Near Instance-Optimal PAC Reinforcement Learning for Deterministic MDPs

Andrea Tirinzoni, Aymen Al-Marjani, Emilie Kaufmann

Near Instance-Optimal PAC Reinforcement Learning for Deterministic MDPs

Andrea Tirinzoni[*]
Meta AI
Paris, France
tirinzoni@fb.com

Aymen Al-Marjani
UMPA, ENS Lyon
Lyon, France
aymen.al_marjani@ens-lyon.fr

Emilie Kaufmann
Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 - CRIStAL
Lille, France
emilie.kaufmann@univ-lille.fr

Abstract

In probably approximately correct (PAC) reinforcement learning (RL), an agent is required to identify an ε-optimal policy with probability $1-\delta$. While minimax optimal algorithms exist for this problem, its instance-dependent complexity remains elusive in episodic Markov decision processes (MDPs). In this paper, we propose the first nearly matching (up to a horizon squared factor and logarithmic terms) upper and lower bounds on the sample complexity of PAC RL in deterministic episodic MDPs with finite state and action spaces. In particular, our bounds feature a new notion of sub-optimality gap for state-action pairs that we call the deterministic return gap. While our instance-dependent lower bound is written as a linear program, our algorithms are very simple and do not require solving such an optimization problem during learning. Their design and analyses employ novel ideas, including graph-theoretical concepts (minimum flows) and a new maximum-coverage exploration strategy.

1 Introduction

In reinforcement learning [RL, 40], an agent interacts with an environment modeled as a Markov decision process (MDP) by sequentially selecting actions and receiving feedback in the form of reward signals. Depending on the application, the agent may seek to maximize the cumulative rewards received during learning (which is typically phrased as a *regret minimization* problem) or to minimize the number of learning interactions (i.e., the *sample complexity*) for identifying a near-optimal policy. The latter *pure exploration* problem was introduced in [21] under the name of Probably Approximately Correct (PAC) RL: given two parameters $\varepsilon, \delta > 0$, the agent must return a policy that is ε-optimal with probability at least $1-\delta$. Our work focuses on this problem in the context of episodic (a.k.a. finite-horizon) tabular MDPs.

The PAC RL problem has been mostly studied under the lens of minimax (or worst-case) optimality. In the episodic setting, the algorithm proposed in [12] has sample complexity bounded by $O(SAH^2 \log(1/\delta)/\varepsilon^2)$ for an MDP with S states, A actions, horizon H, and time-homogeneous transitions and rewards (i.e., not depending on the stage). This is minimax optimal for such a context [11]. Similarly, in [35] the authors designed a strategy with $O(SAH^3 \log(1/\delta)/\varepsilon^2)$ complexity in time-inhomogeneous MDPs, which was later shown to be minimax optimal [17].

[*]Work done while at Inria Lille.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

While the minimax framework provides a strong notion of statistical optimality, it does not account for one of the most desirable properties for an RL algorithm: the ability to adapt to the difficulty of the MDP instance. For this reason, researchers recently started to investigate the instance-dependent complexity of PAC RL. Earlier attempts were made in the simplified setting where the agent has access to a generative model (i.e., it can query observations from any state-action pair using a simulator) in γ-discounted infinite-horizon MDPs [50, 5]. The online setting, where the agent can only sample trajectories from the environment, has been studied in [4] for discounted MDPs and in [46] for episodic time-inhomogeneous MDPs. All these works derive sample complexity bounds that scale with certain gaps between optimal value functions. For instance, in the episodic setting, the *value gap* $\Delta_h(s,a) := V_h^\star(s) - Q_h^\star(s,a)$[2] intuitively characterizes the degree of sub-optimality of action a for state s at stage h. Unfortunately, these bounds are known to be sub-optimal and how to achieve instance optimality remains one of the main open questions. In fact, recent works on regret minimization [42, 13] showed that value gaps are often overly conservative, and the same holds for PAC RL. We refer the reader to Appendix A for a deeper discussion on problem-dependent results in RL and the review of other related PAC learning frameworks.

The main challenge towards instance optimality is that existing lower bounds for exploration problems in MDPs [5, 42, 4, 13] are written in terms of non-convex optimization problems. Their "implicit" form makes it hard to understand the actual complexity of the setting and, thus, to design optimal algorithms. Existing solutions either derive explicit *sufficient* complexity measures that inspire algorithmic design [46], or solve (a relaxation of) the optimization problem from the lower bound using the empirical MDP as a proxy for the unknown MDP [4]. The latter extends the Track-and-Stop idea originally proposed in [22] for bandits ($H = 1$), and requires in particular a large amount of forced exploration. Both solutions have limitations. On the one hand, it is not clear if and how such sufficient complexity measures or relaxations are related to an actual lower bound. On the other hand, strategies solving a black-box optimization problem to find an optimal exploration strategy are typically very inefficient and often come with either only asymptotic ($\delta \to 0$) guarantees or with poor (far from minimax optimal) sample complexity in the regime of moderate δ.

Contributions This paper presents a complete study of PAC RL in tabular *deterministic* episodic MDPs with time-inhomogeneous transitions, a sub-class of stochastic MDPs where state transitions are deterministic and the agent observes stochastic rewards from unknown distributions. Our first contribution is an *instance-dependent lower bound* on the sample complexity of any PAC algorithm. We show that the number of visits $n_h^\tau(s,a)$ to any state-action-stage triplet (s,a,h) at the stopping time τ satisfies

$$\mathbb{E}[n_h^\tau(s,a)] \gtrsim \frac{\log(1/\delta)}{\max(\overline{\Delta}_h(s,a),\varepsilon)^2}, \tag{1}$$

where $\overline{\Delta}_h(s,a) := V_1^\star - \max_{\pi \in \Pi_{s,a,h}} V_1^\pi$, with V_1^π the expected return of policy π, $V_1^\star$ the optimal expected return, and $\Pi_{s,a,h}$ the set of all deterministic policies that visit (s,a) at stage h. We call these quantities the *deterministic return gaps* due to their closeness with the *return gaps* introduced in [13] for general MDPs. In deterministic MDPs, the deterministic return gaps are actually H times larger than the return gaps and they are never smaller than value gaps. Our lower bound on the sample complexity τ is then the value of a *minimum flow* with local lower bounds (1), i.e., roughly the minimum number of policies that must be played to ensure (1) for all (s,a,h). To our knowledge, this is the first instance-dependent lower bound for the PAC setting in episodic MDPs.

On the algorithmic side, we design EPRL, a *generic elimination-based method* for PAC RL, and couple it with a novel adaptive sampling rule called *maximum-coverage sampling*. The latter is a simple strategy which does not require solving the optimization problem from the lower bound at learning time in a Track-and-Stop fashion. Instead, it greedily selects the policy that maximizes the number of visited under-sampled triplets (s,a,h), i.e. those having received the least amount of visits so far. We prove that EPRL is (ε,δ)-correct under any sampling rule. Moreover, we show that the sample complexity of EPRL with max-coverage sampling matches our instance-dependent lower bound up to logarithmic factors and a multiplicative $O(H^2)$ term, while also being minimax optimal. Finally, we perform numerical simulations on random deterministic MDPs which reveal that EPRL can indeed improve over existing minimax-optimal algorithms tailored for the deterministic case.

[2] $V^\star$ and $Q^\star$ respectively denote the optimal value and action-value functions, that are defined in Section 2.

2 Preliminaries

Let $\mathcal{M} := (\mathcal{S}, \mathcal{A}, \{f_h, \nu_h\}_{h\in[H]}, s_1, H)$ be a *deterministic* time-inhomogeneous finite-horizon MDP, where $\mathcal{S}$ is a finite set of S states, $\mathcal{A}$ is a finite set of A actions, $f_h : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ and $\nu_h : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathbb{R})$ are respectively the transition function and the reward distribution at stage $h \in [H]$, $s_1 \in \mathcal{S}$ is the unique initial state, and H is the horizon. Without loss of generality, we assume that, at each stage $h \in [H]$ and state $s \in \mathcal{S}$, only a subset $\mathcal{A}_h(s) \subseteq \mathcal{A}$ of actions is available. We denote by $r_h(s,a) := \mathbb{E}_{x\sim\nu_h(s,a)}[x]$ the expected reward after taking action a in state s at stage h.

A deterministic policy $\pi = \{\pi_h\}_{h\in[H]}$ is a sequence of mappings $\pi_h : \mathcal{S} \to \mathcal{A}$. We let $\Pi := \{\pi \mid \forall h \in [H], s \in \mathcal{S} : \pi_h(s) \in \mathcal{A}_h(s)\}$ be the set of all valid deterministic policies. Executing a policy $\pi \in \Pi$ on MDP $\mathcal{M}$ yields a deterministic sequence of states and actions $(s_h^\pi, a_h^\pi)_{h\in[H]}$, where $s_1^\pi = s_1$, $a_h^\pi = \pi_h(s_h^\pi)$ for all $h \in [H]$, and $s_h^\pi = f_{h-1}(s_{h-1}^\pi, a_{h-1}^\pi)$ for all $h \in \{2, \dots, H\}$. We let $\mathcal{S}_h := \{s \in \mathcal{S} \mid \exists \pi \in \Pi : s_h^\pi = s\}$ be the subset of states that are reachable at stage $h \in [H]$. Finally, we define $N := \sum_{h=1}^H \sum_{s\in\mathcal{S}_h} |\mathcal{A}_h(s)|$ as the total number of reachable state-action-stage triplets.

For each (s, a, h), the *action-value function* $Q_h^\pi(s,a)$ of a policy $\pi \in \Pi$ quantifies the expected return when starting from s at stage h, playing a and following π thereafter. In deterministic MDPs, it has the simple expression $Q_h^\pi(s,a) = r_h(s,a) + V_{h+1}^\pi(f_h(s,a))$, where $V_h^\pi(s) := Q_h^\pi(s, \pi_h(s))$ is the corresponding value function (with $V_{H+1}^\pi(s) = 0$). The expected *return* of π is simply its value at the initial state, i.e., $V_1^\pi(s_1) = \sum_{h=1}^H r_h(s_h^\pi, a_h^\pi)$. We let $\Pi^\star := \{\pi^\star \in \Pi : V_1^{\pi^\star}(s_1) = \max_{\pi\in\Pi} V_1^\pi(s_1)\}$ be the set of *optimal policies*, i.e., those with maximal return. Finally, we denote by $V_h^\star(s)$ and $Q_h^\star(s,a)$ the optimal value and action-value function, respectively. These are related by the Bellman optimality equations as $Q_h^\star(s,a) = r_h(s,a) + V_{h+1}^\star(f_h(s,a))$ and $V_h^\star(s) = \max_{a\in\mathcal{A}_h(s)} Q_h^\star(s,a)$.

Learning problem The agent interacts with an MDP $\mathcal{M}$ in episodes indexed by $t \in \mathbb{N}$. At the beginning of the t-th episode, the agent selects a policy $\pi^t \in \Pi$ based on past history through its *sampling rule*, executes it on $\mathcal{M}$, and observes the corresponding deterministic trajectory $(s_h^{\pi^t}, a_h^{\pi^t})_{h\in[H]}$ together with random rewards $(y_h^t)_{h\in[H]}$, where $y_h^t \sim \nu_h(s_h^{\pi^t}, a_h^{\pi^t})$. At the end of each episode, the agent may decide to terminate the process through its *stopping rule* and return a policy $\widehat{\pi}$ prescribed by its *recommendation rule*. We denote by τ its random stopping time. An algorithm for PAC identification is thus made of a triplet $(\{\pi^t\}_{t\in\mathbb{N}}, \tau, \widehat{\pi})$. The goal of the agent is two-fold. First, for given parameters $\varepsilon, \delta > 0$, it must return an ε-optimal policy with probability at least $1 - \delta$.

