_j \text{ appears in } C_l \\ s_{A_{l,j+1}} & \text{otherwise.} \end{cases} \quad \forall a \in A \\ T(s_{F_{l,j}}, a) &= \begin{cases} s_{\text{acc}} & \text{if } \neg x_j \text{ appears in } C_l \\ s_{A_{l,j+1}} & \text{otherwise.} \end{cases} \quad \forall a \in A. \end{aligned}$$

We additionally have, for each $l \in [m]$,

$$\begin{aligned} T(s_{T_{l,n}}, a) &= \begin{cases} s_{\text{acc}} & \text{if } x_n \text{ appears in } C_l \\ s_{\text{fail}} & \text{otherwise.} \end{cases} \quad \forall a \in A \\ T(s_{F_{l,n}}, a) &= \begin{cases} s_{\text{acc}} & \text{if } \neg x_n \text{ appears in } C_l \\ s_{\text{fail}} & \text{otherwise.} \end{cases} \quad \forall a \in A. \end{aligned}$$

At this stage, we have merely recalled the transition structure from [46], but we will next introduce the observation function.

Before introducing the observation function, we first fix notation. In particular, for $k \in \{1, \dots, m\}$, we use $\varphi_k : [m] \rightarrow [m]$ to denote the function which permutes the index. That is,

$$\varphi_k(l) = \begin{cases} l+k & \text{if } l+k \in [m] \\ l+k-m & \text{otherwise.} \end{cases} \quad (44)$$

In environment k , for $k \in [m]$, we then set the observation function to satisfy

$$O_k(s_{T_{l,j}}) = z_{T_{\varphi_k(l),j}} \quad O_k(s_{F_{l,j}}) = z_{F_{\varphi_k(l),j}} \quad O_k(s_{A_{l,j}}) = z_{A_{\varphi_k(l),j}} \quad \forall l \in [m], j \in [n]. \quad (45)$$

In words, this definition of the observation function implies that, in environment k , variables corresponding to clause l emit observations corresponding to clause $\varphi_k(l)$. For all environments, we set $O_k(s_{\text{acc}}) = z_{\text{acc}}$ and $O_k(s_{\text{fail}}) = z_{\text{fail}}$.

Finally, we set the reward function to penalize reaching the dead state s_{fail} . That is, we set $R(s_{\text{fail}}, a) = -1$ for all actions a in A , and we set the reward to be zero for all other state-action pairs.

Clearly, this ME-POMDP is a MO-MDP, since the shuffling procedure only changes the labels, and for any fixed environment, each state has a unique observation. We can also clearly build this MO-MDP in polynomial time in the size of the QSAT instance.

We now claim that the QSAT instance is true if and only if the MO-MDP has a policy with non-negative reward.

($\implies$) This fact follows similar logic to the original PSPACE-completeness proof for POMDPs [46]. Indeed, suppose $\sigma : \bigcup_{t=0}^{\frac{n}{2}-1} \{0,1\}^{2t} \rightarrow \{0,1\}$ is a winning strategy for the QSAT instance where $\sigma(b_1, \dots, b_{2t})$ encodes an existential assignment for each bit-history that ensures the QSAT instance is satisfied for all possible universal variable assignments. We can define a natural strategy for the MO-MDP that, for each history,

- (1) Translates $z_{T_{\cdot,\cdot}}$ and $z_{F_{\cdot,\cdot}}$ observations to 1 and 0 respectively.
- (2) Deploys the action that σ defines.

Clearly, this policy reaches the acceptance state for any clause branch and for any of the random choices of the universal variables.

($\impliedby$) Now, assume the MO-MDP has a policy π with a non-negative reward.

We can argue that this strategy is, *w.l.o.g.*, deterministic. Indeed, let $\tilde{\pi}$ be a general history-dependent randomized strategy. This strategy has a representation as a mixed strategy, a fact that follows from [36]. As such, for a deterministic history-dependent policy $\pi \in \Pi_{\text{det}}$, let p_π be the probability of π under the mixed-policy representation of $\tilde{\pi}$. By assumption, $\sum_{\pi \in \Pi_{\text{det}}} p_\pi \mathbb{E}_{\mathcal{M}_i}^\pi [\sum_{t=0}^{H-1} R(s_t, a_t)] \geq 0$ for all environments i . Additionally, all rewards are non-positive. Thus, we can deduce that, for any environment $i \in [m]$, and any deterministic policy π such that $p_\pi > 0$, we have $\mathbb{E}_{\mathcal{M}_i}^\pi [\sum_{t=0}^{H-1} R(s_t, a_t)] \geq 0$. Hence, there must exist some deterministic strategy that attains zero reward in all environments.

We can construct a winning strategy for the QSAT instance using the behavior of π on seeing observations in the set

$$\tilde{Z}_1 := \bigcup_{j \in [n]} \{z_{T_{1,j}}, z_{F_{1,j}}, z_{A_{1,j}}\}. \quad (46)$$

Intuitively, because any clause can produce observations in $\tilde{Z}_1$, the corresponding strategy must provide a valid clause-independent assignment strategy. In particular, we first construct $\sigma : \bigcup_{t=0}^{\frac{n}{2}-1} \{0,1\}^{2t} \rightarrow \{0,1\}$ so that a sequence $b_1, \dots, b_{2t}$ is assigned the action that π takes when the corresponding history is observed. More precisely, let $V_j(0) = z_{F_{1,j}}$ and $V_j(1) = z_{T_{1,j}}$ for each $j \in [n]$. Then, fix an even-length bit string $b_1, \dots, b_j$, i.e., a string ending in a universally quantified variable. Next, inductively define a sequence of actions as follows.

$$\begin{aligned} a_1^{(1)} &= \pi(z_{A_{1,1}}) \\ a_1^{(2)} &= \pi(z_{A_{1,1}}, a_1^{(1)}, V_1(b_1)) \\ &\dots \\ a_t^{(1)} &= \pi(z_{A_{1,1}}, a_1^{(1)}, \dots, V_{t-1}(b_{t-1}), a_{t-1}^{(2)}, z_{A_{1,t}}) \\ a_t^{(2)} &= \pi(z_{A_{1,1}}, a_1^{(1)}, \dots, V_{t-1}(b_{t-1}), a_{t-1}^{(2)}, z_{A_{1,t}}, a_t^{(1)}, V_t(b_t)) \end{aligned}$$

We then define $\sigma(b_1, \dots, b_j)$ as

$$\sigma(b_1, \dots, b_j) = \begin{cases} 1 & \text{if } a_{j+1}^{(1)} = \text{setTrue}, \\ 0 & \text{if } a_{j+1}^{(1)} = \text{setFalse}. \end{cases} \quad (47)$$

We now argue that this strategy σ is indeed winning for the QSAT instance. Fix an arbitrary clause $l \in [m]$. Let $x_1, \dots, x_n$ be the bit string resulting from some choice of universal variables under the strategy σ . We argue that if $x_1, \dots, x_n$ does not satisfy clause C_l , then for some environment in the MO-POMDP, the policy π gets negative reward. Indeed, first observe that, in environment $(m+1)-l$, the l^{th} branch of the transition function produces observations in the set $\tilde{Z}_1 = \bigcup_{j \in [n]} \{z_{T_{1,j}}, z_{F_{1,j}}, z_{A_{1,j}}\}$ while visiting states in the set $\bigcup_{j \in [n]} \{s_{A_{l,j}}, s_{F_{l,j}}, s_{T_{l,j}}\}$ and following a transition structure that sends the agent to the accepting state s_{acc} if they satisfy C_l . Now observe that the transition dynamics will follow the bit string $x_1, \dots, x_n$ with positive probability. Indeed, when the agent takes an action using π at existential decision states, the agent will play an action corresponding to the assignment strategy σ , and at universal states, nature will play the next bit in $x_1, \dots, x_n$ with probability $1/2$. Now, by assumption, $x_1, \dots, x_n$ does not satisfy the clause C_l , and so this path will reach the dead state at the last timestep, where it gets reward -1 , but this fact implies that the strategy π obtains negative reward in at least one environment.

Ultimately, the above reasoning follows the same logic as [46]. The main difference appears in how we use the multi-observation structure to ensure that σ must be a satisfying assignment strategy for every clause $l \in [m]$.

□

F.2 Multi-Observation Semi-Observable MDPs with Fixed Support

PROPOSITION 6.2. *The value problem for MO-SOMDPs with fixed support is PSPACE-hard.*

PROOF. Let $\exists x_1 \forall x_2 \dots \exists x_{n-1} \forall x_n F(x_1, \dots, x_n)$ be a quantified formula with n variables, where F is in conjunctive normal form and has m clauses.

We design a MO-SOMDP where there is one environment $\mathcal{M}_i = (S, \underbrace{\{\text{setTrue}, \text{setFalse}\}}_A, Z, T, O_i, R, b, \underbrace{3n+1}_H)$ for each clause $i \in [m]$.

The state space S comprises an acceptance state, a dead state, and five states per variable

$$S = \{s_{\text{acc}}, s_{\text{fail}}\} \cup \bigcup_{j \in [n]} \{s_{A_j}, s_{T_j^{(0)}}, s_{T_j^{(1)}}, s_{T_j^{(-1)}}, s_{F_j^{(0)}}, s_{F_j^{(1)}}, s_{F_j^{(-1)}}\}. \quad (48)$$

The states s_{A_j} correspond to decision nodes where either the agent chooses a value for an existential variable, or a value for a universal variable is drawn randomly. The state triplets $\{s_{T_j^{(0)}}, s_{T_j^{(1)}}, s_{T_j^{(-1)}}\}$ and $\{s_{F_j^{(0)}}, s_{F_j^{(1)}}, s_{F_j^{(-1)}}\}$ then encode true and false assignments to x_j respectively, and the different states within these triplets encode coin flip outcomes.

We have a unique observation for each state, along with a special blind observation. That is,

$$Z = \{z_{\text{acc}}, z_{\text{fail}}\} \cup \left(\bigcup_{j \in [n]} \{z_{A_j}, z_{T_j^{(0)}}, z_{T_j^{(1)}}, z_{T_j^{(-1)}}, z_{F_j^{(0)}}, z_{F_j^{(1)}}, z_{F_j^{(-1)}}\} \right) \cup \{\perp\}. \quad (49)$$

The transition dynamics encode variable assignment and the random coin flips. In particular, if x_j , for $j \in [n]$, is an *existential* variable, then

$$\begin{aligned} T(s_{A_j}, \text{setTrue}) &= s_{T_j^{(0)}} \\ T(s_{A_j}, \text{setFalse}) &= s_{F_j^{(0)}}. \end{aligned}$$

Meanwhile, if x_j is a *universal* variable, the next state is random, i.e.,

$$T(s_{A_j}, a) = \left\{ s_{T_j^{(0)}} \mapsto \frac{1}{2}, s_{F_j^{(0)}} \mapsto \frac{1}{2} \right\} \quad \forall a \in A.$$

From states in $\{s_{T_j^{(0)}}, s_{F_j^{(0)}}\}$, the environment flips a coin, which we encode in the successor state via the superscript lying in $\{(-1), (1)\}$.

$$T(s_{T_j^{(0)}}, a) = \left\{ s_{T_j^{(1)}} \mapsto \frac{1}{2}, s_{T_j^{(-1)}} \mapsto \frac{1}{2} \right\} \quad T(s_{F_j^{(0)}}, a) = \left\{ s_{F_j^{(1)}} \mapsto \frac{1}{2}, s_{F_j^{(-1)}} \mapsto \frac{1}{2} \right\}$$

If the agent matches their action to the coin, they move to the accepting state. Meanwhile, if the agent fails to match the coin, they move to the next variable node, except on the last coin flip, when they move to the dead state. That is, for each $j \in [n]$, we have

$$\begin{aligned} T(s_{T_j^{(1)}}, \text{setTrue}) &= T(s_{F_j^{(1)}}, \text{setTrue}) = T(s_{T_j^{(-1)}}, \text{setFalse}) = T(s_{F_j^{(-1)}}, \text{setFalse}) = s_{\text{acc}} \\ T(s_{T_j^{(1)}}, \text{setFalse}) &= T(s_{F_j^{(1)}}, \text{setFalse}) = T(s_{T_j^{(-1)}}, \text{setTrue}) = T(s_{F_j^{(-1)}}, \text{setTrue}) = \begin{cases} s_{A_{j+1}} & \text{if } j < n \\ s_{\text{fail}} & \text{if } j = n. \end{cases} \end{aligned}$$

The initial state in each SOMDP is s_{A_1} . That is $b = \delta_{s_{A_1}}$.