Definition 1. *An algorithm is (ε, δ)-PAC on a set of MDPs $\mathfrak{M}$ if, for all $\mathcal{M} \in \mathfrak{M}$, it stops a.s. with*

$$\mathbb{P}_{\mathcal{M}}\left(V_1^{\widehat{\pi}}(s_1) \geq V_1^\star(s_1) - \varepsilon\right) \geq 1 - \delta.$$

Second, it should stop as early as possible, i.e., by minimizing the *sample complexity* τ. Henceforth, we assume that the transition function f is known but not the reward distribution ν. Note that if the transitions are unknown, the agent can still estimate them (since it knows that $\mathcal{M}$ is deterministic) with at most $N \leq SAH$ episodes.

Minimum flows We review some basic concepts from graph theory which will be at the core of our algorithms and analyses later. Full details can be found in Appendix B. First note that a deterministic MDP (without reward) can be represented as a *directed acyclic graph* (DAG) with one arc for each available state-action-stage triplet. Let $\mathcal{E} := \{(s,a,h) : h \in [H], s \in \mathcal{S}_h, a \in \mathcal{A}_h(s)\}$ be the set of arcs in the DAG. The minimum flow problem, originally introduced in [45] and later studied in, e.g., [1, 2, 10], consists of findining a flow (i.e., an allocation of visits) of minimal value which satisfies certain demand constraints in each arc of the graph. In our specific setting, we define a *flow* as any non-negative function $\eta : \mathcal{E} \to [0, \infty)$ that belongs to the following set $\Omega := \left\{\eta : \mathcal{E} \to [0,\infty) \mid \sum_{(s',a'):f_{h-1}(s',a')=s} \eta_{h-1}(s',a') = \sum_{a\in\mathcal{A}_h(s)} \eta_h(s,a) \;\; \forall h > 1, s \in \mathcal{S}_h\right\}$.

This implies that a flow, seen as an allocation of visits to the arcs, satisfies the *navigation constraints* (i.e., incoming and outcoming flows are equal at each state). The minimum flow for a non-negative *lower-bound* function $\underline{c} : \mathcal{E} \to [0, \infty)$ is the solution to the following linear program (LP):

$$\varphi^\star(\underline{c}) := \min_{\eta\in\Omega} \sum_{a\in\mathcal{A}_1(s_1)} \eta_1(s_1,a) \quad \text{s.t.} \quad \eta_h(s,a) \geq \underline{c}_h(s,a) \quad \forall (s,a,h) \in \mathcal{E}.$$

Intuitively, the goal is to minimize the amount of flow leaving the initial state while satisfying the navigation and demand constraints. We note that more efficient algorithms exist for this problem than the LP formulation, e.g., the variant of the Ford-Fulkerson method proposed in [10] which is guaranteed to find an integer solution when the lower bound function is integer-valued.

3 The Complexity of PAC RL in Deterministic MDPs

Before stating our lower bound, we formally introduce the new notion of sub-optimality gap it features and compare it with other notions that appeared in the literature.

On sub-optimality gaps The most popular notion of sub-optimality gap is the so-called *value gap*. It was introduced first in the discounted infinite-horizon setting [e.g., 50] and later for episodic MDPs [e.g., 38, 48]. Formally, in the latter context, the value gap of any action $a \in \mathcal{A}_h(s)$ in state $s \in \mathcal{S}_h$ at stage $h \in [H]$ is $\Delta_h(s,a) := V_h^\star(s) - Q_h^\star(s,a)$. Such a notion of gap appears in the complexity measure for PAC RL proposed in [46]. In the deterministic setting, such a complexity measure can be written as $\mathcal{C}(\mathcal{M},\varepsilon) = \sum_{(s,a,h)} \frac{1}{\max(\widetilde{\Delta}_h(s,a),\varepsilon)^2}$, where $\widetilde{\Delta}_h(s,a) = \min_{a':\Delta_h(s,a')>0} \Delta_h(s,a')$ if a is the unique optimal action at (s,h), and $\widetilde{\Delta}_h(s,a) = \Delta_h(s,a)$ otherwise. Intuitively, the (inverse) value gap is proportional to the difficulty of learning whether an action a is sub-optimal for state s at stage h. Then, $\mathcal{C}(\mathcal{M},\varepsilon)$ is proportional to the difficulty of learning a near optimal action at *all* states and stages. Recent works [42, 13] showed that this is actually not necessary: if one only cares about computing a policy maximizing the return at the initial state, it is not necessary to learn an optimal action at states which are not visited by such an optimal policy, in particular when the return of all policies visiting the state is small. The *return gap* [13] was introduced to cope with this limitation. In deterministic MDPs, it can be expressed as $\overline{\mathrm{gap}}_h(s,a) := \frac{1}{H}\min_{\pi\in\Pi_{s,a,h}} \sum_{\ell=1}^h \Delta_\ell(s_\ell^\pi, a_\ell^\pi)$, where we denote by $\Pi_{s,a,h} := \{\pi \in \Pi : s_h^\pi = s, a_h^\pi = a\}$ the subset of deterministic policies that visit (s,a) at stage h. In words, the return gap of (s,a,h) is proportional to the *sum* of value gaps along the best trajectory (i.e., one with maximal return) that visits (s,a) at stage h. Intuitively, this means that, if $\Delta_h(s,a)$ is extremely small but all policies visiting (s,a) at stage h need to play a highly sub-optimal action before, then $\Delta_h(s,a) \ll \overline{\mathrm{gap}}_h(s,a)$. In the deterministic case, our lower bound reveals that the normalization by H is not necessary, and we define the *deterministic return gap* to be

$$\overline{\Delta}_h(s,a) := V^\star(s_1) - \max_{\pi\in\Pi_{s,a,h}} V^\pi(s_1). \tag{2}$$

Using the well-known relationship $V_1^\star(s_1) - V_1^\pi(s_1) = \sum_{h=1}^H \Delta_h(s_h^\pi, a_h^\pi)$ [e.g., 42, Proposition 5], it is easy to see that $\Delta_h(s,a) \le \overline{\Delta}_h(s,a) = H \times \overline{\mathrm{gap}}_h(s,a)$.

Lower Bound We now present our instance-dependent lower bound based on deterministic return gaps, which will guide us in the design and analysis of sample efficient algorithms. This result is the first instance-dependent lower bound for PAC RL in the episodic setting. Lower bounds for ε-best arm identification in a bandit model (which corresponds to $H = S = 1$) were derived in [34, 14, 23], while problem-dependent regret lower bounds for finite-horizon MDPs are provided in [13, 42].

We consider the class $\mathfrak{M}_{\sigma^2}$ of deterministic MDPs with σ^2-*Gaussian* rewards, in which $\nu_h(s,a) = \mathcal{N}(r_h(s,a),\sigma^2)$. Let $\Pi^\varepsilon := \{\pi \in \Pi : V_1^\pi(s_1) \ge V_1^\star(s_1) - \varepsilon\}$ be the set of all ε-optimal policies and denote by $\mathcal{Z}_h^\varepsilon := \{s \in \mathcal{S}_h, a \in \mathcal{A}_h(s) : \Pi_{s,a,h} \cap \Pi^\varepsilon \neq \emptyset\}$ the set of state-action pairs that are reachable at stage h by some ε-optimal policy. Note that $\overline{\Delta}_h(s,a) \le \varepsilon$ for all $(s,a) \in \mathcal{Z}_h^\varepsilon$.

Theorem 1. *Let $\sigma^2 > 0$ and fix any MDP $\mathcal{M} \in \mathfrak{M}_{\sigma^2}$. Then, any algorithm which is (ε,δ)-PAC on the class $\mathfrak{M}_{\sigma^2}$ must satisfy, for any $h \in [H]$, $s \in \mathcal{S}_h$, and $a \in \mathcal{A}_h(s)$,*

$$\mathbb{E}_{\mathcal{M}}[n_h^\tau(s,a)] \ge \underline{c}_h(s,a) := \frac{\sigma^2 \log(1/4\delta)}{4\max(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}^h, \varepsilon)^2}, \tag{3}$$

where $\overline{\Delta}_{\min}^h := \min_{(s',a'):\overline{\Delta}_h(s',a')>0} \overline{\Delta}_h(s',a')$ if $|\mathcal{Z}_h^\varepsilon| = 1$ and $\overline{\Delta}_{\min}^h := 0$ otherwise. Moreover, for $\underline{c} : \mathcal{E} \to [0,\infty)$ the lower bound function defined above,

$$\mathbb{E}_{\mathcal{M}}[\tau] \ge \varphi^\star(\underline{c}). \tag{4}$$

The first lower bound (3) is on the number of visits required for any state-action-stage triplet. It intuitively shows that an (ε,δ)-PAC algorithm must visit each triplet proportionally to its inverse

deterministic return gap. The second one (4) shows that the actual sample complexity of the algorithm must be at least the value of a minimum flow computed with the local lower bounds (3), i.e. that the algorithm must play the minimum number of episodes (i.e., policies) that guarantees (3) for each (s, a, h). Intuitively, due to the navigation constraints of the MDP, there might be no algorithm which tightly matches (3) for each (s, a, h), and (4) is exactly enforcing these constraints. While $\varphi^\star(\underline{c})$ has no explicit form, Lemma 6 in Appendix B gives an idea of how it scales with the gaps:

$$\max_{h\in[H]} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} \frac{\sigma^2 \log(1/4\delta)}{4\max(\overline{\Delta}_h(s,a), \overline{\Delta}^h_{\min}, \varepsilon)^2} \le \varphi^\star(\underline{c}) \le \sum_{h\in[H]} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} \frac{\sigma^2 \log(1/4\delta)}{4\max(\overline{\Delta}_h(s,a), \overline{\Delta}^h_{\min}, \varepsilon)^2}.$$

Observe that the quantity on the right-hand side resembles the complexity measure $\mathcal{C}(\mathcal{M}, \varepsilon)$ [46], except that value gaps are replaced by return gaps. This implies that, in general, our lower bound can be much smaller than this complexity. For instance, in an MDP with extremely small value gaps in states which are not visited by an optimal policy, $\varphi^\star(\underline{c})$ does not scale with such gaps at all.

In Appendix C.2 we further provide a minimax lower bound for PAC RL in deterministic MDPs scaling as $\Omega\left(SAH^2 \log(1/\delta)/\varepsilon^2\right)$, with a reduced H^2 dependency compared to the H^3 that appear in the stochastic case [17]. We note that faster rates for deterministic MDPs have already been obtained in other RL settings [e.g., 49]. The BPI-UCRL algorithm [29] particularized to deterministic MDPs is matching this lower bound and is thus minixal optimal. We now present the first algorithm which is simultaneously minimax optimal for deterministic MDPs and nearly matching (up to $O(H^2)$ and logarithmic factors) the lower bound of Theorem 1.

4 EPRL and Max-Coverage Sampling

We propose a general Elimination-based scheme for PAC RL, called EPRL (Algorithm 1). At each episode $t \in \mathbb{N}$, the algorithm plays a policy π^t selected by some sampling rule. Then, based on the collected samples, the algorithm updates its statistics and eliminates all actions which are detected as sub-optimal with enough confidence. This procedure is repeated until a stopping rule triggers.

Formally, EPRL maintains an estimate $\hat{r}^t_h(s,a) := \frac{1}{n^t_h(s,a)} \sum_{l=1}^t y^l_h \mathbb{1}\left(s^l_h = s, a^l_h = a\right)$, with $\hat{r}^0_h = 0$, of the unknown mean reward $r_h(s,a)$ for each (s,a,h). Here $n^t_h(s,a) := \sum_{l=1}^t \mathbb{1}\left(s^l_h = s, a^l_h = a\right)$ is the number of times (s,a) is visited at stage h up to episode t. We define the following upper and lower confidence intervals to the value functions of a policy $\pi \in \Pi$:

$$\overline{Q}^{t,\pi}_h(s,a) := \hat{r}^t_h(s,a) + b^t_h(s,a) + \overline{V}^{t,\pi}_{h+1}(f_h(s,a)), \quad \overline{V}^{t,\pi}_h(s) := \overline{Q}^{t,\pi}_h(s, \pi_h(s)),$$
$$\underline{Q}^{t,\pi}_h(s,a) := \hat{r}^t_h(s,a) - b^t_h(s,a) + \underline{V}^{t,\pi}_{h+1}(f_h(s,a)), \quad \underline{V}^{t,\pi}_h(s) := \underline{Q}^{t,\pi}_h(s, \pi_h(s)),$$

where $b^t_h(s,a)$ is a *bonus function*, i.e., the width of the confidence interval at (s,a,h). We assume that rewards are σ^2-sub-Gaussian with a known factor σ^2,[3] which allows us to choose

$$b^t_h(s,a) := \sqrt{\frac{\beta(n^t_h(s,a), \delta)}{n^t_h(s,a)}}, \quad \beta(t,\delta) := 2\sigma^2 \log\left(\frac{4t^2 N}{\delta}\right). \tag{5}$$

Elimination rule Algorithm 1 keeps a set of active (or candidate) actions $\mathcal{A}^t_h(s)$ for each stage $h \in [H]$, state $s \in \mathcal{S}_h$, and episode $t \in \mathbb{N}$. Let $\Pi^t := \{\pi \in \Pi \mid \forall s, h : \pi_h(s) \in \mathcal{A}^t_h(s) \vee \mathcal{A}^t_h(s) = \emptyset\}$ be the subset of *active* policies that only play active actions at episode t. Note that an active policy can play an arbitrary action in states where all actions have been eliminated. As can be seen in Line 7 of Algorithm 1, action a is eliminated from $\mathcal{A}^t_h(s)$ if $\max_{\pi\in\Pi_{s,a,h}\cap\Pi^{t-1}} \overline{V}^{t,\pi}_1(s_1) \le \max_{\pi\in\Pi} \underline{V}^{t,\pi}_1(s_1)$, that is, when we are confident that none of the policies visiting (s,a) at stage h is optimal. We recall that $\Pi_{s,a,h}$ denotes the set of all deterministic policies that visit s,a at stage h. The maximum restricted to $\Pi_{s,a,h}$ can be easily computed by standard dynamic programming (e.g., it is enough to set the reward to $-\infty$ for all state-action pairs different than (s,a) at stage h). If $\Pi_{s,a,h} \cap \Pi^{t-1} = \emptyset$, we set the maximum to $-\infty$ so that the elimination rule triggers.

Remark 1. *While defining Π^t simplifies the presentation, EPRL neither stores nor enumerates the set of active policies. In particular, EPRL does not eliminate policies but rather (s,a,h) triplets. The sets $\mathcal{A}^t_h(s)$ can be updated in polynomial time by dynamic programming without ever computing Π^t.*

[3] Note that sub-Gaussianity generalizes the common assumption of bounded rewards in $[0,1]$ (in which case $\sigma^2 = 1/4$) and the one of Gaussian rewards with variance σ^2 (as used in the lower bound of Theorem 1).

Algorithm 1 Elimination-based PAC RL (EPRL) for deterministic MDPs

1: **Input:** deterministic MDP (without reward) $\mathcal{M} := (\mathcal{S}, \mathcal{A}, \{f_h\}_{h\in[H]}, s_1, H)$, ε, δ
2: Initialize $\mathcal{A}_h^0(s) \leftarrow \mathcal{A}_h(s)$ for all $h \in [H], s \in \mathcal{S}_h$
3: Set $n_h^0(s,a) \leftarrow 0$ for all $h \in [H], s \in \mathcal{S}_h, a \in \mathcal{A}_h(s)$
4: **for** $t = 1, \dots$ **do**
5: Play $\pi^t \leftarrow$ SAMPLINGRULE()
6: Update statistics $n_h^t(s,a), \hat{r}_h^t(s,a)$
7: $\mathcal{A}_h^t(s) \leftarrow \mathcal{A}_h^{t-1}(s) \cap \left\{a \in \mathcal{A} : \max_{\pi\in\Pi_{s,a,h}\cap\Pi^{t-1}} \overline{V}_1^{t,\pi}(s_1) \geq \max_{\pi\in\Pi} \underline{V}_1^{t,\pi}(s_1)\right\}$
8: where $\Pi^{t-1} \leftarrow \left\{\pi \in \Pi \mid \forall s,h : \pi_h(s) \in \mathcal{A}_h^{t-1}(s) \vee \mathcal{A}_h^{t-1}(s) = \emptyset\right\}$ (need not be stored/computed)
9: **if** $\max_{\pi\in\Pi^t} \left(\overline{V}_1^{\pi,t}(s_1) - \underline{V}_1^{\pi,t}(s_1)\right) \leq \varepsilon$ or $\forall h \in [H], s \in \mathcal{S}_h : |\mathcal{A}_h^t(s)| \leq 1$ **then**
10: Stop and recommend $\widehat{\pi} \in \arg\max_{\pi\in\Pi^t} \overline{V}_1^{\pi,t}(s_1)$
11: **end if**
12: **end for**
13: **function** MAXCOVERAGE()
14: Let $k_t \leftarrow \min_{h\in[H], s\in\mathcal{S}_h, a\in\mathcal{A}_h^{t-1}(s)} n_h^{t-1}(s,a) + 1$ and $\bar{t}_{k_t} \leftarrow \inf_{l\in\mathbb{N}}\{l : k_l = k_t\}$
15: **if** $t \bmod 2 = 1$ **then**
16: **return** $\pi^t \leftarrow \arg\max_{\pi\in\Pi} \sum_{h=1}^H \mathbb{1}\left(a_h^\pi \in \mathcal{A}_h^{\bar{t}_{k_t}-1}(s_h^\pi), n_h^{t-1}(s_h^\pi, a_h^\pi) < k_t\right)$
17: **else**
18: **return** $\pi^t \leftarrow \arg\max_{\pi\in\Pi^{t-1}} \sum_{h=1}^H b_h^{t-1}(s_h^\pi, a_h^\pi)$ (MAXDIAMETER)
19: **end if**

Stopping rule EPRL uses two different stopping rules (Line 9). The first one checks whether, for all active policies $\pi \in \Pi^t$, the confidence interval on the return, $\overline{V}_1^{\pi,t}(s_1) - \underline{V}_1^{\pi,t}(s_1) = 2\sum_{h=1}^H b_h^t(s_h^\pi, a_h^\pi)$, which we refer to as *diameter*, is below ε. The second one checks whether each set $\mathcal{A}_h^t(s)$ contains either 1 action or 0 actions (which happens when the state is unreachable by an optimal policy). In both cases, we recommend the optimistic (active) policy (Line 10).

Sampling rule While EPRL may be used with different sampling rules, we recommend the max-coverage sampling rule described in Algorithm 1. This sampling rule aims at ensuring that no (s,a,h) triplet remains under-visited for too long. This is achieved by selecting the policy which greedily maximizes the number of visited under-sampled triplets, denoted by $\mathcal{U}_t$. The quantity $k_t = \min_{(s,a,h):a\in\mathcal{A}_h^{t-1}(s)} n_h^{t-1}(s,a) + 1$ can be interpreted as the target minimum number of visits from active triplets that we want to achieve in round t and permit to define

$$\pi^t = \arg\max_{\pi\in\Pi} \sum_{h=1}^H \mathbb{1}\left((s_h^\pi, a_h^\pi, h) \in \mathcal{U}_t\right) \text{ with } \mathcal{U}_t = \left\{(s,a,h) : a \in \mathcal{A}_h^{\bar{t}_{k_t}-1}(s), n_h^{t-1}(s,a) < k_t\right\},$$

where $\bar{t}_k = \inf\{t : k_t = k\}$ is the first round in which the target is set to k. The argmax over Π can be computed using dynamic programming. We emphasize that this argmax is not restricted to the set of active policies, meaning that we may play eliminated actions in order to augment the coverage (that is, the minimal number of visits) faster. Every even round, max-coverage instead chooses an active policy maximizing the diameter featured in the stopping rule (max-diameter sampling). As we shall see in our analysis, this dichotomous behavior is needeed in order to maintain minimax-optimality.

Comparison with other elimination-based algorithms The work of [19] provides a heuristic using action eliminations to find an ε-optimal policy in a discounted MDP. However, no sample complexity guarantees are given for this algorithm, which uses a different elimination rule, based on confidence intervals on the optimal value function, and a uniform sampling rule. The MOCA algorithm [46] also uses a different action elimination rule compared to ours. In particular, the decision to eliminate (s,a,h) is made based only on rewards that can be obtained after visiting (s,a,h). Moreover, this algorithm uses a complex phase-based sampling rule, while the sampling rule of EPRL is fully adaptive.

5 Theoretical Guarantees

Our first result, proved in Appendix D.2.1, shows that EPRL is (ε, δ)-PAC under any sampling rule. It follows from the fact that 1) the choice of bonus function (5) ensures that all the confidence intervals are valid and 2) state-action pairs from optimal trajectories are never eliminated when this holds.

Theorem 2. *Algorithm 1 is (ε, δ)-PAC provided that the sampling rule makes it stop almost surely.*

We now analyze the sample complexity of EPRL combined with max-coverage sampling.

Theorem 3. *(Informal version of Theorem 8 in Appendix D.3) With probability at least $1-\delta$, the sample complexity of EPRL combined with the maximum-coverage sampling rule satisfies $\tau = \widetilde{O}(\varphi^\star(g))$, where $g : \mathcal{E} \to [0, \infty)$ is the lower bound function defined by*

$$g_h(s,a) := \frac{32\sigma^2 H^2}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2}\left(\log\left(\frac{4N^3}{\delta}\right) + 8\log\left(\frac{16\sigma H \log\left(\frac{4N^3}{\delta}\right)}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)}\right)\right) + 2.$$

Moreover, with the same probability, $\tau = \widetilde{O}(\frac{SAH^2}{\varepsilon^2}\log(1/\delta))$, where $\widetilde{O}$ hides logarithmic terms.