The observation changes with the environment in a manner such that satisfying variable assignment allows the agent to see the coin flip value with high probability. We tune the observability of states in the set $\bigcup_{j \in [n]} \{s_{T_j^{(1)}}, s_{T_j^{(-1)}}, s_{F_j^{(1)}}, s_{F_j^{(-1)}}\}$ according to the environment. First, let ϵ be a positive number to be tuned later. If variable j appears positively in clause C_i , then $s_{T_j^{(1)}}$ and $s_{T_j^{(-1)}}$ are fully observable with high probability. That is

$$\begin{aligned} O_i(s_{T_j^{(1)}}) &= \{z_{s_{T_j^{(1)}}} \mapsto 1 - \epsilon, \perp \mapsto \epsilon\} \\ O_i(s_{T_j^{(-1)}}) &= \{z_{s_{T_j^{(-1)}}} \mapsto 1 - \epsilon, \perp \mapsto \epsilon\}. \end{aligned}$$

Conversely, if variable j does not appear positively in clause C_i , then the SOMDP emits the blind observation with probability $1/2$, i.e.,

$$\begin{aligned} O_i(s_{T_j^{(1)}}) &= \left\{ z_{s_{T_j^{(1)}}} \mapsto \frac{1}{2}, \perp \mapsto \frac{1}{2} \right\} \\ O_i(s_{T_j^{(-1)}}) &= \left\{ z_{s_{T_j^{(-1)}}} \mapsto \frac{1}{2}, \perp \mapsto \frac{1}{2} \right\}. \end{aligned}$$

Note that in both cases, the support of $O_i(s_{T_j^{(1)}})$ is $\{z_{s_{T_j^{(1)}}}, \perp\}$, and the same condition holds for $s_{T_j^{(-1)}}$. We define the observation function in an analogous way for the states corresponding to false variable assignment $\{s_{F_j^{(1)}}, s_{F_j^{(-1)}}\}$ depending on whether or not the corresponding variable appears negatively in clause C_i . For every environment i , we design the observation such that all states in the set

$$\{s_{\text{acc}}, s_{\text{fail}}\} \cup \bigcup_{j \in [n]} \{s_{A_j}, s_{T_j^{(0)}}, s_{F_j^{(0)}}\} \quad (50)$$

are fully observable. That is, the only information possibly hidden from the agent is the result of the coin flip at each stage.

Finally, we set the reward function to penalize reaching the dead state s_{fail} . That is, we set $R(s_{\text{fail}}, a) = -1$ for all actions a in A , and we set the reward to be zero for all other state-action pairs.

We now argue that the QSAT instance has a solution if and only if there exists some policy π that attains reward at least $-\nu$, where we set

$$\nu := \frac{1}{2m} \left(\frac{1}{8} \right)^n \quad \epsilon := \frac{1}{2m} \left(\frac{1}{8} \right)^n. \quad (51)$$

Note that we can represent both ν and ϵ as rationals where the denominator is $2m8^n$. As such ν and ϵ have a polynomial bit representation in terms of the QSAT instance, and thus we can still implement this reduction in polynomial time. However, in Remark 1, we also discuss how to modify this transformation so that all integers used to represent the MO-SOMDP are themselves polynomially bounded in terms of the size of the QSAT instance.

($\Rightarrow$) First, assume that the QSAT instance is satisfiable with some variable assignment strategy σ . Then, let π be the corresponding MO-SOMDP policy that

- (1) Takes the action σ recommends at the existential decision nodes.
- (2) Takes action `setTrue` whenever a state in $\{s_{T_j^{(1)}}, s_{F_j^{(1)}}\}$ is revealed, and takes action `setFalse` whenever a state in $\{s_{T_j^{(-1)}}, s_{F_j^{(-1)}}\}$ is revealed.

Note that the MO-SOMDP observation functions reveal the prior true-false assignments to the agent, before the hidden coin flip occurs, so that the agent can implement the action-taking strategy described in the first point. The agent can take any arbitrary action when it observes $\perp$.

Now fix an environment and clause corresponding to the index i . As σ is a winning strategy for the QSAT instance, it follows that, for any positive-probability path, the agent will either reach $s_{T_j^{(0)}}$ for a variable x_j that appears positively in C_i , or the agent will reach $s_{F_j^{(0)}}$ for a variable x_j that appears negatively in C_i . On reaching such a state, the environment flips a coin and reveals its value with probability $1 - \epsilon$. As such, the agent's policy reaches s_{acc} before the final time step with probability at least $1 - \epsilon$, and the agent's reward is at least $-\epsilon = -\nu$.

($\Leftarrow$) Now, conversely, assume that a possibly randomized strategy π^* exists that attains reward at least $-\nu$, and let this strategy be represented by some distribution on the set Π_{det} of deterministic policies where p_π is the probability of π for $\pi \in \Pi_{\text{det}}$. We additionally assume that these deterministic policies are observation-based, i.e., they are mappings

$$\pi : \bigcup_{t=1}^H Z^t \rightarrow A, \quad (52)$$

and we use $\Pi_{\text{det,obs}}$ to denote the set of observation-based deterministic policies. This assumption is *w.l.o.g.*, as for any deterministic policy, there exists a deterministic observation-based policy which induces the same path distribution in any POMDP. We argue that the randomized strategy π^* must place positive weight on a deterministic MO-SOMDP strategy that corresponds to a satisfying assignment.

We first fix some notation. In particular, for each $j \in [n]$, let

$$V_j(0) = z_{F_j^{(0)}} \quad \text{and} \quad V_j(1) = z_{T_j^{(0)}}.$$

That is, V_j encodes a mapping to convert bits to relevant MO-SOMDP observations.

We now define a mapping ξ from $\Pi_{\text{det,obs}}$ to the set Σ of mappings $\sigma : \bigcup_{t=0}^{n/2-1} \{0, 1\}^{2t} \rightarrow \{0, 1\}$ that represent QSAT assignment strategies. In particular, for a deterministic observation-based policy π , and a bit string $b_1, \dots, b_{2t}$, we define an assignment strategy $\xi(\pi)$ according to

$$(\xi(\pi))(b_1, \dots, b_{2t}) = \pi(z_{A_1}, V_1(b_1), \perp, z_{A_2}, V_2(b_2), \perp, \dots, z_{A_{2t}}, V_{2t}(b_{2t}), \perp, z_{A_{2t+1}}). \quad (53)$$

That is, we define $\xi(\pi)$ using the assignment strategy that π encodes on the event that the agent never sees the coin flip's value. Note that we slightly abuse notation, as the policy outputs actions in the set $\{\text{setFalse}, \text{setTrue}\}$ rather than bits, but we use the natural remapping.

We next define a partition of the space of deterministic policies. First, for each $i \in [m]$, define $\mathcal{P}_i$ as the set of observation-based policies π such that strategy $\xi(\pi)$ fails to ensure the clause C_i is satisfied. Next, define $\mathcal{U}_1 = \mathcal{P}_1$, and for all $i \in \{2, \dots, m\}$, define $\mathcal{U}_i = \mathcal{P}_i \setminus (\mathcal{P}_1 \cup \dots \cup \mathcal{P}_{i-1})$. Finally, define $\mathcal{U}_0$ as $\Pi_{\text{det,obs}} \setminus (\mathcal{P}_1 \cup \dots \cup \mathcal{P}_m)$. Clearly $\mathcal{U}_0, \dots, \mathcal{U}_m$ forms a partition of $\Pi_{\text{det,obs}}$. Also, note that $\mathcal{U}_0$ contains deterministic POMDP policies which map, via ξ , to QSAT strategies that satisfy all clauses for any universal variable assignment.

We can now bound the probability with which policies in $\mathcal{U}_i$ are played as part of the mixed strategy for each $i \in [m]$. In particular, fix $\pi \in \mathcal{U}_i$ as an arbitrary observation-based policy such that the corresponding QSAT assignment strategy fails to satisfy clause i for some universal variable assignment. We first lower bound the probability of reaching the dead state in environment i when following π . By definition, there exists a universal variable assignment such that, when paired with the assignment strategy $\xi(\pi)$, clause C_i is not satisfied. Denote the resulting bit string $b_1, \dots, b_n$, where by definition $b_1, \dots, b_n \not\models C_i$. We now argue by induction that the agent will reach the dead state with probability at least $(1/8)^n$. First, assume the history is $z_{A_1}, V_1(b_1), \perp, z_{A_2}, V_2(b_2), \perp, \dots, z_{A_{2t-1}}, V_{2t-1}(b_{2t-1}), \perp, z_{A_{2t}}$, ending in a decision-state for a universal variable. With probability $1/2$, the environment chooses its action matching b_{2t} , so that the bit string continues to be produced and the clause remains unsatisfied. For the guessing stage following this assignment, the next stage has a $+1$ -superscript with probability $1/2$, and a -1 -superscript with probability $1/2$, and each of these states emits the blind observation $\perp$ with probability $1/2$. Thus, for the guessing game, no policy can do better than a $3/4$ success rate, and so the agent must fail the guessing game and transition to $s_{A_{2t+1}}$ with probability at least $1/4$. For the next case, assume the history is $z_{A_1}, V_1(b_1), \perp, z_{A_2}, V_2(b_2), \perp, \dots, z_{A_{2t}}, V_{2t}(b_{2t}), \perp, z_{A_{2t+1}}$, ending in an *existential* variable. By assumption, the policy's next bit choice matches b_{2t+1} , and the clause C_i remains unsatisfied. Thus, by the same logic as above, the environment moves to the state $s_{A_{2t+2}}$ with probability at least $1/4$. By repeating this reasoning inductively, we can deduce that, when the agent follows the deterministic policy π , they will reach the dead state with probability $(1/2)^{n/2} (1/4)^n \geq (1/8)^n$.

We can use this fact to then upper bound $p_{\mathcal{U}_i} := \sum_{\pi \in \mathcal{U}_i} p_\pi$. In particular, by the above argument, we have

$$\mathbb{E}_{\mathcal{M}_i}^{\pi^*} \left[\sum_{t=0}^{H-1} R(s_t, a_t) \right] \leq \sum_{\pi \in \mathcal{U}_i} p_\pi \mathbb{E}_{\mathcal{M}_i}^{\pi} \left[\sum_{t=0}^{H-1} R(s_t, a_t) \right] \leq - \sum_{\pi \in \mathcal{U}_i} p_\pi \left(\frac{1}{8} \right)^n = -p_{\mathcal{U}_i} \left(\frac{1}{8} \right)^n \quad (54)$$

The first inequality follows from the non-positivity of the reward, the second inequality follows from the above reasoning about $\mathcal{U}_i$, and the final equality follows by definition of $p_{\mathcal{U}_i}$.

By assumption, we have

$$-\frac{1}{2m} \left(\frac{1}{8} \right)^n = -v \leq \mathbb{E}_{\mathcal{M}_i}^{\pi^*} \left[\sum_{t=0}^{H-1} R(s_t, a_t) \right] \leq -p_{\mathcal{U}_i} \left(\frac{1}{8} \right)^n, \quad (55)$$

from which we can deduce that $p_{\mathcal{U}_i} \leq \frac{1}{2m}$.

Finally, we can deduce that

$$p_{\mathcal{U}_0} = 1 - \sum_{i \in [m]} p_{\mathcal{U}_i} \geq 1 - \frac{1}{2} > 0. \quad (56)$$

That is, the randomized policy must place positive weight on policies corresponding to assignment strategies that satisfy every clause for every universal variable assignment. However, this fact holds only if there exists a policy in $\mathcal{U}_0$, which implies the satisfiability of the QSAT instance. $\square$

REMARK 1. In the proof of Proposition 6.2, the rationals used in the observation function have numerators and denominators that grow exponentially in the size of the problem, as we depict in Figure 7. We remark that one can prove the same result while keeping the values of the numerators and denominators polynomially bounded, and we sketch this argument below. In particular, we discuss a procedure to build a ME-POMDP $(\mathcal{M}'_i)_{i \in [m]}$ such that $\max_{\pi} V_{(\mathcal{M}'_i)_{i \in [m]}}^{\pi} \geq 0$ if and only if the original QSAT instance is satisfiable, and all the revelation probabilities in $(\mathcal{M}'_i)_{i \in [m]}$ have numerators and denominators bounded by $2n + 2m$.

We first distribute the revelation probability across multiple copies of the coin flip states in the manner depicted in Figure 8. For each state of the form $s_{T_j^{(\cdot)}}$ or $s_{F_j^{(\cdot)}}$ in the original ME-POMDP, we make $n + 1$ copies, so that the agent stays in such a state for $n + 1$ time steps. In particular, if variable x_j appears positively in clause i , then, in environment i , the first copy of $s_{T_j^{(1)}}/s_{T_j^{(-1)}}$ is revealed with probability $1 - \frac{1}{2m}$, and the remaining n copies are revealed with probability $1 - \frac{1}{8}$. With this revelation structure, the probability that the agent sees a copy of the state at least once is $1 - \frac{1}{2m} \frac{1}{8^n}$. Similar reasoning holds for $s_{F_j^{(1)}}/s_{F_j^{(-1)}}$. Meanwhile, if variable x_j does not appear positively in clause i , then each of the $n + 1$ copies of $s_{T_j^{(1)}}/s_{T_j^{(-1)}}$ is revealed with probability $\frac{1}{2(n+1)}$, so that the probability of the agent seeing the state is

$$1 - \left(1 - \frac{1}{2(n+1)} \right)^{n+1} \leq \frac{1}{2}. \quad (57)$$

The above inequality follows from the fact that $(1 - x)^r \geq 1 - rx$ for any $x \in [0, 1], r \in \mathbb{N}$. Ultimately, this splitting procedure lets us simulate the reduction from Proposition 6.2 without requiring small probabilities.