First note that EPRL combined with such a sampling rule is *minimax optimal*, since it matches the worst-case lower bound derived in Appendix C.2. In addition, the leading term in the instance-dependent complexity is the value of a minimum flow with a lower bound function g that, in case multiple disjoint optimal trajectories exist[4], matches the gap-dependence in (1). If we suppose that there exist at least two disjoint optimal trajectories, in which case $\overline{\Delta}_{\min} = \overline{\Delta}^h_{\min} = 0$, then, thanks to Lemma 7 in Appendix B, one can easily see that $\varphi^\star(g) \leq \alpha H^2 \varphi^\star(\underline{c}) + \varphi^\star(g')$, where $g'_h(s,a) := \widetilde{O}(H^2/\max\left(\overline{\Delta}_h(s,a), \varepsilon\right)^2)$ does not depend on δ, $\underline{c}$ is the "optimal" lower bound function from (1), and α is a numerical constant. Hence, in the asymptotic regime ($\delta \to 0$), $\varphi^\star(g)$ matches our lower bound up to a $O(H^2)$ multiplicative factor.

Remark 2. *Since Theorem 1 was derived for Gaussian rewards, EPRL is instance-optimal only when the reward distribution is Gaussian. This is not surprising since it is well known from the bandit literature [e.g., 32] that sample complexity bounds scaling with a sum of inverse squared gaps are optimal only for Gaussian distributions. Note, however, that EPRL works in greater generality and achieves complexity $\varphi^\star(\underline{c})$ for any σ^2-sub-Gaussian distribution without knowing its specific form (e.g., whether it is Gaussian or not). What is the optimal rate for other common distributions (e.g., bounded rewards in $[0,1]$) and how to achieve it remains an open question.*

Finally, our sample complexity bound has an extra multiplicative logarithmic term which roughly scales as $O(\log(H)\log(H\log(1/\delta)/\varepsilon))$. While this term makes the dependence on δ sub-optimal by a $\log\log(1/\delta)$ factor, we show in Appendix E that it can be removed in the specific case of tree-based MDPs [13].

Remark 3. *We believe that the sub-optimality on H could be reduced to a single H factor by boosting the lower bound. In Appendix E, we show that this is indeed possible in tree-based MDPs. As for the upper bound, reducing H^2 to H is likely to require tighter concentration bounds on values.*

Remark 4. *In Appendix D.4, we prove that, when using the max-diameter sampling rule (Line 18 in Algorithm 1) at each step, the sample complexity is $\widetilde{O}(\sum_{(s,a,h)} H^2/\max(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon)^2)$. While this scales with the same gaps as Theorem 3, it is only a naive upper bound to the minimum flow value (see Section 3). The intuition is that max-diameter sampling alone does not ensure that all triplets are visited sufficiently often, which prevents us from tightly controlling their elimination times.*

Proof sketch The complete proof is given in Appendix D.3. It first relies on the following crucial result which relates the deterministic return gaps to the sum of confidence bonuses.

Lemma 1 (Diameter vs gaps)**.** *With probability at least $1-\delta$, for any $t \in \mathbb{N}, h \in [H], s \in \mathcal{S}_h, a \in \mathcal{A}_h(s)$, if $a \in \mathcal{A}^t_h(s)$ and the algorithm did not stop at the end of episode t,*

$$\max\left(\frac{\overline{\Delta}_h(s,a)}{4}, \frac{\overline{\Delta}_{\min}}{4}, \frac{\varepsilon}{2}\right) \leq \max_{\pi \in \Pi^{t-1}} \sum_{h=1}^{H} b^t_h(s^\pi_h, a^\pi_h),$$

[4]When there is a unique optimal trajectory, our upper bound scales with $\overline{\Delta}_{\min} = \min_{h\in[H]} \overline{\Delta}^h_{\min}$ at all stages h, while the lower bound scales with $\overline{\Delta}^h_{\min}$ at stage h. We believe the latter should be improvable to obtain a dependence on $\overline{\Delta}_{\min}$ matching the one in the upper bound.

where $\overline{\Delta}_{\min} := \min_{h\in[H]} \min_{s\in\mathcal{S}_h} \min_{a:\overline{\Delta}_h(s,a)>0} \overline{\Delta}_h(s,a)$ *if there exists a unique optimal trajectory* $(s_h^\star, a_h^\star)_{h\in[H]}$*, and* $\overline{\Delta}_{\min} := 0$ *in the opposite case.*

In our analysis, we refer to the set of consecutive time steps $\{t \in \mathbb{N} : k_t = k\}$ as the k-th *period*. Using the fact that in period $k+1$ each active triplet has been visited at least k times (which allows to upper bound each bonus $b_h^t(s_h^\pi, a_h^\pi)$ for $\pi \in \Pi^{t-1}$ by a quantity scaling in $\sqrt{1/k}$), one can use Lemma 1 to obtain an upper bound $\overline{\kappa}_{s,a,h} \simeq \frac{H^2 \log(1/\delta)}{\max(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon)^2}$ on the last period in which (s,a,h) is active (Lemma 18 in Appendix D.3). A crucial step of the proof is then to upper bound the duration of the k-th period, $d_k := \sum_{t=1}^{\tau} \mathbb{1}(k_t = k)$.

Lemma 2. $d_k \le 2(\log(H)+1)\varphi^\star(\underline{c}^k)$ *where* $\underline{c}_h^k(s,a) = \mathbb{1}(a \in \mathcal{A}_h^{\bar{t}_k - 1}(s), n_h^{\bar{t}_k-1}(s,a) < k)$.

The intuition behind this result is as follows. Recall that the goal of the max-coverage sampling rule in period k is to visit at least once each (s,a,h) that is active (i.e., $a \in \mathcal{A}_h^{\bar{t}_k-1}(s)$) and undersampled (i.e., $n_h^{\bar{t}_k-1}(s,a) < k$). By definition, the minimum flow $\varphi^\star(\underline{c}^k)$ is the minimum number of policies that need to be played to achieve this goal. Interestingly, Lemma 2 shows that the number of policies played by max-coverage to visit all active undersampled triplets is very close to its theoretical minimum, despite the fact that the algorithm never computes an actual minimum flow. We prove this by interpreting max-coverage sampling as a greedy maximization of some coverage function (related to a minimum flow problem) and leveraging the theory of sub-modular maximization [e.g., 31].

Thanks to Lemma 2, we have that

$$\tau \le 2(\log(H)+1) \sum_{k=1}^{k_\tau} \varphi^\star(\underline{c}^k),$$

where k_τ is the index of the period at which the algorithm stops. To bound this quantity we carefully apply the theory of minimum flows and their dual problem of *maximum cuts*. Let us define a cut $\mathcal{C}$ as any subset of states containing the initial state and let $\mathcal{E}(\mathcal{C})$ be the set of arcs that connect states in $\mathcal{C}$ with states not in $\mathcal{C}$. The well-known min-flow-max-cut theorem (Theorem 4 stated in Appendix B) states that, for any lower bound function $\underline{c}$, $\varphi^\star(\underline{c}) = \max_{\mathcal{C}\in\mathfrak{C}} \sum_{(s,a,h)} \underline{c}_h(s,a)$, where $\mathfrak{C}$ denotes the set of all valid cuts. Then,

$$k\varphi^\star(\underline{c}^k) \le \max_{\mathcal{C}\in\mathfrak{C}} \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C})} k\mathbb{1}\left(a \in \mathcal{A}_h^{\bar{t}_k-1}(s)\right) \le \max_{\mathcal{C}\in\mathfrak{C}} \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C})} (\overline{\kappa}_{s,a,h}+1) = \varphi^\star(g),$$

where $g : \mathcal{E} \to [0,\infty)$ is defined by $g_h(s,a) = \overline{\kappa}_{s,a,h} + 1$. It follows that

$$\begin{aligned}
\tau &\le 2(\log(H)+1) \sum_{k=1}^{k_\tau} \frac{1}{k} \varphi^\star(g) \\
&\le 2(\log(H)+1)\,(\log(k_\tau)+1)\,\varphi^\star(g) \\
&\le 2(\log(H)+1) \left(\max_{(s,a,h)} \log(\overline{\kappa}_{s,a,h}) + 1\right) \varphi^\star(g).
\end{aligned}$$

Using the expression of $\overline{\kappa}_{s,a,h}$ given in Lemma 18 of Appendix D.3 concludes the proof of the stated $\widetilde{O}(\varphi^\star(g))$ instance-dependent bound. For the worse-case bound, we refer the reader to Theorem 12.

□

6 Experiments

We compare numerically EPRL to the minimax optimal BPI-UCRL algorithm [29], adapted to the deterministic setting, on synthetic MDP instances. For EPRL, we experiment with two sampling rules: max-coverage (maxCov) and max-diameter (maxD, see Line 18 of Algorithm 1). We defer to Appendix F some implementation details, including a precise description of the BPI-UCRL baseline.

We generate random "easy" deteministic MDP instances with Gaussian rewards of variance 1 using the following protocol. For fixed S, A, H the mean rewards $r_h(s,a)$ are drawn i.i.d. from a uniform

distribution over $[0, 1]$ and for each state-action pair, the next state is chosen uniformly at random in $\{1, \dots, S\}$. Finally, we only keep MDP instances whose minimum value gap, denoted by $\Delta_{\min}$, is larger than 0.1. Our first observation is that depending on the MDP, the identity of the best performing algorithm can be different. In Figure 1 we show the distribution of the sample complexity (estimated over 10 Monte Carlo simulations) for three different MDPs obtained from our sampling procedure with $S, A = 2$ and $H = 3$ and for algorithms that are run with parameters $\delta = 0.1$ and $\varepsilon = 1.5\Delta_{\min}$.

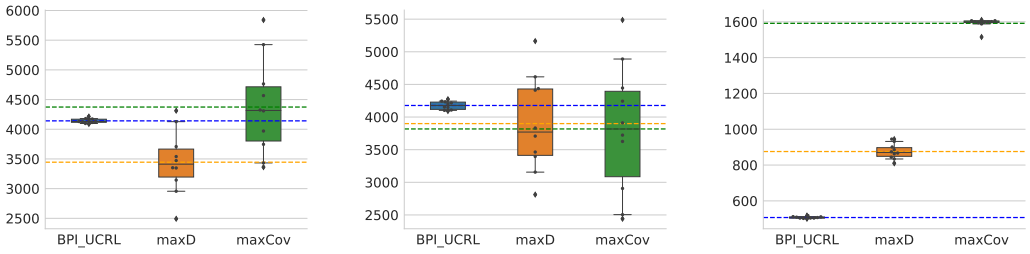

Figure 1: Distribution of stopping times on particular MDPs over 10 runs, with $\varepsilon = 1.5\Delta_{\min}$. The horizontal lines represent the average sample complexity.