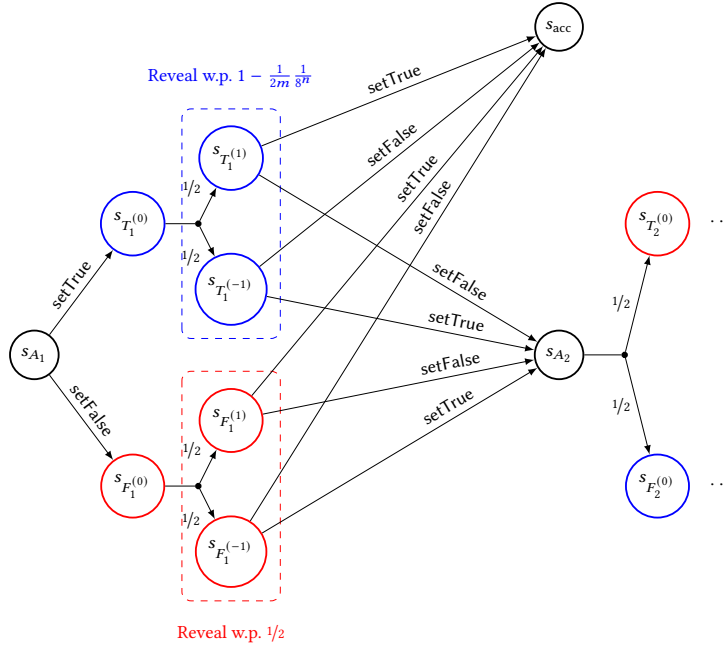


Figure 7: A fragment of an environment corresponding to the clause $(x_1 \vee \neg x_2)$ for the MO-SOMDP constructed in the proof of Proposition 6.2. Lines leaving nodes correspond to actions, and small dots encode stochastic resolution of the transition dynamics. When an arrow leaves a node without a label, then all actions induce the same dynamics. Blue nodes correspond to satisfaction of the clause, while red nodes correspond to the clause not being satisfied yet.

To handle the small threshold value v , we also add the component shown in Figure 9 to the start of the POMDP. In particular, the agent starts in this component in every environment, which guarantees it an expected reward of at least $\frac{1}{2m} \frac{1}{8^n}$. Adding this component to the POMDP then allows us to use 0 as the threshold for the MO-SOMDP decision problem.

We now argue that this new MO-SOMDP $(M'_i)_{i \in [m]}$ has non-negative reward if and only if the original QSAT instance is true. The $(\implies)$ direction follows the exact same logic as the proof of Proposition 6.2. Meanwhile, the $(\impliedby)$ direction requires the following modifications.

- When defining the mapping ξ we need to replace $\perp$ with $\underbrace{\perp, \dots, \perp}_{n+1 \text{ times}}$.
- When reasoning about the probability of reaching the dead state, we need to observe that the probability of observing the sequence $\underbrace{\perp, \dots, \perp}_{n+1 \text{ times}}$ is $\left(1 - \frac{1}{2(n+1)}\right)^{n+1} \geq \frac{1}{2}$ at each stage.

F.3 Multi-Observation Intermittently-Observable MDPs with Fixed Support

PROPOSITION 6.3. *The value problem for MO-IOMDPs with fixed support is in NP.*

PROOF. First, consider the case where one of the IOMDPs is unobservable, i.e., $\lambda = 0$. In this case, for all states s , $\text{supp}(O_i(s)) = \{\perp_i\}$ where $\perp_i \in Z$ is the blind observation in environment i . As such, by the fixed support condition, there exists some observation $\perp$ such that

$$O_i(s) = \delta_{\perp} \quad \forall i \in [n], s \in S. \quad (58)$$

That is, a fixed support MO-IOMDP where one environment is unobservable is equivalent to a single BlindMDP. For the remainder of the proof, we assume that the revelation probability is non-zero in all environments.

We first argue that each state's unique observation is the same across environments. As each environment i is an IOMDP with non-zero revelation probability, we can write the support of the observation function at any state s as

$$\text{supp}(O_i(s)) = \{z_i^s, \perp_i\}, \quad (59)$$

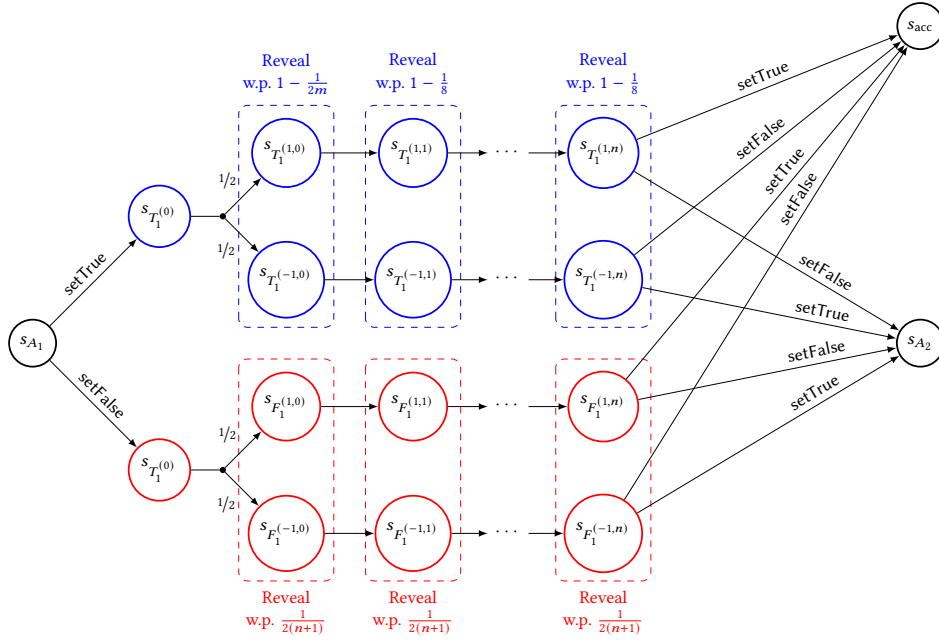


Figure 8: A fragment of an environment in the MO-SOMDP constructed during the “splitting” procedure described in Remark 1.

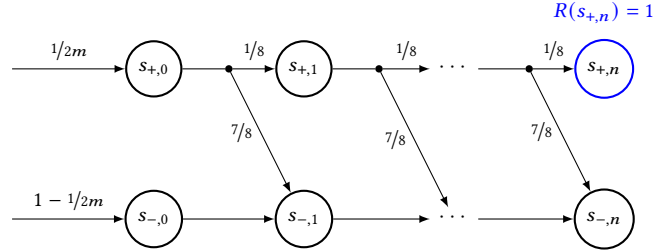


Figure 9: A component one can add to the start of a POMDP to increase the reward by $\frac{1}{2m} \frac{1}{8^n}$. The agent gets a reward of 1 with small probability. From the states in the set $\{s_{-,n}, s_{+,n}\}$, the agent transitions to the component shown in Figure 8.

where z_i^s denotes the unique observation of state s in environment i , and $\perp_i$ denotes the no-revelation observation in environment i . By the fixed-support condition, for any state s and pair of environments i, j , $\{z_i^s, \perp_i\} = \{z_j^s, \perp_j\}$. Thus, we can deduce that

$$\{\perp_i\} = \bigcap_{s \in S} \{z_i^s, \perp_i\} = \bigcap_{s \in S} \{z_j^s, \perp_j\} = \{\perp_j\}, \quad (60)$$

i.e., the no-revelation observation is the same in all environments. The first and last equalities follow from the fact that $\perp_i$ is the only observation in environment i that is common to all states, while the middle equality follows from the fixed-support condition. Since the blind observation is common to all environments, we can then deduce also that $z_i^s = z_j^s$, i.e., each state’s unique observation is constant across the environment.

We can now apply the results of [21] which describe how the value of a policy in an IOMDP changes as we increase the revelation rate, without shuffling observations. In particular, we first recall the following monotonicity result from [21] for IOMDPs.

LEMMA F.1 (ADAPTED FROM COROLLARY 1 IN [21]). *Fix an underlying MDP $\mathcal{M}'$, and let $\mathcal{M}_\lambda$ denote the IOMDP constructed from $\mathcal{M}'$ with revelation probability λ . Additionally, let π_λ be an optimal policy for the belief-MDP corresponding to $\mathcal{M}_\lambda$. Finally, for an arbitrary policy π and revelation probability v , let V_v^π be the value of policy π in IOMDP $\mathcal{M}_v$. Then, for any $u \geq \lambda$, we have*

$$V_\lambda^{\pi_\lambda} \leq V_u^{\pi_\lambda}.$$

In words, Lemma F.1 shows that there exists an optimal policy π_λ for $\mathcal{M}_\lambda$ such that increasing the revelation probability above λ will increase the value that π_λ attains. While the result in [21] is given for infinite-horizon IOMDPs, one can easily apply it to finite-horizon IOMDPs as well. For example, one can encode the finite-horizon IOMDP as an infinite-horizon IOMDP where

Algorithm 1 MO-IOMDP

Require: IOMDPs $\mathcal{M}_1, \dots, \mathcal{M}_n$ that differ in revelation probability λ , along with a value threshold B .

- 1: Determine the index i^* of the IOMDP with the lowest revelation rate.
 - 2: Non-deterministically guess a policy π in the class of optimal IOMDP policies
 - 3: **if** $\mathbb{E}_{\mathcal{M}_i}^\pi [\sum_{t=0}^{H-1} R(s_t, a_t)] \geq B$ **then return YES**
 - 4: **end if**
-

- States are augmented to include a time component,
- The reward corrects the discount factor,
- After the agent has taken H actions, the infinite-horizon IOMDP transitions to a dead state with zero reward.

We next argue that we can decide the MO-IOMDP problem in non-deterministic polynomial time.

We first upper bound the value of a given MO-IOMDP instance. Without loss of generality, we assume that the IOMDPs are ordered such that $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ where λ_i is the revelation probability for $\mathcal{M}_i$. By Theorem 1 in [21], the values of these IOMDPs reflect their revelation probabilities. That is,

$$V_{\mathcal{M}_1}^* \leq V_{\mathcal{M}_2}^* \leq \dots \quad (61)$$

We can then easily upper bound the MO-IOMDP value by $V_{\mathcal{M}_1}$. Indeed, for any policy π , we have

$$\min_i V_{\mathcal{M}_i}^\pi \leq \min_i V_{\mathcal{M}_i}^* = V_{\mathcal{M}_1}^*. \quad (62)$$

We now observe that an optimal belief-based policy for $\mathcal{M}_1$ achieves this value. Indeed, let π_1 be an optimal belief-based policy in $\mathcal{M}_1$. Then, by Lemma F.1, we have

$$\min_i V_{\mathcal{M}_i}^{\pi_1} \geq \min_i V_{\mathcal{M}_i}^{\pi_1} = V_{\mathcal{M}_1}^*. \quad (63)$$

Thus, a MO-IOMDP with fixed support has value at least B if and only if there exists a policy for the environment with the lowest revelation probability with value at least B , and this fact suggests Algorithm 1 as a simple method for solving the value problem in MO-IOMDPs. In short, we can guess and evaluate a policy in the class of optimal IOMDP policies, and by the proof of Proposition 4.2, we can implement this procedure in non-deterministic polynomial time. $\square$

G Proofs for Section 7

G.1 Time-Dependent Markovian Policies in ME-POMDPs

PROPOSITION 7.1. *The time-dependent Markovian policy problem for deterministic ME-BlindMDPs is $\exists\mathbb{R}$ -hard.*

PROOF. Fix a tensor $(a_{ijk})_{i \in [m], j \in [n], k \in [n]}$. This tensor will encode a 2-vs.-1 game, where the adversary chooses action $i \in [m]$ and two players choose actions j and k independently according to a pair of mixed strategies.

We build a ME-BlindMDP $(\mathcal{M}_i)_{i \in [n]}$ such that

$$\max_{\pi \in \Pi_{MR}} \min_{i \in [m]} V_{\mathcal{M}_i}^\pi = \max_{\mathbf{x} \in \Delta([n]), \mathbf{y} \in \Delta([n])} \min_{i \in [m]} \sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k] \quad (64)$$

In particular, we define a deterministic BlindMDP $\mathcal{M}_i$ as

$$\mathcal{M}_i = (\underbrace{\{s_0\} \cup [n]}_S, \underbrace{[n]}_A, \underbrace{\{\perp\}}_Z, T, O, R_i, \underbrace{\delta_{s_0}}_b, \underbrace{2}_H), \quad (65)$$

where we set T and O to satisfy

$$T(s_0, j) = j \quad \forall j \in [n], \quad O(s) = \perp \quad \forall s \in \{s_0\} \cup [n], \quad (66)$$

and we set R_i to satisfy

$$R_i(s_0, j) = 0 \quad \forall i \in [m], j \in [n], \quad (67)$$

$$R_i(j, k) = a_{ijk} \quad \forall i \in [m], j \in [n], k \in [n]. \quad (68)$$

A time-dependent Markovian policy in this ME-BlindMDP comprises a pair π_0 and π_1 of distributions in $\Delta([n])$, and this policy will have the following value.

$$\min_{i \in [m]} \mathbb{E}_{j \sim \pi_0, k \sim \pi_1, s \sim T(s_0, j)} [R_i(s, k)] = \min_{i \in [m]} \mathbb{E}_{j \sim \pi_0, k \sim \pi_1} [R_i(j, k)] = \min_{i \in [m]} \sum_{j \in [n], k \in [n]} a_{ijk} \pi_0[j] \pi_1[k]. \quad (69)$$

Thus, for any fixed rational B , the ME-POMDP has value at least B in time-dependent Markovian policies if and only if the maxmin value of the 2-vs.-1 team game that a_{ijk} defines is greater than B . The $\exists\mathbb{R}$ -hardness of the latter problem is then given by Lemma 24 in [17]. Alternatively, one can deduce the $\exists\mathbb{R}$ -hardness of the 2-vs.-1 game problem by observing that deciding the existence of a solution $\mathbf{x}, \mathbf{y} \in \Delta([n])$ to the system of equations

$$\sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k] = 0 \quad \forall i \in [m], \quad (70)$$

is $\exists\mathbb{R}$ -hard, as detailed in [27]. $\square$

PROPOSITION 7.2. *The time-dependent Markovian policy problem for deterministic multi-environment MDPs is $\exists\mathbb{R}$ -hard.*

PROOF. Fix a tensor $(a_{ijk})_{i \in [m], j \in [n], k \in [n]}$. This tensor will encode a 2-vs.-1 game, where the adversary chooses action $i \in [m]$ and two players choose actions j and k independently according to a pair of mixed strategies. Additionally let $\mathcal{L} = \{0, \dots, n-1\}$ and $\mathcal{I} = [m]$.

We build a ME-MDP $(\mathcal{M}_{(i,l)})_{(i,l) \in \mathcal{I} \times \mathcal{L}}$ such that

$$\max_{\pi \in \Pi_{MR}} \min_{(i,l) \in \mathcal{I} \times \mathcal{L}} V_{\mathcal{M}_{(i,l)}}^\pi = \max_{\mathbf{x} \in \Delta([n]), \mathbf{y} \in \Delta([n])} \min_{i \in [m]} \sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k]. \quad (71)$$

In particular, we define a deterministic ME-MDP $(\mathcal{M}_{(i,l)})_{(i,l) \in \mathcal{I} \times \mathcal{L}}$ with

$$\mathcal{M}_{(i,l)} = (\underbrace{\{s_0\} \cup \mathcal{L}}_S, \underbrace{[n]}_A, T_l, R_{(i,l)}, \underbrace{\delta_{s_0}}_b, \underbrace{2}_H). \quad (72)$$

We use the trivial observation set and function, where each state emits a unique observation deterministically. We set T_l to satisfy

$$T_l(s_0, j) = ((j-1+l) \bmod n) \quad \forall j \in [n], \quad (73)$$

so that the state at $t = 1$ encodes a shuffled version of the agent's initial action. We then set the reward function to encode the tensor $(a_{ijk})_{i \in [m], j \in [n], k \in [n]}$.

$$\begin{aligned} R_{(i,l)}(s_0, j) &= 0 & \forall i \in [m], l \in \mathcal{L}, j \in [n], \\ R_{(i,l)}(j', k) &= a_{i((j'-l) \bmod n + 1)k} & \forall i \in [m], j' \in \mathcal{L}, l \in \mathcal{L}, k \in [n]. \end{aligned}$$

We remark that, due to the above transition and reward structure, the agent gets a_{ijk} when it plays actions j and k in environment i . Indeed, for any $i \in [m]$, $j, k \in [n]$, and $l \in \mathcal{L}$, we have

$$R_{(i,l)}((j-1+l) \bmod n, k) = a_{ijk} \quad (74)$$

as, by basic modular arithmetic rules,

$$((j-1+l) \bmod n) - l \bmod n + 1 = j.$$

We now prove (71), and we first fix notation. A time-dependent Markovian policy in this MEMDP comprises two mappings π_0 and π_1 such that

$$\pi_t : \{s_0\} \cup \mathcal{L} \rightarrow \Delta([n]) \text{ for } t \in \{0, 1\}. \quad (75)$$

By an abuse of notation, we will identify π_0 with $\pi_0(s_0)$ since the first state is always s_0 . We will then use $\pi_1^{(j')}$ to denote $\pi_1(j') \in \Delta([n])$ for $j' \in \mathcal{L}$.

To show the $\leq$ direction of (71), we show that nature can uniformly randomize over the $\mathcal{L}$ component of the environment to ensure that the agent can not use state information to improve their reward. Fix a time-dependent Markovian randomized policy (π_0, π_1) , and define $\tilde{\pi}_1 \in \Delta([n])$ as follows.

$$\tilde{\pi}_1 = \sum_{j' \in \mathcal{L}} \frac{1}{n} \pi_1^{(j')}. \quad (76)$$

We next evaluate the policy (π_0, π_1) when nature uniformly randomizes over $\mathcal{L}$. That is, we evaluate $\sum_{l \in \mathcal{L}} \frac{1}{n} V_{(i,l)}^\pi$ as follows.

$$\begin{aligned} \sum_{l \in \mathcal{L}} \frac{1}{n} \mathbb{E}_{j \sim \pi_0, j' \sim T_l(s_0, j), k \sim \pi_1^{(j')}} [R_{(i,l)}(j', k)] &= \sum_{l \in \mathcal{L}} \frac{1}{n} \mathbb{E}_{j \sim \pi_0, k \sim \pi_1^{((j-1+l) \bmod n)}} [R_{(i,l)}((j-1+l) \bmod n, k)] \\ &= \sum_{l \in \mathcal{L}} \frac{1}{n} \mathbb{E}_{j \sim \pi_0, k \sim \pi_1^{((j-1+l) \bmod n)}} [a_{ijk}] \\ &= \sum_{j \in [n]} \pi_0[j] \sum_{k \in [n]} \sum_{l \in \mathcal{L}} \frac{1}{n} \pi_1^{((j-1+l) \bmod n)}[k] a_{ijk} = \sum_{j \in [n]} \pi_0[j] \sum_{k \in [n]} \tilde{\pi}_1[k] a_{ijk}. \end{aligned} \quad (77)$$

The first equality above uses the transition dynamics definition. The second equality uses (74). The third equality makes the expectation explicit and reorders the sums. The final equality uses the definition of $\tilde{\pi}_1$. Obviously, nature can do better than uniformly randomizing over $\mathcal{L}$ by choosing the worst value of l , and so we have

$$\min_{i \in [m]} \min_{l \in \mathcal{L}} V_{(i,l)}^\pi \leq \min_{i \in [m]} \sum_{l \in \mathcal{L}} \frac{1}{n} V_{(i,l)}^\pi = \min_{i \in [m]} \sum_{j \in [n], k \in [n]} \pi_0[j] \tilde{\pi}_1[k] a_{ijk} \leq \max_{\mathbf{x} \in \Delta([n]), \mathbf{y} \in \Delta([n])} \min_{i \in [m]} \sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k]. \quad (78)$$

The $\leq$ of (71) follows by observing that (78) holds for any time-dependent Markovian policy π .

We now deduce the $\geq$ direction of (71). Fix $\mathbf{x} \in \Delta([n])$, $\mathbf{y} \in \Delta([n])$. We define π_0 and π_1 such that

$$\pi_0[j] = \mathbf{x}[j] \quad \forall j \in [n] \quad \pi_1^{(j')}[k] = \mathbf{y}[k] \quad \forall k \in [n], \forall j' \in \mathcal{L}. \quad (79)$$

We then observe that, by following similar steps to the $(\leq)$ direction, that for any $i \in [m]$, $l \in \mathcal{L}$, we have

$$\mathbb{E}_{j \sim \pi_0, j' \sim T_l(s_0, j), k \sim \pi_1^{(j')}} [R_{(i,l)}(j', k)] = \mathbb{E}_{j \sim \mathbf{x}, k \sim \mathbf{y}} [R_{(i,l)}((j-1+l) \bmod n, k)] = \mathbb{E}_{j \sim \mathbf{x}, k \sim \mathbf{y}} [a_{ijk}] = \sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k],$$

from which we can deduce the $\geq$ direction of (71). $\square$

PROPOSITION 7.3. *The time-dependent Markovian policy problem for multi-observation POMDPs is $\exists\mathbb{R}$ -hard.*

PROOF. Fix a tensor $(a_{ijk})_{i \in [m], j \in [n], k \in [n]}$. We build a MO-POMDP such that

$$\max_{\pi \in \Pi_{\text{MR}}} \min_{i \in [m]} V_{\mathcal{M}_i}^\pi \geq 0 \iff \max_{\mathbf{x} \in \Delta([n]), \mathbf{y} \in \Delta([n])} \min_{i \in [m]} \sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k] \geq 0 \quad (80)$$

As in the proofs of the preceding propositions, we build an environment where the agent must choose their first two actions independently. However, we must overcome the fact that the adversary can not directly edit the environment.

To overcome this difficulty, we build an environment in which the adversary reveals information to the agent conditional on the agent taking a certain action. In particular, after the agent has chosen their first two actions, an environment-dependent observation is emitted, and the agent then chooses an action a^* as a function of this observation. Following this stage, the agent plays a guessing game in which they must guess the value of a coin flip. If the agent chose the right value for a^* , the agent sees the true coin value, whereas if the agent chose the wrong action, they see nothing, and the agent suffers a large penalty. As such, the adversarial nature can control the environment dynamics by incentivizing the agent to choose certain actions conditional on the environment.

We now formally define the MO-POMDP

$$\mathcal{M}_i = (S, A, Z, T, O_i, R, b, H).$$

We set the state space to have the following structure.

$$\begin{aligned} S = & \{s_0\} \\ & \cup ([n]) \\ & \cup ([n] \times [n]) \\ & \cup ([n] \times [n] \times [m]) \\ & \cup ([n] \times [n] \times [m] \times \{\perp, \top\}) \end{aligned}$$

We then set the action space to

$$A = [n] \cup [m] \cup \{1, 2\}, \quad (81)$$

and the observation space to

$$Z = \{z_j | j \in [m]\} \cup \{z_\top, z_\perp\} \cup \{z_{\text{blind}}\}. \quad (82)$$

We set $b = \delta_{s_0}$ so that the environment starts in s_0 , and we set $H = 5$.

We design the transition functions to track the actions taken in each time step, and additionally encode a coin flip at the last stage.

$$\begin{aligned} T(s_0, j) &= \begin{cases} j & \text{if } j \in [n] \\ 1 & \text{otherwise} \end{cases} \\ T(j, k) &= \begin{cases} (j, k) & \text{if } k \in [n] \\ (j, 1) & \text{otherwise} \end{cases} \\ T((j, k), i) &= \begin{cases} (j, k, i) & \text{if } i \in [m] \\ (j, k, 1) & \text{otherwise} \end{cases} \\ T((j, k, i), \tilde{a}) &= \begin{cases} (j, k, i, \top) \mapsto \frac{1}{2}, (j, k, i, \perp) \mapsto \frac{1}{2} \end{cases} \quad \forall \tilde{a} \in A. \end{aligned}$$

The “otherwise” cases effectively ensure that the agent always chooses integers from the right set at each stage.

We then design the observation function for environment i such that, at the third stage, the agent sees an observation z_i that is unique to that environment. The transition function definition also allows the state to encode whether or not the agent played action i in environment i . As such, we can define the observation function to reveal the coin flip result conditional on whether the agent played action i at the third stage. In particular, for any $i \in [m]$, we define the observation function O_i in environment i as follows.

$$\begin{aligned} O_i(s_0) &= z_{\text{blind}} \\ O_i(j) &= z_{\text{blind}} & \forall j \in [n] \\ O_i((j, k)) &= z_i & \forall j, k \in [n] \\ O_i((j, k, l)) &= z_{\text{blind}} & \forall j, k \in [n], l \in [m] \\ O_i((j, k, i, \top)) &= z_\top & \forall j, k \in [n] \\ O_i((j, k, i, \perp)) &= z_\perp & \forall j, k \in [n] \\ O_i((j, k, l, \top)) &= \begin{cases} z_\top \mapsto \frac{1}{2}, z_\perp \mapsto \frac{1}{2} \end{cases} & \forall j, k \in [n], l \in ([m] \setminus \{i\}) \\ O_i((j, k, l, \perp)) &= \begin{cases} z_\top \mapsto \frac{1}{2}, z_\perp \mapsto \frac{1}{2} \end{cases} & \forall j, k \in [n], l \in ([m] \setminus \{i\}) \end{aligned}$$

Note that, in all cases, the observation function still takes as input a *state*, but, for these POMDPs, the state may be a tuple of indices, hence the double-parentheses.

Finally, the rewards encode

- (1) The game defined by a_{ijk} , and
- (2) The rewards from the matching problem.

In particular, let $M = \max_{ijk} |a_{ijk}|$ represent the highest magnitude value in $(a_{ijk})_{i \in [m], j, k \in [n]}$. We set R as follows

$$\begin{aligned}
R(s_0, \tilde{a}) &= 0 & \forall \tilde{a} \in A \\
R(j, \tilde{a}) &= 0 & \forall j \in [n], \tilde{a} \in A \\
R((j, k), \tilde{a}) &= 0 & \forall j, k \in [n], \tilde{a} \in A \\
R((j, k, i), \tilde{a}) &= a_{ijk} & \forall j, k \in [n], i \in [m], \tilde{a} \in A \\
R((j, k, i, \top), \tilde{a}) &= \begin{cases} 0 & \text{if } \tilde{a} = 1 \\ -3M & \text{otherwise} \end{cases} & \forall j, k \in [n], i \in [m] \\
R((j, k, i, \perp), \tilde{a}) &= \begin{cases} 0 & \text{if } \tilde{a} = 2 \\ -3M & \text{otherwise} \end{cases} & \forall j, k \in [n], i \in [m]
\end{aligned}$$

We now argue that (80) holds.