To get a better understanding of this phenomenon, we then generated 10 MDP instances of size $(S, A, H) = (2, 2, 3)$ and for each MDP we ran EPRL and BPI-UCRL for 25 values of ε in a grid $[0.05\Delta_{\min}, 10\Delta_{\min}]$ and $\delta = 0.1$. We ran 10 Monte-Carlo simulations for each value of the triplet (MDP, algorithm A, ε), in order to estimate the expected sample complexity $\mathbb{E}_{\mathsf{A}}[\tau_\delta]$. In Figure 2 we plot the relative performance (ratio of sample complexities) of different algorithms as a function of the value of $\varepsilon/\Delta_{\min}$: each point corresponds to a different MDP and a different value of ε. We observe that for large values of $\varepsilon/\Delta_{\min}$, BPI-UCRL has a smaller sample complexity than both versions of EPRL, with a ratio never exceeding 2 (resp. 3) for max-diameter (resp. max-coverage). However, in the more interesting small $\varepsilon/\Delta_{\min}$ regime EPRL is better by several orders of magnitude. This is expected since, for small ε, EPRL is able, through its elimination rule, to identify the optimal policy long before the diameter goes below ε. We observe that the threshold of $\varepsilon/\Delta_{\min}$ at which EPRL algorithms become a better choice than BPI-UCRL seems to vary with the MDP.

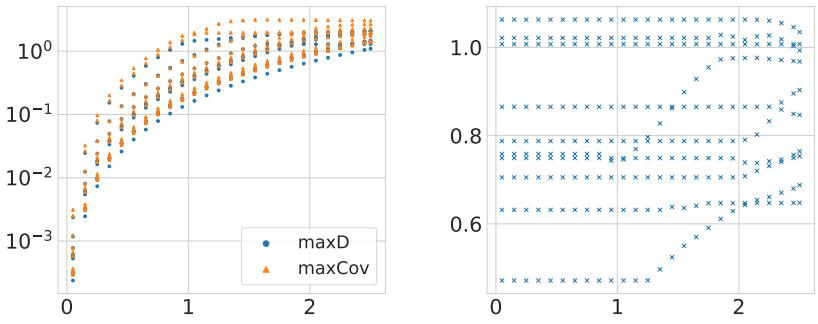

Figure 2: Ratios in log-scale $\mathbb{E}_{\mathrm{A}}[\tau_\delta]/\mathbb{E}_{\mathrm{BPI-UCRL}}[\tau_\delta]$ for A in $\{\text{maxD}, \text{maxCov}\}$ (left) and $\mathbb{E}_{\mathrm{maxD}}[\tau_\delta]/\mathbb{E}_{\mathrm{maxCov}}[\tau_\delta]$ (right) as a function of $\varepsilon/\Delta_{\min}$.

Our experiments also reveal an intriguing phenomenon: the use of max-diameter sampling within EPRL often outperforms max-coverage sampling, even if there exists MDPs (2 out of 10 in our experiments) in which max-coverage is indeed empirically better. We leave as future work to obtain a better characterizations of MDPs for which EPRL with max-coverage sampling performs best.

7 Discussion

We derived an instance-dependent and a worst-case lower bound characterizing the complexity of PAC RL in deterministic MDPs, and proposed a general elimination algorithm together with a novel maximum-coverage sampling rule that nearly matches them (up to $O(H^2)$ and logarithmic factors). We conclude with some discussion about our results and future directions.

Max-coverage vs max-diameter While minimax optimality can be easily achieved with very simple strategies (like max-diameter or BPI-UCRL), instance optimality requires careful algorithmic design. Our coverage-based strategy is built around the idea of "uniformly" exploring the whole MDP, while using an elimination strategy to ensure that no (s, a, h) is sampled much more than what the lower bound prescribes. Notably, this sampling rule is very simple, while exiting PAC RL algorithms with instance-dependent complexity are all quite involved [46, 4]. Moreover, max-coverage sampling naturally extends to stochastic MDPs, e.g., by doing optimistic planning on an MDP with a reward function equal to 1 for under-sampled triplets and 0 for the others. Finally, in our experiments on random instances, we observed that max-diameter is often comparable or better than max-coverage. We leave as future work to investigate whether the latter is also provably near instance-optimal.

Computational aspects Our sampling rule requires solving one dynamic program per episode, which takes $O(N)$ time. The bottleneck is the elimination rule, which requires $O(N^2)$ per-episode time complexity to solve one dynamic program for each active triplet. However, we note that eliminations could be checked periodically (e.g., even at exponentially-separated times) without significantly compromising the sample complexity guarantees.

Improving our results Our instance-dependent upper bound for max-coverage sampling is sub-optimal by a factor H^2 and a multiplicative $O(\log\log(1/\delta))$ term. In Appendix E, we show that, for the specific sub-class of tree-based MDPs [13], we can obtain improved results in all these aspects. In particular, we show that (1) the lower bound scales with an extra factor H and it is fully explicit, (2) the multiplicative log terms in the sample complexity of coverage-based sampling can be removed, and (3) maximum-diameter sampling also achieves near instance-optimal guarantees.

Beyond Gaussian distributions As it is common, e.g., in the bandit literature, the gaps in our lower and upper bounds are optimal only for Gaussian reward distributions. Extending Theorem 1 to general distributions is actually simple (see, e.g., [28] and Lemma 8 in Appendix C). However, this would yield gaps written in terms of KL divergences between arm distributions rather than differences of mean rewards as in the Gaussian case. How to match such gaps is an interesting open question.

Instance optimality in stochastic MDPs The main open question is how to achieve (near) instance-optimality for PAC RL in stochastic MDPs. We believe that many of the results presented in this paper could help in this direction. First, our instance-dependent lower bound could be extended to the stochastic case by modifying return gaps to include visitation probabilities and minimum flows to account for stochastic navigation constraints. Second, on the algorithmic side, our maximum-coverage sampling rule easily extends to stochastic MDPs as mentioned above, while our elimination rule could also be adapted by computing the optimistic return of policies visiting a certain (s, a, h) with a least some probability, which corresponds to a constrained MDP problem [e.g., 18]. Studying how these components behave in stochastic MDPs is an exciting direction for future work.

Acknowledgments and Disclosure of Funding

Aymen Al-Marjani ackowledges the support of the Chaire SeqALO (ANR-20-CHIA-0020). Emilie Kaufmann acknoweldges the support of the French National Research Agency under the BOLD project (ANR-19-CE23-0026-04).

References

[1] Veena Adlakha, Barbara Gladysz, and Jerzy Kamburowski. Minimum flows in (s, t) planar networks. *Networks*, 21(7):767–773, 1991.

[2] Veena G Adlakha. An alternate linear algorithm for the minimum flow problem. *Journal of the Operational Research Society*, 50(2):177–182, 1999.

[3] Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning. *Advances in Neural Information Processing Systems*, 33:13399–13412, 2020.

[4] Aymen Al Marjani, Aurélien Garivier, and Alexandre Proutiere. Navigating to the best policy in markov decision processes. *Advances in Neural Information Processing Systems*, 34, 2021.

[5] Aymen Al Marjani and Alexandre Proutiere. Adaptive sampling for best policy identification in markov decision processes. In *International Conference on Machine Learning*, pages 7459–7468. PMLR, 2021.

[6] Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21, 2008.

[7] Marco Brandizi, Natalja Kurbatova, Ugis Sarkans, and Philippe Rocca-Serra. graph2tab, a library to convert experimental workflow graphs into tabular formats. *Bioinformatics*, 28(12):1665–1667, 2012.

[8] S Bubeck and R Munos. Open loop optimistic planning. In *Conference on Learning Theory*, 2010.

[9] Apostolos N Burnetas and Michael N Katehakis. Optimal adaptive policies for markov decision processes. *Mathematics of Operations Research*, 22(1):222–255, 1997.

[10] Eleonor Ciurea and Laura Ciupala. Sequential and parallel algorithms for minimum flows. *Journal of Applied Mathematics and Computing*, 15(1):53–75, 2004.

[11] Christoph Dann and Emma Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[12] Christoph Dann, Lihong Li, Wei Wei, and Emma Brunskill. Policy certificates: Towards accountable reinforcement learning. In *International Conference on Machine Learning*, pages 1507–1516. PMLR, 2019.

[13] Christoph Dann, Teodor V. Marinov, Mehryar Mohri, and Julian Zimmert. Beyond value-function gaps: Improved instance-dependent regret bounds for episodic reinforcement learning. *CoRR*, abs/2107.01264, 2021.

[14] Rémy Degenne and Wouter M. Koolen. Pure exploration with multiple correct answers. In *NeurIPS*, 2019.

[15] Rémy Degenne, Wouter M Koolen, and Pierre Ménard. Non-asymptotic pure exploration by solving games. In *Advances in Neural Information Processing Systems*, pages 14492–14501, 2019.

[16] Omar Darwiche Domingues, Yannis Flet-Berliac, Edouard Leurent, Pierre Ménard, Xuedong Shang, and Michal Valko. rlberry - A Reinforcement Learning Library for Research and Education. `https://github.com/rlberry-py/rlberry`, 2021.

[17] Omar Darwiche Domingues, Pierre Ménard, Emilie Kaufmann, and Michal Valko. Episodic reinforcement learning in finite mdps: Minimax lower bounds revisited. In *Algorithmic Learning Theory (ALT)*, 2021.

[18] Yonathan Efroni, Shie Mannor, and Matteo Pirotta. Exploration-exploitation in constrained mdps. *arXiv preprint arXiv:2003.02189*, 2020.

[19] E. Even-Dar, S. Mannor, and Y. Mansour. Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

[20] Zohar Feldman and Carmel Domshlak. Simple regret optimization in online planning for markov decision processes. *Journal of Artifial Intelligence Research*, 51:165–205, 2014.

[21] Claude-Nicolas Fiechter. Efficient reinforcement learning. In *Proceedings of the Seventh Conference on Computational Learning Theory (COLT)*, 1994.

[22] Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, pages 998–1027. PMLR, 2016.

[23] Aurélien Garivier and Emilie Kaufmann. Nonasymptotic sequential tests for overlapping hypotheses applied to near-optimal arm identification in bandit models. *Sequential Analysis*, 40(1):61–96, 2021.

[24] J.-B. Grill, M. Valko, and R. Munos. Blazing the trails before beating the path: Sample-efficient monte-carlo planning. In *Neural Information Processing Systems (NeurIPS)*, 2016.

[25] Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. *arXiv:2002.02794*, 2020.

[26] Anders Jonsson, Emilie Kaufmann, Pierre Ménard, Omar Darwiche Domingues, Edouard Leurent, and Michal Valko. Planning in markov decision processes with gap-dependent sample complexity. *Advances in Neural Information Processing Systems*, 33:1253–1263, 2020.

[27] Sham Kakade. *On the Sample Complexity of Reinforcement Learning*. PhD thesis, University College London, 2003.

[28] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1):1–42, 2016.

[29] Emilie Kaufmann, Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Edouard Leurent, and Michal Valko. Adaptive reward-free exploration. In *Algorithmic Learning Theory (ALT)*, 2021.

[30] Michael J. Kearns, Yishay Mansour, and Andrew Y. Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine Learning*, 49(2-3):193–208, 2002.

[31] Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability*, 3:71–104, 2014.

[32] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[33] Tor Lattimore and Csaba Szepesvari. *Bandit Algorithms*. Cambridge University Press, 2019.

[34] S. Mannor and J. Tsitsiklis. The Sample Complexity of Exploration in the Multi-Armed Bandit Problem. *Journal of Machine Learning Research*, pages 623–648, 2004.

[35] Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Emilie Kaufmann, Edouard Leurent, and Michal Valko. Fast active learning for pure exploration in reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2021.

[36] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

[37] Jungseul Ok, Alexandre Proutiere, and Damianos Tranos. Exploration in structured reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 8874–8882, 2018.