($\implies$) A memoryless policy in this MO-POMDP comprises a sequence $(\pi_t)_{t \in \{0,1,2,3,4\}}$ of five mappings. Ignoring their behavior on observations that occur with zero probability, these policies take the form

$$\begin{aligned}
\pi_0 &: \{z_{\text{blind}}\} \rightarrow \Delta(A) \\
\pi_1 &: \{z_{\text{blind}}\} \rightarrow \Delta(A) \\
\pi_2 &: \{z_i\}_{i \in [m]} \rightarrow \Delta(A) \\
\pi_3 &: \{z_{\text{blind}}\} \rightarrow \Delta(A) \\
\pi_4 &: \{z_{\perp}, z_{\top}\} \rightarrow \Delta(A)
\end{aligned}$$

By an abuse of notation, we will identify $\pi_0(z_{\text{blind}})$ with π_0 and use similar notation for π_1 .

We assume that under these policies the agent receives a non-negative reward in all environments, and we prove that there exist $\mathbf{x}$ and $\mathbf{y}$ to certify that the right-hand side of (80) holds. In particular, we define the following vectors where $\mathbf{x}, \mathbf{y} \in \Delta([n])$, $\mathbf{v}^{(i)} \in \Delta([m])$ for each $i \in [m]$, and $\mathbf{w}_{\top}, \mathbf{w}_{\perp} \in \Delta(\{1, 2, 3\})$.

$$\begin{aligned}
\mathbf{x}[j] &= \begin{cases} \pi_0[1] + \sum_{j' \in A \setminus [n]} \pi_0[j'] & \text{if } j = 1 \\ \pi_0[j] & \text{otherwise.} \end{cases} \\
\mathbf{y}[k] &= \begin{cases} \pi_1[1] + \sum_{k' \in A \setminus [n]} \pi_1[k'] & \text{if } k = 1 \\ \pi_1[k] & \text{otherwise.} \end{cases} \\
\mathbf{v}^{(i)}[i'] &= \begin{cases} \pi_2(z_i)[1] + \sum_{l \in A \setminus [m]} \pi_2(z_i)[l] & \text{if } i' = 1 \\ \pi_2(z_i)[i'] & \text{otherwise.} \end{cases} \\
\mathbf{w}_{\top}[1] &= \pi_4(z_{\top})[1] & \mathbf{w}_{\perp}[1] &= \pi_4(z_{\perp})[1] \\
\mathbf{w}_{\top}[2] &= \pi_4(z_{\top})[2] & \mathbf{w}_{\perp}[2] &= \pi_4(z_{\perp})[2] \\
\mathbf{w}_{\top}[3] &= \sum_{l \in A \setminus \{1,2\}} \pi_4(z_{\top})[l] & \mathbf{w}_{\perp}[3] &= \sum_{l \in A \setminus \{1,2\}} \pi_4(z_{\perp})[l]
\end{aligned}$$

We now fix an index $i \in [m]$, and argue that

$$\sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k] \geq 0. \quad (83)$$

In particular, we evaluate the reward in each environment. When evaluating the reward, we can ignore the state at timesteps 0, 1, and 2. The expected reward in environment i at time 3 is given by

$$\mathbb{E}_{\mathcal{M}_i}^{\pi} [R(s_3, a_3)] = \sum_{i' \in [m]} \mathbf{v}^{(i)}[i'] \sum_{j \in [n], k \in [n]} a_{i'jk} \mathbf{x}[j] \mathbf{y}[k]. \quad (84)$$

Indeed, the probability of visiting state (j, k, i') at $t = 3$ is $\mathbf{x}[j] \mathbf{y}[k] \mathbf{v}^{(i)}[i']$ by definition of the transition dynamics. We can then upper bound this step's reward as follows.

$$\begin{aligned}
\mathbb{E}_{\mathcal{M}_i}^{\pi} [R(s_3, a_3)] &= \mathbf{v}^{(i)}[i] \sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k] + \sum_{i' \in [m] \setminus \{i\}} \mathbf{v}^{(i)}[i'] \sum_{j \in [n], k \in [n]} a_{i'jk} \mathbf{x}[j] \mathbf{y}[k] \\
&\leq \mathbf{v}^{(i)}[i] \sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k] + \sum_{i' \in [m] \setminus \{i\}} \mathbf{v}^{(i)}[i'] M \\
&= \mathbf{v}^{(i)}[i] \sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k] + (1 - \mathbf{v}^{(i)}[i])M \quad (85)
\end{aligned}$$

In words, the above inequality chain upper bounds the benefit the agent can get in its reward at $t = 3$ by not matching the environment-specific observation.

The expected reward at time 4 is given by

$$\begin{aligned} \mathbb{E}_{\mathcal{M}_i}^\pi [R(s_4, a_4)] &= \mathbf{v}^{(i)}[i] \left(\frac{1}{2}(1 - \mathbf{w}_\top[1]) + \frac{1}{2}(1 - \mathbf{w}_\top[2]) \right) (-3M) \\ &\quad + (1 - \mathbf{v}^{(i)}[i]) \left(\frac{1}{4}(1 - \mathbf{w}_\top[1]) + \frac{1}{4}(1 - \mathbf{w}_\perp[1]) + \frac{1}{4}(1 - \mathbf{w}_\top[2]) + \frac{1}{4}(1 - \mathbf{w}_\perp[2]) \right) (-3M). \end{aligned}$$

Indeed, the first component of the sum handles the case where the agent chooses i to match the environment, and the observation reveals the final state. Meanwhile, the second component handles the case where the observation provides no information about the final state.

We can then upper bound the expected reward at time 4 as follows.

$$\mathbb{E}_{\mathcal{M}_i}^\pi [R(s_4, a_4)] \leq (1 - \mathbf{v}^{(i)}[i]) \left(\frac{1}{4}(1 - \mathbf{w}_\top[1]) + \frac{1}{4}(1 - \mathbf{w}_\perp[1]) + \frac{1}{4}(1 - \mathbf{w}_\top[2]) + \frac{1}{4}(1 - \mathbf{w}_\perp[2]) \right) (-3M) \leq -\frac{3M}{2}(1 - \mathbf{v}^{(i)}[i])$$

Note that, in the second inequality, we use the fact that $1 - \mathbf{w}_\top[1] + 1 - \mathbf{w}_\top[2] \geq 1$.

We now upper bound the total reward.

$$\mathbb{E}_{\mathcal{M}_i}^\pi [R(s_3, a_3) + R(s_4, a_4)] \leq \mathbf{v}^{(i)}[i] \sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k] - \frac{M}{2}(1 - \mathbf{v}^{(i)}[i]).$$

We can combine this upper bound with the assumption that the reward is non-negative in environment i to deduce that (83) holds. First, consider the case where $M = 0$. If $M = 0$, then $a_{ijk} = 0$ for all i, j, k and thus $\mathbf{x}$ and $\mathbf{y}$ trivially satisfy (83). Now, assume that $M > 0$, and observe that $\mathbf{v}^{(i)}[i]$ must be positive, as otherwise the expected reward is negative. Using the fact that the expected reward is non-negative, we deduce that

$$\mathbf{v}^{(i)}[i] \sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k] \geq \frac{M}{2}(1 - \mathbf{v}^{(i)}[i]) \geq 0. \quad (86)$$

Finally, using the fact that $\mathbf{v}^{(i)}[i]$ is positive, we conclude that $\sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k] \geq 0$.

($\Leftarrow$) We now assume that some $\mathbf{x}, \mathbf{y} \in \Delta([n])$ exist satisfying the inequalities $\sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k] \geq 0$ for all i in $[m]$. We then define the following policy

$$\begin{aligned} \pi_0(z_{\text{blind}}) &= \mathbf{x} \\ \pi_1(z_{\text{blind}}) &= \mathbf{y} \\ \pi_2(z_i) &= \delta_i \\ \pi_4(z_\top) &= \delta_1 \\ \pi_4(z_\perp) &= \delta_2. \end{aligned} \quad (87)$$

Note that we abuse notation as $\pi_0(z_{\text{blind}}) \in \Delta(A)$ while $\mathbf{x} \in \Delta([n])$ and we may have $[n] \subsetneq A$. However, we can easily extend π_0 as required. We leave π_3 undefined as it does not affect the transition dynamics.

Fix an environment i . The reward at $t = 3$ satisfies

$$\mathbb{E}_{\mathcal{M}_i}^\pi [R(s_3, a_3)] = \sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k] \geq 0 \quad (88)$$

where the last inequality holds by assumption. Meanwhile, the reward at $t = 4$ is zero as $\mathbf{v}^{(i)}[i] = 1$, and $\mathbf{w}_\top[1] = \mathbf{w}_\perp[2] = 1$, where $\mathbf{w}$ and $\mathbf{v}^{(i)}$ are defined as above. $\square$

LEMMA 7.4. *The time-dependent Markovian policy problem for ME-POMDPs is in $\exists\mathbb{R}$.*

PROOF. A time-dependent Markovian policy that attains value at least B exists if and only if there exists a solution μ_i, v_i, π to the following system of bilinear inequalities and equalities.

$$\begin{aligned}
v_i(s, 0) &= b_i(s) & \forall s \in S, i \in [n], \\
\mu_i(s, a, t) &= v_i(s, t) \sum_{z \in Z} O_i(s)(z) \pi_t(z, a) & \forall i \in [n], s \in S, a \in A, t \in \{0, \dots, H-1\}, \\
v_i(s, t+1) &= \sum_{s' \in S, a \in A} \mu_i(s', a, t) T_i(s', a)(s) & \forall s \in S, t \in \{0, \dots, H-2\}, i \in [n], \\
\sum_a \pi_t(z, a) &= 1 & \forall z \in Z, t \in \{0, \dots, H-1\}, \\
\pi_t(z, a) &\geq 0 & \forall z \in Z, t \in \{0, \dots, H-1\}, a \in A, \\
\sum_{t \in \{0, \dots, H-1\}} \sum_{s \in S} \sum_{a \in A} \mu_i(s, a, t) R_i(s, a) &\geq B & \forall i \in [n].
\end{aligned} \tag{89}$$

For each i , the tensor $v \in \mathbb{R}^{S \times \{0, \dots, H-1\}}$ encodes the probability of visiting each state at each time, and $\mu \in \mathbb{R}^{S \times A \times \{0, \dots, H-1\}}$ encodes the probability of visiting each state and taking each action at each time. Meanwhile, $\pi_t(z, a)$ encodes the probability of taking action a upon seeing observation z at time t . Since we can express the value problem as such a system, inclusion in $\exists \mathbb{R}$ immediately follows. $\square$

G.2 Finite-State Controllers in ME-BlindMDPs

In this section, we prove the following theorem regarding the complexity of policy synthesis for FSCs.

THEOREM 7.6. *For any fixed rational $\alpha \in (0, 1)$, the α -FSC value problem for ME-BlindMDPs is $\exists \mathbb{R}$ -complete.*

We first give a series of Lemmas on the link between non-negative matrix factorization and ME-POMDPs. We then give the final proof of Theorem 7.6 in Section G.2.2.

G.2.1 Supporting results for Theorem 7.6.

DEFINITION G.1 (Non-negative matrix factorization [55]). A non-negative matrix $P \in \mathbb{R}^{m \times n}$ has a non-negative factorization with rank k if P can be written as the sum of k non-negative rank-one matrices.

LEMMA G.2. *A joint action distribution on (a_0, a_1) in a BlindMDP can be implemented by a $|Q|$ -state time-dependent FSC if and only if the distribution has a non-negative factorization with fewer than $|Q|$ elements.*

PROOF. In the case of a BlindMDP, the FSC update rule at time t is equivalent to two mappings

$$\gamma_t : Q \rightarrow \Delta(A) \quad \eta_t : Q \times A \rightarrow \Delta(Q) \tag{90}$$

($\implies$) Fix an FSC with $|Q|$ states. We can write the joint distribution of the actions (a_0, a_1) at the first two timesteps as follows

$$p(a_0, a_1) = \sum_{q_1 \in Q} \sum_{q_0 \in Q} p(q_0) p(a_0 | q_0) p(q_1 | a_0, q_0) p(a_1 | q_1, q_0, a_0) = \sum_{q_1 \in Q} \sum_{q_0 \in Q} p(q_0) p(a_0 | q_0) p(q_1 | a_0, q_0) p(a_1 | q_1).$$

Note that in the second equality, we use the fact that the distribution of the action a_1 only depends on the memory state q_1 at $t = 1$. Now, let $P[a_0, q_1]$ denote the sum $\sum_{q_0 \in Q} p(q_0) p(a_0 | q_0) p(q_1 | a_0, q_0)$, and let $W[a_1, q_1] = p(a_1 | q_1)$. Note that P and W encode two matrices in $\mathbb{R}^{|A| \times |Q|}$ with non-negative entries. We can now write the joint distribution as follows

$$p(a_0, a_1) = \sum_{q_1 \in Q} P[a_0, q_1] W[a_1, q_1] = (PW^\top)[a_0, a_1], \tag{91}$$

from which we can deduce that the joint distribution of a_0 and a_1 has a non-negative factorization with $|Q|$ components.