[38] Max Simchowitz and Kevin G. Jamieson. Non-asymptotic gap-dependent regret bounds for tabular mdps. In *NeurIPS*, pages 1151–1160, 2019.

[39] Alexander L Strehl, Lihong Li, and Michael L Littman. Reinforcement learning in finite mdps: Pac analysis. *Journal of Machine Learning Research*, 10(11), 2009.

[40] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[41] Ambuj Tewari and Peter L. Bartlett. Optimistic linear programming gives logarithmic regret for irreducible mdps. In *NIPS*, pages 1505–1512. Curran Associates, Inc., 2007.

[42] Andrea Tirinzoni, Matteo Pirotta, and Alessandro Lazaric. A fully problem-dependent regret lower bound for finite-horizon mdps. *arXiv preprint arXiv:2106.13013*, 2021.

[43] Andrea Tirinzoni, Matteo Pirotta, Marcello Restelli, and Alessandro Lazaric. An asymptotically optimal primal-dual incremental algorithm for contextual linear bandits. *Advances in Neural Information Processing Systems*, 33:1417–1427, 2020.

[44] Damianos Tranos and Alexandre Proutière. Regret analysis in deterministic reinforcement learning. In *CDC*. IEEE, 2021.

[45] Yu V Voitishin. Algorithms for solving for the minimal flow in a network. *Cybernetics*, 16(1):131–134, 1980.

[46] Andrew Wagenmaker, Max Simchowitz, and Kevin G. Jamieson. Beyond no regret: Instance-dependent PAC reinforcement learning. In *Conference On Learning Theory (COLT)*, 2022.

[47] Po-An Wang, Ruo-Chun Tzeng, and Alexandre Proutiere. Fast pure exploration via frank-wolfe. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[48] Haike Xu, Tengyu Ma, and Simon S Du. Fine-grained gap-dependent bounds for tabular mdps via adaptive multi-step bootstrap. *arXiv preprint arXiv:2102.04692*, 2021.

[49] Ming Yin and Yu-Xiang Wang. Towards instance-optimal offline reinforcement learning with pessimism. In *NeurIPS*, 2021.

[50] Andrea Zanette, Mykel J. Kochenderfer, and Emma Brunskill. Almost horizon-free structure-aware best policy identification with a generative model. In *NeurIPS*, pages 5626–5635, 2019.

[51] Xuezhou Zhang, Yuzhe Ma, and Adish Singla. Task-agnostic exploration in reinforcement learning. In *NeurIPS*, 2020.

[52] Zihan Zhang, Simon Du, and Xiangyang Ji. Near optimal reward-free reinforcement learning. In *International Conference on Machine Learning, (ICML)*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See the discussion section.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Our results are theoretical and the paper is not oriented towards a specific application, so a wider broader impact discussion is not applicable.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] These are stated in the preamble of every Theorem.
   (b) Did you include complete proofs of all theoretical results? [Yes] See the appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See the appendix.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See the appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See the appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

Table of Contents

A Additional Related Work

In this section, we mention other PAC learning problems that are related to our setting, and discuss problem-dependent guarantees for regret minimization.

Other PAC frameworks The PAC RL problem, which concerns the *identification* of an ε-optimal policy, should not be confused with the PAC-MDP setting introduced by [27]. The latter is closer to regret minimization, as the agent interacts with the MDP online and seeks to maximize the number of learning steps where an ε-optimal policy is played. It has been studied mostly in the discounted infinite-horizon setting. We refer the reader to [39] for a review of this setting. Other PAC pure exploration problems have been studied in the literature. In Monte-Carlo planning, the goal is to find an ε-optimal action in a given state with high probability, rather than a complete policy. Most works have obtained worst-case upper bounds on the sample complexity of planning, scaling in terms of ε and some appropriate notion of near-optimality dimension [30, 8, 24]. Another line of work has derived problem-dependent guarantees in MDPs with a finite branching factor [20, 26], exhibiting a scaling with the value gap at step $h = 1$. Finally, in reward-free exploration [25, 29, 52, 35] or task agnostic exploration [51] the goal is to explore the MDP in order to be able to find a near-optimal policy w.r.t. a reward function which is revealed only after the exploration phase. In the minimax sense, this setting is harder than our PAC RL problem: with an arbitray set of possible reward functions, existing algorithms exhibit an extra multiplicative dependence on the number of states in their sample complexity.

Instance-dependent bounds for regret minimization The majority of instance-dependent results in the RL literature concerns regret minimization. The earliest works in this context focused on ergodic average-reward MDPs [9, 41, 37]. These works presented asymptotic lower bounds (expressed as linear programs) on the expected regret of any "good" strategy, together with algorithms matching them as the number of learning steps tend to infinity. Average-reward communicating MDPs were studied by [6, 44], with the latter proposing an asymptotic regret lower bound for deterministic MDPs. In the episodic setting, [38, 48] derived finite-time regret bounds which are roughly $O(\sum_{s,a,h} \frac{\log T}{\Delta_h(s,a)})$, where T is the number of episodes and $\Delta_h(s,a)$ are the value gaps defined above. These results were later improved by [13], who derived regret bounds of the same shape but scaling with tighter "return gaps". Moreover, [13] and [42] concurrently derived similar asymptotic instance-dependent lower bounds for the episodic setting. However, similarly to the one of [5], these lower bounds are written as non-convex optimization problems and it is an open question whether and how they can be matched.

PAC identification in structured bandits Learning in a finite-horizon MDP can be seen as a *structured bandit* problem with one arm for each deterministic policy whose return can be described by only N parameters (the mean rewards), see, e.g., Appendix B.6 of [42]. As in our setting, instance-dependent lower bounds for structured bandits are often written as optimization problems with no closed-form solution. For this reason, the majority of (near) instance-optimal algorithms for structured bandits either repeatedly solve such an optimization problem during the learning process [22] or solve it incrementally using, e.g., no-regret learners [15], primal-dual methods [43], or Frank-Wolfe [47]. Notably, our coverage-based sampling rules achieve near instance optimality without doing any of this. This is advantageous for at least two reasons: (1) the optimization problem in (4) depends on unknown quantities (such as the gaps) whose estimation typically requires performing additional exploration than what prescribed by the lower bound, hence negatively affecting the sample complexity. (2) Repeatedly solving the minimum flow problem (4), despite being a linear program, can be very computationally demanding, while findind an integer minimum flow or computing its greedy approximation is much more efficient. We wonder whether, taking inspiration from our work, near optimal strategies for general structured bandits could be designed without ever solving the optimization problems from lower bounds.

B Minimum Flows and Maximum Cuts

First note that a deterministic MDP (without reward) $\mathcal{M} := (\mathcal{S}, \mathcal{A}, \{f_h\}_{h\in[H]}, s_1, H)$ can be represented as a *layered directed acyclic graph* (DAG) $\mathcal{G}(\mathcal{M}) := (\mathcal{N}, \mathcal{E}, s_1, s_{H+1})$ with nodes $\mathcal{N} := \{(s,h) : h \in [H], s \in \mathcal{S}_h\}$, arcs $\mathcal{E} := \{(s,a,h) : h \in [H], s \in \mathcal{S}_h, a \in \mathcal{A}_h(s)\}$, a unique

source node $(s_1, 1)$, and a fictitious *sink* node $(s_{H+1}, H+1)$ which is the endpoint of every arc $(s, a, H) \in \mathcal{E}$. In particular, for node $(s, h) \in \mathcal{N}$, there is one arc for each $a \in \mathcal{A}_h(s)$ which connects the node to $(f_h(s, a), h+1)$. The graph is *layered*, in the sense that the set of nodes can be partitioned into H subsets $(\{(s, h) : s \in \mathcal{S}_h\})_{h \in [H]}$, one for each stage, and transitions are possible only between adjacent stages. Let $\mathcal{I}_h(s) := \{(s', a') \in \mathcal{S} \times \mathcal{A} \mid s' \in \mathcal{S}_{h-1}, a' \in \mathcal{A}_{h-1}(s), f_{h-1}(s', a') = s\}$ be the set of incoming arcs into (s, h).

B.1 The minimum flow problem

In Section 2, we introduced a specific instance of the minimum flow problem for layered DAGs with unbounded capacities. Here we introduce the general problem as described, e.g., by [10]. While we still use notation for layered DAGs, we note that all results in this section hold for general directed graphs.

Recall that a *flow* is a non-negative function $\eta : \mathcal{E} \to [0, \infty)$ satisfying the navigation constraints whose value is given by $\varphi(\eta) := \sum_{a \in \mathcal{A}_1(s_1)} \eta_1(s_1, a)$. Let $\underline{c}, \overline{c} : \mathcal{E} \to [0, \infty)$ be two non-negative functions. We say that a flow η is *feasible* if

$$\underline{c}_h(s, a) \le \eta_h(s, a) \le \overline{c}_h(s, a) \quad \forall (s, a, h) \in \mathcal{E}.$$

That is, $\underline{c}_h(s, a)$ acts as a lower bound on the flow we require through arc (s, a, h), while $\overline{c}_h(s, a)$ is the capacity of that arc. Finding a feasible flow of minimum value can be clearly solved as a linear program,

$$\begin{aligned}
&\underset{\eta \in \mathbb{R}^{SAH}}{\text{minimize}} \sum_{a \in \mathcal{A}_1(s_1)} \eta_1(s_1, a), \\
&\text{subject to} \\
&\sum_{(s', a') \in \mathcal{I}_h(s)} \eta_{h-1}(s', a') = \sum_{a \in \mathcal{A}_h(s)} \eta_h(s, a) \quad \forall (s, h) \in \mathcal{N} \setminus \{(s_1, 1), (s_{H+1}, H+1)\}, \\
&\underline{c}_h(s, a) \le \eta_h(s, a) \le \overline{c}_h(s, a) \quad \forall (s, a, h) \in \mathcal{E}.
\end{aligned}$$

We let $\varphi^\star(\underline{c}, \overline{c})$ be its optimal value.

Residual graph The *residual* of an arc $(s, a, h) \in \mathcal{E}$ is defined as

$$\rho_h(s, a) := \eta_h(s, a) - \underline{c}_h(s, a)$$

For each $(s, a, h) \in \mathcal{E}$, we also define the residual of a fictitious backward arc (which does not exist in our layered DAG) as

$$\rho_h^{\text{bw}}(s, a) := \overline{c}_h(s, a) - \eta_h(s, a).$$

Then, we define the *residual graph* $\mathcal{G}_\eta(\mathcal{M})$ as a graph with the same nodes as $\mathcal{G}(\mathcal{M})$ and one arc for each forward or backward arc of $\mathcal{G}(\mathcal{M})$ with strictly positive residual. Note that, in our layered DAG setting, even if the original graph $\mathcal{G}_\eta(\mathcal{M})$ has only forward arcs (transitions are only possible from two successive stages), its residual graph $\mathcal{G}_\eta(\mathcal{M})$ might contain backward arcs if the fictitious backward arcs introduced before have positive residual. Intuitively, a forward arc (s, a, h) in $\mathcal{G}_\eta(\mathcal{M})$ means that we can decrease the flow in (s, a, h) by at most $\rho_h(s, a)$ units, while its corresponding backward arc means that we can increase the flow by at most $\rho_h^{\text{bw}}(s, a)$ units. Finally, we call any path from the source node $(s_1, 1)$ to the sink node $(s_{H+1}, H+1)$ in $\mathcal{G}_\eta(\mathcal{M})$ a *decreasing path*. This is a path where we can reduce the amount of flow while still satisfying all constraints.

Maximum cuts A *source-sink cut* is a partition of the set of nodes $\mathcal{N}$ into two subsets $\mathcal{C} \subseteq \mathcal{N}$ and $\mathcal{N} \setminus \mathcal{C}$ such that $(s_1, 1) \in \mathcal{C}$ and $(s_{H+1}, H+1) \in \mathcal{N} \setminus \mathcal{C}$. As such, we will identify a cut by a single subset of states $\mathcal{C} \subseteq \mathcal{N} \setminus \{(s_{H+1}, H+1)\}$ such that $(s_1, 1) \in \mathcal{C}$. The set of *forward arcs* of a cut $\mathcal{C}$ is

$$\mathcal{E}(\mathcal{C}) := \{(s, a, h) \in \mathcal{E} : (s, h) \in \mathcal{C}, (f_h(s, a), h+1) \in \mathcal{N} \setminus \mathcal{C}\}.$$

The value of a cut $\mathcal{C}$, as defined by [10], is

$$\psi(\mathcal{C}, \underline{c}, \overline{c}) := \sum_{(s,a,h) \in \mathcal{E}(\mathcal{C})} \underline{c}_h(s, a) - \sum_{(s,a,h) \in \mathcal{E}_{\text{bw}}(\mathcal{C})} \overline{c}_h(s, a),$$

where we also define $\mathcal{E}_{\text{bw}}(\mathcal{C}) := \{(s,a,h) \in \mathcal{E} : (s,h) \in \mathcal{N} \setminus \mathcal{C}, (f_h(s,a), h+1) \in \mathcal{C}\}$ as the set of *backward arcs* in the cut. We now present an important result (Lemma 4) which shows that the value of any feasible flow is at least the value of any cut. Its proof is based on the following lemma.

Lemma 3. *Let η be any flow (not necessarily feasible) and $\mathcal{C}$ be any source-sink cut. Then,*

$$\varphi(\eta) = \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C})} \eta_h(s,a) - \sum_{(s,a,h)\in\mathcal{E}_{\text{bw}}(\mathcal{C})} \eta_h(s,a).$$

Proof. We have

$$\begin{aligned}
\varphi(\eta) &\overset{(a)}{=} \sum_{a\in\mathcal{A}_1(s_1)} \eta_1(s_1,a) \\
&\overset{(b)}{=} \sum_{(s,h)\in\mathcal{C}} \left(\sum_{a\in\mathcal{A}_h(s)} \eta_h(s,a) - \sum_{(s',a')\in\mathcal{I}_h(s)} \eta_{h-1}(s',a') \right) \\
&\overset{(c)}{=} \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C})} \eta_h(s,a) - \sum_{(s,a,h)\in\mathcal{E}_{\text{bw}}(\mathcal{C})} \eta_h(s,a).
\end{aligned}$$

where (a) is from the definition of flow value, (b) uses the balance constraints together with $(s_1,1) \in \mathcal{C}$, and (c) uses that the flow through every arc with both endpoints in $\mathcal{C}$ cancels since it appears once in the left term and once in the right one. □

Lemma 4. *Let η be any feasible flow and $\mathcal{C}$ be any source-sink cut. Then,*

$$\varphi(\eta) \geq \psi(\mathcal{C}, \underline{c}, \overline{c}).$$

Proof. We have

$$\begin{aligned}
\varphi(\eta) &\overset{(a)}{=} \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C})} \eta_h(s,a) - \sum_{(s,a,h)\in\mathcal{E}_{\text{bw}}(\mathcal{C})} \eta_h(s,a) \\
&\overset{(b)}{\geq} \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C})} \underline{c}_h(s,a) - \sum_{(s,a,h)\in\mathcal{E}_{\text{bw}}(\mathcal{C})} \overline{c}_h(s,a) \\
&\overset{(c)}{=} \psi(\mathcal{C}, \underline{c}, \overline{c}),
\end{aligned}$$

where (a) follows from Lemma 3, (b) uses the feasibility constraints, and (c) uses the definition of value of a cut. □

Thanks to Lemma 4, we know that, if we find a flow and a cut whose values coincide, then it must be that we found a minimum flow and its value coincides with the one of a maximum cut. This is what is shown in the next theorem.

Theorem 4 (Theorem 1.1 of [10])**.** *If there exists a feasible flow, the value of the minimum flow with non-negative lower bounds $\underline{c}$ equals the value of the maximum source-sink cut.*

Theorem 5 (Theorem 1.2 of [10])**.** *A feasible flow η is minimum if, and only if, the residual graph $\mathcal{G}_\eta(\mathcal{M})$ contains not decreasing path (i.e., no path from source to sink).*

Note that Theorem 4 is the equivalent of Theorem 6 stated in Section 2 for DAGs with unbounded capacities. We prove both theorems in the following unified proof.

Proof of Theorem 4 and Theorem 5. Let η be a feasible flow. We prove both theorems by showing that the following three statements are equivalent:

1. there exists a cut $\mathcal{C}$ such that $\varphi(\eta) = \psi(\mathcal{C}, \underline{c}, \overline{c})$;
2. η is a minimum flow;
3. there is no path from source to sink in $\mathcal{G}_\eta(\mathcal{M})$.

Note that, by Lemma 4 we clearly have that $1 \implies 2$. In fact, if a stricly better flow that η existed, call it η', then we would have

$$\varphi(\eta') < \varphi(\eta) = \psi(\mathcal{C}, \underline{c}, \overline{c}),$$

which is a contradiction since $\varphi(\eta') \geq \psi(\mathcal{C}, \underline{c}, \overline{c})$ by Lemma 4.

Clearly, $2 \implies 3$ by definition of decreasing path. If a path from source to sink existed in $\mathcal{G}_\eta(\mathcal{M})$, then we could decrease the flow along it while still satisfying all constraints. Hence, η would not be a minimum flow, which is a contradiction.

It remains to prove that $3 \implies 1$. We do so by building an explicit cut $\mathcal{C}$ from the residual graph $\mathcal{G}_\eta(\mathcal{M})$ which satifies property 1. This uses the same construction as in the well-known proof of the max-flow-min-cut theorem. Suppose that η is a feasible flow with no decreasing paths in $\mathcal{G}_\eta(\mathcal{M})$. Let $\mathcal{C}$ be the set of nodes that are reachable from $(s_1, 1)$ in $\mathcal{G}_\eta(\mathcal{M})$. It must be that $(s_1, 1) \in \mathcal{C}$ and $(s_{H+1}, H+1) \notin \mathcal{C}$ since the sink node is unreachable from the source node in $\mathcal{G}_\eta(\mathcal{M})$. Therefore, $\mathcal{C}$ is a valid cut. From Lemma 3, we know that

$$\varphi(\eta) = \sum_{(s,a,h) \in \mathcal{E}(\mathcal{C})} \eta_h(s,a) - \sum_{(s,a,h) \in \mathcal{E}_{\text{bw}}(\mathcal{C})} \eta_h(s,a).$$

It only remains to prove that $\eta_h(s,a) = \underline{c}_h(s,a)$ for all $(s,a,h) \in \mathcal{E}(\mathcal{C})$ and $\eta_h(s,a) = \overline{c}_h(s,a)$ for all $(s,a,h) \in \mathcal{E}_{\text{bw}}(\mathcal{C})$.

Take any $(s,a,h) \in \mathcal{E}(\mathcal{C})$. Since $(s,h) \in \mathcal{C}$ and $(f_h(s,a), h+1) \notin \mathcal{C}$, we must have that the forward arc (s,a,h) does not belong to $\mathcal{G}_\eta(\mathcal{M})$. But this means that $\rho_h(s,a) = 0$, which in turns implies that $\eta_h(s,a) = \underline{c}_h(s,a)$. This proves the first claim.

Now take any $(s,a,h) \in \mathcal{E}_{\text{bw}}(\mathcal{C})$. Here we have the opposite situation: $(f_h(s,a), h+1) \in \mathcal{C}$ but $(s,h) \notin \mathcal{C}$. This means that there is no arc from the first node to the second in $\mathcal{G}_\eta(\mathcal{M})$. But if $\eta_h(s,a) < \overline{c}_h(s,a)$ we would have $\rho_h^{\text{bw}}(s,a) > 0$ and thus there would be a backward arc between those two nodes. This is a contradiction, and thus it must be that $\eta_h(s,a) = \overline{c}_h(s,a)$. This concludes the proof of $3 \implies 1$, which in turns proves the main theorems. □

B.2 Layered DAGs with unlimited capacity

In all our applications, we will consider DAGs with unlimited capacity, i.e., $\overline{c}_h(s,a) = \infty$ for all $s \in \mathcal{S}, a \in \mathcal{A}, h \in [H]$. In this case, some of the previously-introduced quantities can be simplified using the notation adopted in Section 2. First, we can simplify the notation for a minimum flow $\varphi^\star(\underline{c})$ and for the value of a cut $\psi(\mathcal{C}, \underline{c})$ by dropping $\overline{c}$. The upper-bound constraints in the definition of feasible flow and in the LP can be simply dropped. Moreover, backward arcs in the residual graph have always residual equal to ∞. This means that backward arcs are always present in the residual graph, which has the intuitive meaning that we can always arbitrarily increase the flow along each forward arc in the original graph.

Now note that, by definition of value of a cut, if a cut $\mathcal{C}$ contains an available backward arc (i.e., one of the arcs in the original graph connects a node outside the cut with a node inside the cut), its value is $-\infty$. Therefore, if a feasible flow exists (whose value must be non-negative), then, by Theorem 6, we know that a cut $\mathcal{C}$ with backward arcs cannot be a maximum cut. Therefore, we can define the set of *valid cuts* $\mathfrak{C}$ as

$$\mathfrak{C} := \{\mathcal{C} \subseteq \mathcal{N} \setminus \{(s_{H+1}, H+1)\} \mid (s_1, 1) \in \mathcal{C}, \mathcal{E}_{\text{bw}}(\mathcal{C}) = \emptyset\}.$$

Then, for all $\mathcal{C} \in \mathfrak{C}$, we clearly have $\psi(\mathcal{C}, \underline{c}) = \sum_{(s,a,h) \in \mathcal{E}(\mathcal{C})} \underline{c}_h(s,a)$ since there is no backward arc. Moreover, by Theorem 4 together with the fact that cuts not belonging to $\mathfrak{C}$ cannot be maximizers,

$$\varphi^\star(\underline{c}) = \max_{\mathcal{C} \in \mathfrak{C}} \psi(\mathcal{C}, \underline{c}) = \max_{\mathcal{C} \in \mathfrak{C}} \sum_{(s,a,h) \in \mathcal{E}(\mathcal{C})} \underline{c}_h(s,a).$$

We formally state this result in the following theorem.

Theorem 6. *Consider a layered DAG with unlimited capacity. If there exists a feasible flow,*

$$\varphi^\star(\underline{c}) = \max_{\mathcal{C} \in \mathfrak{C}} \psi(\mathcal{C}, \underline{c}).$$

Useful properties We prove some simple properties of flows and cuts which will be useful later on.