($\impliedby$) Conversely, let $P \in \mathbb{R}^{|A| \times |A|}$ be a non-negative matrix such that the entry-wise 1-norm satisfies $\|P\|_{1,1} = 1$ and P has a factorization

$$P = \sum_{l \in [k]} \mathbf{u}^l (\mathbf{v}^l)^\top, \tag{92}$$

where $\mathbf{u}^l$ and $\mathbf{v}^l$ are non-negative vectors for each $l \in [k]$. We can assume w.l.o.g. that each component matrix is non-zero so that $\|\mathbf{u}^l\|_1 > 0$ and $\|\mathbf{v}^l\|_1 > 0$. Next, observe that

$$1 = \|P\|_{1,1} = \sum_{i \in [|A|], j \in [|A|]} \left| \sum_{l \in [k]} \mathbf{u}^l[i] \mathbf{v}^l[j] \right| = \sum_{i \in [|A|], j \in [|A|]} \sum_{l \in [k]} |\mathbf{u}^l[i] \mathbf{v}^l[j]| = \sum_{l \in [k]} \|\mathbf{u}^l\|_1 \|\mathbf{v}^l\|_1$$

The first, second, and last equalities follow by definition. The third equality uses the non-negativity of the elements of $\mathbf{u}^l$ and $\mathbf{v}^l$. We can then write P as

$$P = \sum_{l \in [k]} \|\mathbf{u}^l\|_1 \|\mathbf{v}^l\|_1 \frac{\mathbf{u}^l}{\|\mathbf{u}^l\|_1} \frac{(\mathbf{v}^l)^\top}{\|\mathbf{v}^l\|_1}. \quad (93)$$

That is, we can write P as a convex combination of k non-negative rank 1 matrices $\mathbf{w}\mathbf{z}^\top$ where $\mathbf{w}$ and $\mathbf{z}$ are distributions. We can then implement this as the following FSC with $Q = [k]$. In particular, for each $l \in [k]$, we have

$$\begin{aligned} \iota(l) &= \|\mathbf{u}^l\|_1 \|\mathbf{v}^l\|_1, \\ \eta_0(l, a)(l') &= \begin{cases} 1 & l = l' \\ 0 & \text{otherwise} \end{cases} \quad \forall l' \in [k], a \in A, \\ \gamma_0(l) &= \frac{\mathbf{u}^l}{\|\mathbf{u}^l\|_1}, \\ \gamma_1(l) &= \frac{\mathbf{v}^l}{\|\mathbf{v}^l\|_1}. \end{aligned}$$

Note that we abuse notation slightly by implicitly identifying distributions on the action set with distributions on $[|A|]$. $\square$

We can prove the hardness of finding such a time-dependent FSC by appealing to non-negative factorization results. In particular, we will construct a ME-POMDP where any policy that gets a non-negative reward must have a joint distribution on a_0 and a_1 that satisfies a set of constraints, and due to the FSC constraint, this joint distribution will need to have low non-negative rank.

A naive route to prove $\exists\mathbb{R}$ -hardness would be to apply the $\exists\mathbb{R}$ -hardness of non-negative matrix factorization [50, 51]. However, this approach is insufficient and does not immediately lead to the desired result. A non-negative matrix factorization instance comprises a matrix W and a rank parameter r , and asks whether the matrix W has a non-negative factorization with r components. We can easily encode this decision problem as a ME-POMDP problem by designing a ME-POMDP instance such that the agent gets a non-negative reward if and only if each entry of the joint distribution on a_0 and a_1 matches an appropriate entry of $W/\|W\|_{1,1}$ where $\|W\|_{1,1}$ is the entry-wise 1-norm. Clearly, the matrix W has an r -component non-negative factorization if and only if there exists an r -memory-state FSC in the ME-POMDP that attains reward at least zero. However, this reduction does not provide information about the ratios between the number of memory-states $|Q|$ and the number of environments n for which the FSC-synthesis problem is $\exists\mathbb{R}$ -hard. In particular, we need to carefully examine the values of W and r for which the non-negative factorization problem is $\exists\mathbb{R}$ -hard to establish the ratios $|Q|/n$ for which the FSC-synthesis problem is $\exists\mathbb{R}$ -hard in ME-POMDPs.

Ultimately, adapting the $\exists\mathbb{R}$ -hardness proof in [42] for the related problem of *non-negative Gram feasibility* provides an easy route to carefully control the parameters of these W and r matrices. We now present this analysis.

We will reduce from the following ETR-complete problem.

Definition G.3 (ETR-INV [1, 2]). An ETR-INV instance comprises a set of variables $x_1, \dots, x_n$, and a set of constraints $C_1, \dots, C_m$, where each C_j is a constraint of the form

$$x + y = z, \quad x \cdot y = 1, \quad x = 1 \quad (94)$$

for $x, y, z \in \{x_1, \dots, x_n\}$. The ETR-INV problem asks if there exists a solution to these constraints with $x_i \in [1/2, 2]$.

We will also use the following ETR-complete satisfaction problem during the proof. The ETR with addition, multiplication, and inequalities (ETR-AMI) problem appears in [2] without the non-negativity constraints. However, we refer to the version used in [42], where all variables in the solution must be non-negative.

Definition G.4 (ETR-AMI). An ETR-AMI instance comprises a set of variables $x_1, \dots, x_n$, and a set of constraints $C_1, \dots, C_m$, where each C_j is a constraint of the following form

$$x + y = z, \quad x \cdot y = z, \quad x = 1 \quad (95)$$

for $x, y, z \in \{x_1, \dots, x_n\}$. The ETR-AMI problem asks if there exists a solution to these constraints with $x_i \in [0, \infty)$.

We first present the following lemma, which shows that we can convert an ETR-INV instance into an ETR-AMI instance with a solution in which the variables lie in the interval $[0, 2]$. This lemma is highly similar to those presented in [2], and we present it to smooth exposition.

LEMMA G.5 (ADAPTED FROM [2]). *Given an ETR-INV instance with n variables and m constraints, there exists an ETR-AMI instance such that*

- (1) *The ETR-AMI instance has a solution if and only if the ETR-INV instance has a solution.*
- (2) *The ETR-AMI instance has $3n + 3$ variables, and $3 + 2n + m$ constraints (ignoring the implicit non-negativity constraints).*
- (3) *The solution of the ETR-AMI instance satisfies $x_i \in [0, 2]$.*

PROOF. Let x_i be the variables of the original ETR-INV instance. For each x_i we add two auxiliary variables $\bar{x}_i$ and $\underline{x}_i$, along with variables $\llbracket 1 \rrbracket$, $\llbracket 2 \rrbracket$ and $\llbracket 1/2 \rrbracket$ to represent constants. We can then encode the original ETR-INV problem as the following ETR-AMI problem.

$$\begin{aligned}
 &\llbracket 1 \rrbracket = 1 \\
 &\llbracket 1 \rrbracket + \llbracket 1 \rrbracket = \llbracket 2 \rrbracket \\
 &\llbracket 2 \rrbracket \llbracket 1/2 \rrbracket = \llbracket 1 \rrbracket \\
 &x_i + \bar{x}_i = \llbracket 2 \rrbracket \quad \forall i \in [n] \\
 &x_i = \llbracket 1/2 \rrbracket + \underline{x}_i \quad \forall i \in [n] \\
 &(C_j)_{j \in [m]} \\
 &\text{All variables non-negative.}
 \end{aligned} \tag{96}$$

Note that we encode any constraint $xy = 1$ in the ETR-INV instance as $xy = \llbracket 1 \rrbracket$ in the ETR-AMI instance.

Clearly, any solution to the original ETR-INV instance defines a feasible solution to this ETR-AMI instance. The converse also holds, as any assignment to x_i in the ETR-AMI problem must satisfy $x_i \in [1/2, 2]$ and satisfy the constraints $(C_j)_{j \in [m]}$. $\square$

Following [42], we show that ETR-AMI problems can be converted to constrained non-negative matrix factorization. We provide a slightly different proof from [42] as

- (1) We don't constrain the factorization to be symmetric, and
- (2) We define constraints differently so that we can tightly control the number of factorization constraints we add.

LEMMA G.6 (ADAPTED FROM [42]). *Fix an integer $r \geq 2$, and an ETR-AMI instance with n variables and m constraints.*

There exists a pair of integers (f_A, f_B) , and two sets of matrices $A^{(1)}, \dots, A^{(f_A)}, B^{(1)}, \dots, B^{(f_B)}$ in $\mathbb{R}^{n+r, n+r}$ such that the ETR-AMI instance has a solution if and only if there exists a solution $W \in \mathbb{R}^{n+r, n+r}$ to

$$\langle A^{(i)}, W \rangle = 0 \quad \forall i \in [f_A] \tag{97}$$

$$\langle B^{(\kappa)}, W \rangle = 1 \quad \forall \kappa \in [f_B] \tag{98}$$

where W has a non-negative factorization with r components. Additionally, $f_A + f_B \leq r + n + m + 1$ and every entry in the matrices $(A^{(i)})_{i \in [f_A]}$ and $(B^{(\kappa)})_{\kappa \in [f_B]}$ is in $\{-1, 0, 1, 2\}$.

PROOF. We first add the constraints

$$W[i, j] = \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in [r], j \in [r]. \tag{99}$$

These constraints serve the same purpose as the anchor constraints in [42]. Indeed, suppose W has the following non-negative factorization, where $U_{(1)}, V_{(1)} \in \mathbb{R}^{r \times r}$, $U_{(2)} \in \mathbb{R}^{n \times r}$, and $V_{(2)} \in \mathbb{R}^{r \times n}$.

$$W = \begin{bmatrix} U_{(1)} \\ U_{(2)} \end{bmatrix} \begin{bmatrix} V_{(1)} & V_{(2)} \end{bmatrix}. \tag{100}$$

The above factorization implies that $U_{(1)}V_{(1)} = I$, where I is the $r \times r$ identity matrix. We can then use the following fact, which generalizes Lemma 5.5 in [42].

LEMMA G.7 ([23]). *If A is non-negative and has a non-negative inverse, then $A = DP$, where D is a diagonal matrix with positive diagonal entries, and P is a permutation matrix.*

We can thus write $U_{(1)}$ and $V_{(1)}$ in the following forms, where $(j_1, \dots, j_n)$ is a permutation of $[n]$ that encodes the permutation matrix P , and $a_1, \dots, a_n$ is a sequence of positive real numbers.

$$U_{(1)} = \begin{bmatrix} a_1 \mathbf{e}_{j_1}^\top \\ \vdots \\ a_n \mathbf{e}_{j_n}^\top \end{bmatrix} \quad V_{(1)} = \begin{bmatrix} \frac{1}{a_1} \mathbf{e}_{j_1} & \dots & \frac{1}{a_n} \mathbf{e}_{j_n} \end{bmatrix}. \tag{101}$$

We next add constraints to define the variables. In particular, we add

$$\begin{aligned}
 W[r+i, j] &= 0 \quad \forall j \in \{2, \dots, r\}, \forall i \in [n] \\
 W[j, r+i] &= 0 \quad \forall j \in \{2, \dots, r\}, \forall i \in [n] \\
 W[r+i, 1] &= W[1, r+i] \quad \forall i \in [n].
 \end{aligned} \tag{102}$$

The first constraint enforces that only the j_1^{th} column of $U_{(2)}$ be non-zero. Similarly, the second constraint enforces that only the j_1^{th} row of $V_{(2)}$ be non-zero. That, is $U_{(2)}$ and $V_{(2)}$ have the following forms in any valid factorization, where $(q_i)_{i \in [n]}$ and $(y_i)_{i \in [n]}$ are non-negative real numbers.

$$U_{(2)} = \begin{bmatrix} q_1 \mathbf{e}_{j_1}^\top \\ q_2 \mathbf{e}_{j_1}^\top \\ \vdots \\ q_n \mathbf{e}_{j_1}^\top \end{bmatrix} \quad V_{(2)} = \begin{bmatrix} y_1 \mathbf{e}_{j_1} & y_2 \mathbf{e}_{j_1} & \dots & y_n \mathbf{e}_{j_1} \end{bmatrix} \quad (103)$$

The final constraint in (102) enforces that, for each $i \in [n]$, we have

$$a_1 y_i = \langle a_1 \mathbf{e}_{j_1}, y_i \mathbf{e}_{j_1} \rangle = W[r + i, 1] = W[r + i, 1] = \left\langle \frac{1}{a_1} \mathbf{e}_{j_1}, q_i \mathbf{e}_{j_1} \right\rangle = \frac{1}{a_1} q_i. \quad (104)$$

With these constraints, we can then observe that, for any $i, j \in [n]$, we have

$$W[r + i, r + j] = \langle q_i \mathbf{e}_{j_1}, y_j \mathbf{e}_{j_1} \rangle = q_i y_j = \frac{1}{a_1} q_i a_1 y_j = a_1 y_i a_1 y_j. \quad (105)$$

Ultimately, the value $a_1 y_i = \frac{1}{a_1} q_i$ will represent variable x_i . The constraints in (102) take the same role as the constraints used in [42] to encode variables.

We can finally add the constraints to encode the ETR-AMI problem. In particular, as in [42], for each constraint of the form $x_i + x_j = x_k$, we add the constraint

$$W[r + i, 1] + W[r + j, 1] = W[r + k, 1]. \quad (106)$$

For any constraint of the form $x_i x_j = x_k$ we add

$$W[r + i, r + j] = W[r + k, 1]. \quad (107)$$

For a constraint of the form $x_i = 1$, we add

$$W[r + i, 1] = 1. \quad (108)$$

We have now specified all constraints.

We now count the number of constraints, omitting the entry-wise zero constraints. In particular, for the constraints in (99), we have r non-zero equality constraints for the diagonal entries. For the constraints in (102), we have n non-zero-entry equality constraints, one for each variable. Finally, for each of the m constraints in the ETR-AMI instance, we add a linear constraint on W . Overall, the following table summarises these constraints.

Source	Number of constraints
(99)	r
(102)	n
(106),(107),(108)	m

Each of the constraints counted above is linear, and we can represent it in one of the following forms.

$$\langle A^{(i)}, W \rangle = 0 \quad \langle B^{(\kappa)}, W \rangle = 1. \quad (109)$$

By inspecting the constraints, we can see that each entry in these matrices is in $\{-1, 0, 1, 2\}$. Additionally, let $S \subset [r + n] \times [r + n]$ denote the indices of matrix entries which we constrain to be zero in W , and let D^S be the indicator of these entries, i.e.,

$$D^S = \sum_{(i,j) \in S} \mathbf{e}_i \mathbf{e}_j^\top. \quad (110)$$

Since W is non-negative, we have $\langle D^S, W \rangle = 0$ if and only if $W[i, j] = 0$ for all $(i, j) \in S$. Indeed,

$$\langle D^S, W \rangle = 0 \iff \sum_{(i,j) \in S} W[i, j] = 0 \iff W[i, j] = 0 \quad \forall (i, j) \in S. \quad (111)$$

The second $\iff$ holds as W must be non-negative. Note also that D^S has binary entries. Thus, we can encode the above constraints with matrices $A^{(1)}, \dots, A^{(f_A)}, B^{(1)}, \dots, B^{(f_B)}$ so that $f_A + f_B \leq r + n + m + 1$.

We now argue that the ETR-AMI instance has a solution if and only if there exists a matrix W with an r -non-negative rank factorization that satisfies the specified constraints.

($\Leftarrow$) Suppose the constraints have a solution W with non-negative factorization UV such that $U \in \mathbb{R}^{n+r \times r}$ and $V \in \mathbb{R}^{r \times n+r}$. Now, set x_i to $W[r + i, 1] = a_1 y_i$ for each i , noting that, by assumption $a_1 y_i$ is non-negative. We argued earlier in the proof that $W[r + i, r + j] = a_1 y_i a_1 y_j$, so we can also deduce that $W[r + i, r + j] = x_i x_j$. We can now conclude that, since we imposed the constraints in (106), (107), and (108), the values $x_1, \dots, x_n$ must satisfy the constraints of the ETR-AMI instance. Thus, we can conclude that a solution to the ETR-AMI problem exists.

($\implies$) Suppose the ETR-AMI instance has a solution $x_1, \dots, x_n$, and let $\mathbf{x}$ denote the vector where $\mathbf{x}[i] = x_i$ for each $i \in [n]$. Then, the following matrix W is a solution to the factorization problem.

$$W = \begin{bmatrix} 1 & \mathbf{0}_{r-1}^\top \\ \mathbf{0}_{r-1} & I_{r-1 \times r-1} \\ \mathbf{x} & \mathbf{0}_{n \times r-1} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0}_{r-1}^\top & \mathbf{x}^\top \\ \mathbf{0}_{r-1} & I_{r-1 \times r-1} & \mathbf{0}_{r-1 \times n} \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0}_{r-1}^\top & \mathbf{x}^\top \\ \mathbf{0}_{r-1} & I_{r-1 \times r-1} & \mathbf{0}_{r-1 \times n} \\ \mathbf{x} & \mathbf{0}_{n \times r-1} & \mathbf{xx}^\top \end{bmatrix}$$

Indeed, W has a non-negative factorization with r components, and it trivially satisfies the constraints in (99) and (102). Furthermore, because $x_1, \dots, x_n$ represents a solution to the ETR-AMI problem, the constraints (106), (107) and (108) are satisfied. $\square$

We now connect Lemmas G.5 and G.6 to demonstrate that we can reduce ETR-INV to a non-negative factorization problem with an additional simplex constraint on the factorized matrix.

LEMMA G.8. *Given an ETR-INV instance with n variables and m constraints, and a fixed integer $r \geq 2$, there exists a sequence of matrices $C^1, \dots, C^{(f)}$ in $\mathbb{R}^{r+n' \times r+n'}$ such that the ETR-INV instance is feasible if and only if there exists a matrix $X \in \mathbb{R}^{r+n' \times r+n'}$ such that*

- X has non-negative entries,
- $\|X\|_{1,1} = 1$, i.e., the entries of X sum to 1,
- $\langle C^{(i)}, X \rangle \geq 0$ for all $i \in [f]$,
- X has a non-negative factorization with rank r ,

where $f \leq 5n + m + r + 10$ and $n' = 3n + 3$.

PROOF. Fix an ETR-INV instance with n variables and m constraints. Next, build an ETR-AMI instance with n' variables and m' constraints according to Lemma G.5, and build the associated matrices $A^{(1)}, \dots, A^{(f_A)}, B^{(1)}, \dots, B^{(f_B)}$ according to Lemma G.6. The ETR-AMI instance will have $n' := 3n+3$ variables, and $m' := 3+2n+m$ constraints, while f_A and f_B satisfy $f_A + f_B \leq n' + m' + r + 1 = 3n+3+3+2n+m+r+1 = 5n+m+r+7$.

Additionally, assume w.l.o.g. that $B^{(1)}$ encodes the constraint $W[1, 1] = 1$.

We build the following linear inequality system from the above matrices.

$$\begin{aligned} \langle A^{(i)} - A^{(i+1)}, X \rangle &\geq 0 & \forall i \in \{1, \dots, f_A - 1\} \\ \langle A^{(f_A)} - A^{(1)}, X \rangle &\geq 0 \\ \langle A^{(1)}, X \rangle &\geq 0 \\ \langle -A^{(1)}, X \rangle &\geq 0 \\ \langle B^{(\kappa)} - B^{(\kappa+1)}, X \rangle &\geq 0 & \forall \kappa \in \{1, \dots, f_B - 1\} \\ \langle B^{(f_B)} - B^{(1)}, X \rangle &\geq 0 \\ \left\langle B^{(1)} - \frac{1}{2(r+4n+4n^2)} \mathbf{1}\mathbf{1}^\top, X \right\rangle &\geq 0 \end{aligned} \tag{112}$$

Note that this system involves $f_A + f_B + 3$ inequalities of the form $\langle C, X \rangle \geq 0$.

We claim that (112) has a solution with X such that

- $X \geq 0$,
- $\|X\|_{1,1} = 1$, and
- X has a non-negative factorization with r components.

if and only if the original ETR-INV problem has a solution.

($\implies$) Suppose X is a solution to (112) with the desired properties. We immediately deduce that it must satisfy

$$\begin{aligned} \langle A^{(i)}, X \rangle &= 0 & \forall i \in \{1, \dots, f_A\} \\ \langle B^{(\kappa)}, X \rangle &= \nu & \forall \kappa \in \{1, \dots, f_B\} \end{aligned} \tag{113}$$

for some $\nu \in \mathbb{R}$. Additionally, the final constraint implies, along with the fact that $\|X\|_{1,1} = 1$, implies that

$$\langle B^{(1)}, X \rangle \geq \left\langle \frac{1}{2(r+4n+4n^2)} \mathbf{1}\mathbf{1}^\top, X \right\rangle = \frac{1}{2(r+4n+4n^2)} > 0, \tag{114}$$

which in turn implies that, in fact, $\nu > 0$. Now, define $\tilde{X} = X/\nu$. $\tilde{X}$ is now a matrix that satisfies (113) with $\nu = 1$, and $\tilde{X}$ also has an r -component non-negative factorization. However, by Lemma G.6 and Lemma G.5, we can now deduce that the original ETR-INV instance is feasible.

($\impliedby$) Suppose the ETR-INV instance has a solution. As a result, the constructed ETR-AMI instance has a solution $x_1, \dots, x_n$ where $x_i \in [0, 2]$ for every i by Lemma G.5. As in the proof of Lemma G.6, we can now deduce that the following matrix W is a solution to (113)

with $\nu = 1$, where we use $\mathbf{x}$ to denote the vector with the ETR-AMI solution as its elements.

$$W = \begin{bmatrix} 1 & \mathbf{0}_{r-1}^\top \\ \mathbf{0}_{r-1} & I_{r-1 \times r-1} \\ \mathbf{x} & \mathbf{0}_{n \times r-1} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0}_{r-1}^\top & \mathbf{x}^\top \\ \mathbf{0}_{r-1} & I_{r-1 \times r-1} & \mathbf{0}_{r-1 \times n} \\ \mathbf{x} & \mathbf{0}_{n \times r-1} & \mathbf{xx}^\top \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0}_{r-1}^\top & \mathbf{x}^\top \\ \mathbf{0}_{r-1} & I_{r-1 \times r-1} & \mathbf{0}_{r-1 \times n} \\ \mathbf{x} & \mathbf{0}_{n \times r-1} & \mathbf{xx}^\top \end{bmatrix}$$

We need to construct a scaling of W that lies in the simplex and satisfies (112). By counting the elements of W and using the bound on the value of the elements of $\mathbf{x}$, we have

$$\|W\|_{1,1} \leq \|I_{r \times r}\|_{1,1} + 2\|\mathbf{x}\|_1 + \|\mathbf{xx}^\top\|_{1,1} \leq r + 4n + 4n^2. \quad (115)$$

Also, note that $\|W\|_{1,1} > 0$ as W is non-zero. Thus, we can define $X = W/\|W\|_{1,1}$ and observe that

$$\langle B^{(1)}, X \rangle = X[1, 1] = 1/\|W\|_{1,1} \geq 1/(r + 4n + 4n^2) \geq 1/2(r + 4n + 4n^2).$$

Furthermore, by linearity, X satisfies (113) for some ν , from which it follows that X is a matrix that satisfies (112). Also, because W has an r -component non-negative factorization, so does X , and trivially, X satisfies $\|X\|_{1,1} = 1$. $\square$

G.2.2 Proof of Theorem 7.6. Fix a rational $\alpha \in (0, 1)$.

Fix an ETR-INV instance with n variables and m constraints, and construct an integer r such that

$$r = \left\lceil \frac{\alpha}{1-\alpha} (5n + m + 10) \right\rceil, \quad (116)$$

where, for a real number u , $\lceil u \rceil$ is the smallest integer greater than or equal to u . Note that, since α is fixed, r is bounded by $O(m + n)$, and the time to compute r is bounded by a polynomial in the representation size of α , which is fixed, and the size of the original ETR-INV instance.

Next, build the system of constraints $C^{(1)} \dots C^{(f)}$ specified in Lemma G.8, where $f \leq 5n + m + r + 10$. We will build a ME-POMDP with an environment for each $i \in [f]$.

Note that we have

$$r \geq \frac{\alpha}{1-\alpha} (5n + m + 10) \implies (1-\alpha)r \geq \alpha(5n + m + 10) \implies r \geq \alpha(5n + m + r + 10) \geq \alpha f. \quad (117)$$

From this stage, we follow the same construction as in the proof of Proposition 7.1. The only modifications we make are that

- The set of environments is $[f]$.
- In environment i , the reward at state (j, k) is $R_i(j, k) = C^{(i)}[j, k]$.

Note that, in environment i , the reward is

$$\mathbb{E}_{(j,k) \sim P^\pi} [R_i(j, k)] = \langle P^\pi, C^{(i)} \rangle, \quad (118)$$

where P^π is the joint distribution on a_0 and a_1 induced by the time-dependent finite-state controller π .

We now claim that the value of this ME-POMDP is non-negative if and only if the ETR-INV instance has a solution.

($\implies$) Suppose a time-dependent FSC policy π , with r memory states exists such that

$$\langle P^\pi, C^{(i)} \rangle \geq 0 \quad \forall i \in [f]. \quad (119)$$

By Lemma G.2, P^π is a joint-distribution matrix with an r -component non-negative factorization. Then, by Lemma G.8, the original ETR-INV instance has a solution.

($\impliedby$) If the ETR-INV instance has a solution, then by Lemma G.8, there exists a matrix X with an r -component non-negative factorization such that X has non-negative entries, $\|X\|_{1,1} = 1$, and

$$\langle X, C^{(i)} \rangle \geq 0 \quad \forall i \in [f]. \quad (120)$$

By Lemma G.2, there exists an r -state time-dependent FSC policy such that $P^\pi = X$, and thus there exists a policy π such that

$$\mathbb{E}_{(j,k) \sim P^\pi} [R_i(j, k)] \geq 0 \quad \forall i \in [f]. \quad (121)$$

Finally, we claim that the ME-POMDP

- (1) Lies in the specified α -decision class, and
- (2) Can be constructed in polynomial time from the given ETR-INV instance.

The constructed POMDP has f environments, and we look for an FSC with r states. It then follows from (117) that the ratio of memory-states to environments is greater than α .

For the second claim, we first note that the ETR-INV to ETR-AMI instance conversion can obviously be done in polynomial time. We next note that $r \leq 1 + \frac{\alpha}{1-\alpha} (5n + m + 10)$, which means the $A^{(i)}$ and $B^{(k)}$ matrices have polynomial size in the original ETR-INV instance size. We can clearly translate the ETR-AMI instance into these $A^{(i)}$ and $B^{(k)}$ in polynomial time, as writing each matrix requires at most looking up data from the ETR-AMI instance and translating it to these matrices as described in the proof of Lemma G.6. Building all but one of the $C^{(i)}$ matrices described in Lemma G.8 then simply requires basic addition and subtraction between some $A^{(i)}$ and $B^{(k)}$ matrices, all of which have values in $\{-1, 0, 1, 2\}$. The remaining $C^{(i)}$ matrix is of the form

$$\mathbf{e}_1 \mathbf{e}_1^\top - \frac{1}{2(r + 4n + 4n^2)} \mathbf{1} \mathbf{1}^\top, \quad (122)$$

which can clearly be built in polynomial time. Finally, once we have constructed the $C^{(i)}$ matrices, we can build the POMDP in polynomial time. Indeed, the POMDP has size defined by the following inequalities

- $|S| = 1 + r + n' \leq 1 + (3n + 3) + 1 + \frac{\alpha}{1-\alpha}(5n + m + 10)$.
- $|A| = |S| - 1$.
- $|Z| = 1$.
- The number of environments is $f \leq 5n + m + r + 10 \leq 5n + m + 11 + \frac{\alpha}{1-\alpha}(5n + m + 10)$.
- The number of memory states satisfies $r \leq f$.

We can clearly write the transition and observation functions in polynomial time, as all transitions/observation are deterministic. To encode the reward functions, we simply need to copy the $C^{(i)}$ matrices into the reward function as appropriate. $\square$

G.3 Finite-State Controllers in General ME-POMDPs

PROPOSITION 7.7. *For $\alpha = 1$, the α -FSC value problem for ME-POMDPs is $\exists\mathbb{R}$ -complete.*

PROOF. Fix a tensor $(a_{ijk})_{i \in [m], j \in [n], k \in [n]}$. Furthermore, assume that matrices come in pairs with opposite signs. That is,

$$\forall i \in [m] \exists i' \in [m] : a_{ijk} = -a_{i'jk} \quad \forall j \in [n], k \in [n]. \quad (123)$$

Indeed, this class of tensors can encode the problem of deciding if there exists a solution $\mathbf{x}, \mathbf{y} \in \Delta([n])$ such that

$$\sum_{j \in [n], k \in [n]} b_{ijk} \mathbf{x}[j] \mathbf{y}[k] = 0 \quad \forall i \in [m] \quad (124)$$

for some tensor b_{ijk} , and this problem is $\exists\mathbb{R}$ -complete [27].

We build a ME-POMDP $(\mathcal{M}_i)_{i \in [m]}$ such that there exists an FSC-based policy with non-negative reward in all environments if and only if there exists $\mathbf{x}, \mathbf{y} \in \Delta([n])$ such that

$$\sum_{j, k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k] \geq 0 \quad \forall i \in [m]. \quad (125)$$

In particular, we define POMDP $\mathcal{M}_i$, for $i \in [m]$, according to

$$\mathcal{M}_i = (S, \underbrace{[n] \cup [m]}_A, \underbrace{[m] \cup \{\perp\}}_Z, T, O, R_i, b, \underbrace{4}_H). \quad (126)$$

We define the state space as

$$S = \{s_0\} \cup ([n] \times [m] \times \{1, 2, 3\}). \quad (127)$$

We define the transition function in a similar manner to the proof of Proposition 7.2, but we additionally add a randomly drawn component of the environment. In particular, we define T as follows.

$$\begin{aligned} T(s_0, j) &= \left\{ (j, 1, 1) \mapsto \frac{1}{m}, \dots, (j, m, 1) \mapsto \frac{1}{m} \right\} & \forall j \in [n] \\ T(s_0, j) &= \left\{ (1, 1, 1) \mapsto \frac{1}{m}, \dots, (1, m, 1) \mapsto \frac{1}{m} \right\} & \forall j \notin [n] \\ T((j, l, 1), a) &= (j, l, 2) & \forall j \in [n], l \in [m], a \in A \\ T((j, l, 2), a) &= (j, l, 3) & \forall j \in [n], l \in [m], a \in A \end{aligned}$$

We then define the observation function such that the random component of the environment is perfectly observed at time 1, i.e.,

$$\begin{aligned} O(s_0) &= \perp \\ O((j, l, 1)) &= l \quad \forall j \in [n], l \in [m], \\ O((j, l, 2)) &= \perp \quad \forall j \in [n], l \in [m], \\ O((j, l, 3)) &= \perp \quad \forall j \in [n], l \in [m]. \end{aligned}$$

We define the reward function at states in $[n] \times [m] \times 2$ so that the agent must guess the random component of the state at $t = 2$,

$$R_i((j, l, 2), a) = \begin{cases} 0 & \text{if } a = l, \\ -1 & \text{otherwise,} \end{cases} \quad (128)$$

Meanwhile, the reward function at states in $[n] \times [m] \times 3$ encodes the tensor. That is

$$R_i((j, l, 3), k) = \begin{cases} a_{ijk} & \text{if } k \in [n], \\ a_{ij1} & \text{otherwise.} \end{cases} \quad (129)$$

We set rewards for all other states to be zero for all actions.

We now argue that this ME-POMDP has a time-dependent-FSC-based policy with m states if and only if there exists $\mathbf{x}, \mathbf{y} \in \Delta([n])$ such that (125) holds.

($\implies$) Fix a policy π . Since all environments share transition and observation dynamics, the joint distribution on the memory state q_2 at time 2, and the observation at time 1 is constant across the environments.

We define the matrix $P \in \mathbb{R}^{m \times m}$ to encode the distribution of q_2 conditioned on each possible observation at time 1. That is,

$$P[k, j] := \mathbb{P}_{\mathcal{M}_i}^\pi[q_2 = k | z_1 = j] \quad \forall k, j \in [m], \quad (130)$$

where the definition of P does not depend on the choice of environment i . Note that, by definition of the observations, the observation z_1 at time 1 is l if and only if the state at time 1 lies in the set $S_l := \{(j, l, 1) | j \in [n]\}$.

At time 2, the state lies in the set $[n] \times [m] \times \{2\}$, and so the observation is $\perp$. As such, the relationship between the action and memory-state at time 2 on any feasible path is determined by a matrix $F \in \mathbb{R}^{|A| \times m}$ where $F[k, l]$ is the probability of taking action k conditioned on the memory state being l at time 2. We use F_{trunc} to denote the matrix formed by restricting F to its first m rows, so that it becomes a square matrix.

With this notation fixed, we can then evaluate reward at time 2 in environment i . In particular, we have

$$\begin{aligned} \mathbb{E}_{\mathcal{M}_i}^\pi[R_i(s_2, a_2)] &= \sum_{j \in [n], l \in [m]} \sum_{k \in A} R_i((j, l, 2), k) \mathbb{P}_{\mathcal{M}_i}^\pi[s_2 = (j, l, 2), a_2 = k] \\ &= - \sum_{l \in [m]} \sum_{k \in A \setminus \{l\}} \mathbb{P}_{\mathcal{M}_i}^\pi[s_2 \in \{(j, l, 2) | j \in [n]\}, a_2 = k] \\ &= - \sum_{l \in [m]} \sum_{k \in A \setminus \{l\}} \mathbb{P}_{\mathcal{M}_i}^\pi[z_1 = l, a_2 = k] = - \sum_{l \in [m]} \sum_{k \in A \setminus \{l\}} \mathbb{P}_{\mathcal{M}_i}^\pi[a_2 = k | z_1 = l] \mathbb{P}_{\mathcal{M}_i}^\pi[z_1 = l] \end{aligned} \quad (131)$$

The first equality just involves expanding the expectation. The second equality follows by definition of the reward. The third equality applies the definition of the POMDP, and in particular, it applies the fact that the state is in $\{(j, l, 2) | j \in [n]\}$ at time 2 if and only if the first observation was l . The fourth equality just applies conditional probability rules.

We next argue that $\mathbb{P}_{\mathcal{M}_i}^\pi[a_2 = l | z_1 = l] = 1$ in all environments for all $l \in [m]$. We first note that the reward at time 2 is the same across environments, since the expression derived in (131) depends only on the probabilities of states and actions along the path. However, by the assumption encoded in (123), for any policy π , there exists an environment such that the reward at $t = 3$ is non-positive. Thus, for the policy π to have non-negative value, it must hold that $\mathbb{E}^\pi[R_i(s_2, a_2)] \geq 0$, where we now drop the $\mathcal{M}_i$ subscript to emphasize that values are constant across all environments for any policy π . We also note that for every $l \in [m]$, $\mathbb{P}[z_1 = l] = \frac{1}{m}$. Thus we can deduce from (131) that

$$\sum_{k \in A \setminus \{l\}} \mathbb{P}^\pi[a_2 = k | z_1 = l] = 0 \quad \forall l \in [m]. \quad (132)$$

The requirement that $a_2 = z_1$ then allows us to argue about the memory structure. Indeed, we have, for all $k, l \in [m]$,

$$\mathbb{P}[a_2 = k | z_1 = l] = \sum_{j \in [m]} F[k, j] P[j, l] = \begin{cases} 1 & k = l \\ 0 & \text{otherwise} \end{cases}, \quad (133)$$

which in turn implies that

$$F_{\text{trunc}} P = I_{m \times m}. \quad (134)$$

However, since F_{trunc} and P are non-negative, it follows from [23] that P is a permutation matrix. That is, the memory state at time 2 must be equal to the observation at time 1, up to some relabeling.

Since the agent observes only $\perp$ at times 2 and 3, the action a_3 at time 3 is independent of a_0 , conditioned on the memory state q_2 at time 2. Thus, we have

$$\mathbb{P}^\pi[a_3 = k, a_0 = j] = \sum_{l \in [m]} \mathbb{P}^\pi[a_3 = k | q_2 = l] \mathbb{P}^\pi[q_2 = l, a_0 = j], \quad (135)$$

where we continue to drop the $\mathcal{M}_i$ from the subscript of the probability measure to emphasize when properties are independent of the environment. However, we argued above that q_2 must be a permutation of the observation at time 1, which is independent of the action a_0 and has a uniform distribution. As such, we can rewrite the joint distribution of a_3 and a_0 as

$$\mathbb{P}^\pi[a_3 = k, a_0 = j] = \underbrace{\sum_{l \in [m]} \mathbb{P}^\pi[a_3 = k | q_2 = l]}_{=: g[k]} \frac{1}{m} \mathbb{P}^\pi[a_0 = j]. \quad (136)$$

In a similar manner to previous proofs, we can then define two vectors $\mathbf{x}, \mathbf{y} \in \Delta([n])$

$$\mathbf{x}[j] = \begin{cases} \mathbb{P}^\pi[a_0 = 1] + \sum_{j' \in A \setminus [n]} \mathbb{P}[a_0 = j'] & \text{if } j = 1, \\ \mathbb{P}^\pi[a_0 = j] & \text{otherwise.} \end{cases} \quad (137)$$

$$\mathbf{y}[k] = \begin{cases} g[1] + \sum_{k' \in A \setminus [n]} g[k'] & \text{if } k = 1, \\ g[k] & \text{otherwise.} \end{cases} \quad (138)$$

By definition of the ME-POMDP, the reward in environment i when $t = 3$ is equal to

$$\sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k]. \quad (139)$$

As we established earlier, the reward at $t = 2$ is zero. As such, for the ME-POMDP to have non-negative reward, the reward at $t = 3$ must be non-negative in all environments. However, we can then deduce the existence of the requisite $\mathbf{x}$ and $\mathbf{y}$.

($\Leftarrow$) Suppose there exists $\mathbf{x}$ and $\mathbf{y}$ such that

$$\sum_{j \in [n], k \in [n]} a_{ijk} \mathbf{x}[j] \mathbf{y}[k] \geq 0 \quad \forall i \in [m]. \quad (140)$$

Then, we can define a FSC-based policy π where

- The action at time 0 is drawn from $[n]$ according to $\mathbf{x}$ for any memory state.
- At time 1, the memory is updated to store the observation z_1 .
- At time 2, the policy plays the action corresponding to the stored observation.
- The action at time 3 is drawn from $[n]$ according to $\mathbf{y}$ for any memory state.

At time 2 we get zero reward in all POMDPs as we have $a_2 = z_1$. Meanwhile, at time 3 we have non-negative expected reward in all POMDPs as the joint distribution of a_0 and a_3 satisfies

$$\mathbb{P}[a_0 = j, a_3 = k] = \mathbf{x}[j] \mathbf{y}[k]. \quad (141)$$

□