Lemma 5 (Monotonicity)**.** *Let $\underline{c}^1, \underline{c}^2 : \mathcal{E} \to [0, \infty)$ be such that $\underline{c}^1_h(s,a) \leq \underline{c}^2_h(s,a)$ for all $(s,a,h) \in \mathcal{E}$. Then,*

$$\varphi^\star(\underline{c}^1) \leq \varphi^\star(\underline{c}^2).$$

Proof. This can be immediately seen from the LP formulation: any feasible flow η for $\underline{c}^2$ is also feasible for $\underline{c}^1$. □

Lemma 6 (Flow bounds)**.** *For any lower bound function $\underline{c}$,*

$$\max_{h\in[H]} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} \underline{c}_h(s,a) \leq \varphi^\star(\underline{c}) \leq \sum_{h\in[H]} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} \underline{c}_h(s,a).$$

Proof. Both inequalities are easy to see from Theorem 6 and the definition of value of a cut. The upper bound is trivial since $\varphi^\star(\underline{c}) = \max_{\mathcal{C}\in\mathfrak{C}} \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C})} \underline{c}_h(s,a)$ and the set of outgoing arcs $\mathcal{E}(\mathcal{C})$ from a cut $\mathcal{C}$ can contain at most all possible arcs $\mathcal{E}$. To see the lower bound, note that $\mathcal{C}_h := \{s \in \mathcal{S}_l : l \leq h\}$ is a valid cut for any $h \in [H]$ whose outgoing arcs are all those connecting states at stage h with states at stage $h+1$. Thus,

$$\varphi^\star(\underline{c}) = \max_{\mathcal{C}\in\mathfrak{C}} \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C})} \underline{c}_h(s,a) \geq \max_{h\in[H]} \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C}_h)} \underline{c}_h(s,a) = \max_{h\in[H]} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} \underline{c}_h(s,a).$$

□

Lemma 7. *Let $\underline{c}^1, \underline{c}^2$ be two non-negative lower bound functions and $\alpha > 0$. Then,*

$$\varphi^\star(\alpha\underline{c}^1 + \underline{c}^2) \leq \alpha\varphi^\star(\underline{c}^1) + \varphi^\star(\underline{c}^2).$$

Proof. We first prove that $\varphi^\star(\alpha\underline{c}^1) = \alpha\varphi^\star(\underline{c}^1)$. This can be easily seen from the linear programming formulation stated above: by performing the change of variables $\eta'_h(s,a) = \eta_h(s,a)/\alpha$, we obtain exactly α times the value of a minimum flow with lower bound function $\underline{c}^1$. Next, we prove that $\varphi^\star(\underline{c}^1 + \underline{c}^2) \leq \varphi^\star(\underline{c}^1) + \varphi^\star(\underline{c}^2)$. Let η^1 and η^2 be minimum flows for the problems with lower bounds $\underline{c}^1$ and $\underline{c}^2$, respectively. The proof follows by noting that $\eta := \eta^1 + \eta^2$ is a feasible flow for the problem with lower bound function $\underline{c}^1 + \underline{c}^2$ and that is value is exactly $\varphi(\eta) = \varphi(\eta^1) + \varphi(\eta^2) = \varphi^\star(\underline{c}^1) + \varphi^\star(\underline{c}^2)$. Combining these two results concludes the proof. □

B.3 Minimum flows and minimum policy covers

A crucial problem in the analysis of our sampling rules is the problem of computing a *minimum policy cover*. Formally, given a subset $\mathcal{E}' \subseteq \mathcal{E}$ of the arcs (i.e., of the state-action-stage triplets), the goal is to find a set of policies $\Pi_{\text{cover}} \subseteq \Pi$ of *minimum size* such that

$$\forall (s,a,h) \in \mathcal{E}', \exists \pi \in \Pi_{\text{cover}} : (s^\pi_h, a^\pi_h) = (s,a).$$

That is, Π_{cover} is the smallest set of policies that, played together, visit all arcs in $\mathcal{E}'$. This problem can be easily reduced to a minimum flow problem with lower bound function

$$\underline{c}_h(s,a) := \mathbb{1}\left((s,a,h) \in \mathcal{E}'\right),$$

which intuitively demands at least one visit to all $(s,a,h) \in \mathcal{E}'$, and zero visits from the other triplets. Moreover, since $\underline{c}$ is integer-valued, an integer minimum flow exists which can be computed by existing algorithms [e.g., 7]. Suppose that η is one such integer minimum flow. A policy cover can be easily extracted from it by the procedure shown in Algorithm 2, which is similar to the method proposed by [7] to obtain a minimum path cover in a layered DAG.

Algorithm 2 Extract policy cover from minimum flow

Input: deterministic MDP (without reward) $\mathcal{M} := (\mathcal{S}, \mathcal{A}, \{f_h\}_{h\in[H]}, s_1, H)$, feasible integer flow η
Initialize $\Pi_{\text{cover}} \leftarrow \emptyset$
while $\varphi(\eta) > 0$ **do**
  Initialize a policy π with arbitrary actions
  for $h = 1, \ldots, H$ **do**
    $\pi_h(s_h) \leftarrow \arg\max_{a\in\mathcal{A}_h(s_h)} \eta_h(s, a)$
    $\eta_h(s_h, \pi_h(s_h)) \leftarrow \eta_h(s_h, \pi_h(s_h)) - 1$
    $s_{h+1} \leftarrow f_h(s_h, \pi_h(s_h))$
  end for
  $\Pi_{\text{cover}} \leftarrow \Pi_{\text{cover}} \cup \{\pi\}$
end while

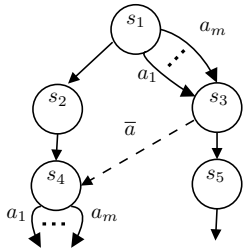

Figure 3: Example to show why eliminating actions from the sampling rule is not a good idea. The size of a minimum policy cover is $m + 1$ when $\bar{a}$ is used and $2m$ when it is unavailable.

Eliminating arcs In our applications, we compute minimum policy covers for subsets of the arcs $\mathcal{E}'$ that contain only non-eliminated actions. One natural question is: what if, instead of setting the lower bound function to zero for arcs not in $\mathcal{E}'$, we use a lower bound function that is uniformly equal to one but solve the minimum flow problem on a sub-graph with only arcs in $\mathcal{E}'$ available? One argument against this idea is that the resulting minimum policy cover might have a strictly larger size. Figure 3 shows an example. Suppose that we want to compute a minimum policy cover visiting all arcs except $\bar{a}$. Then, if we set the lower bound function for $\bar{a}$ to zero and solve the minimum flow problem on the full MDP, we get an optimal value of $1 + m$ by sending a flow of one on each action in s_1 and then redirecting $m - 1$ flow to $\bar{a}$. If instead we make $\bar{a}$ unavailable, we cannot do this trick and the optimal flow becomes $2m$.

C Lower Bounds

We first present an important result to derive lower bounds, a change-of-distribution lemma which is a direct instantiation of Lemma 1 in [28] (see also Lemma 8 in [42] and references therein).

Lemma 8. *Let $\mathcal{M}$ and $\widetilde{\mathcal{M}}$ be two MDPs with identical state-action space and deterministic transitions but possibly different rewards distributions denoted by $(\nu_h^{\mathcal{M}}(s,a))_{s,a,h}$ and $(\nu_h^{\widetilde{\mathcal{M}}}(s,a))_{s,a,h}$ respectively. For every algorithm, every stopping time τ and every event $\mathcal{E} \in \mathcal{F}_\tau$, it holds that*

$$\sum_{h=1}^{H} \sum_{s \in \mathcal{S}_h} \sum_{a \in \mathcal{A}_h} \mathbb{E}_{\mathcal{M}}[n_h^\tau(s,a)] \mathrm{KL}\left(\nu_h^{\mathcal{M}}(s,a), \nu_h^{\widetilde{\mathcal{M}}}(s,a)\right) \geq \mathrm{kl}\left(\mathbb{P}_{\mathcal{M}}(\mathcal{E}), \mathbb{P}_{\widetilde{\mathcal{M}}}(\mathcal{E})\right)$$

where KL *denotes the Kullback-Leibler divergence and* $\mathrm{kl}(x,y) = x\log(x/y) + (1-x)\log((1-x)/(1-y))$ *the binary relative entropy.*

If the rewards follow Gaussian distributions with variance σ^2, we have

$$\mathrm{KL}\left(\nu_h^{\mathcal{M}}(s,a), \nu_h^{\widetilde{\mathcal{M}}}(s,a)\right) = \frac{\left(r_h^{\mathcal{M}}(s,a) - r_h^{\widetilde{\mathcal{M}}}(s,a)\right)^2}{2\sigma^2}.$$

C.1 Instance-dependent lower bound

In this section, we prove Theorem 1. We first state and prove three lemmas which bound the local number of visits to different (s,a,h). Then, we combine them to prove the main result.

Lemma 9 (Lower bound for sub-optimal pairs)**.** *For any $h \in [H]$ and any non-ε-optimal pair $(s,a) \notin \mathcal{Z}_h^\varepsilon$,*

$$\mathbb{E}[n_h^\tau(s,a)] \geq \frac{2\sigma^2 \log(1/3\delta)}{(\overline{\Delta}_h(s,a) + \varepsilon)^2}.$$

Proof. Consider the alternative MDP $\widetilde{\mathcal{M}} := (\mathcal{S}, \mathcal{A}, \{f_h, \widetilde{\nu}_h\}_{h\in[H]}, s_1, H)$ which is equivalent to $\mathcal{M}$ except that the reward is modified only at (s,a,h) as $\widetilde{\nu}_h(s,a) = \mathcal{N}(r_h(s,a) + \Delta, \sigma^2)$ with $\Delta > \overline{\Delta}_h(s,a) + \varepsilon$, while $\widetilde{\nu}_{h'}(s',a') = \nu_{h'}(s',a')$ on all other state-action-stage triplets. It is easy to see that, for any $\pi \in \Pi_{s,a,h}$,

$$\widetilde{V}_1^\pi(s_1) = \sum_{l=1}^{H} \widetilde{r}_l(s_l^\pi, a_l^\pi) = \sum_{l=1}^{H} r_l(s_l^\pi, a_l^\pi) + \Delta = V_1^\pi(s_1) + \Delta.$$

Similarly, for any policy $\pi \notin \Pi_{s,a,h}$, we have $\widetilde{V}_1^\pi(s_1) = V_1^\pi(s_1)$. Let $\pi^0 \in \arg\max_{\pi \in \Pi_{s,a,h}} V_1^\pi(s_1)$. Then,

$$\begin{aligned}
\widetilde{V}_1^{\pi^0}(s_1) &= V_1^{\pi^0}(s_1) + \Delta \\
&> V_1^{\pi^0}(s_1) + \overline{\Delta}_h(s,a) + \varepsilon \\
&= V_1^\star(s_1) + \varepsilon \\
&\geq \max_{\pi \notin \Pi_{s,a,h}} V_1^\pi(s_1) + \varepsilon \\
&= \max_{\pi \notin \Pi_{s,a,h}} \widetilde{V}_1^\pi(s_1) + \varepsilon.
\end{aligned}$$

This means that[5] $\mathbb{P}_{\widetilde{\mathcal{M}}}(\widehat{\pi} \in \Pi_{s,a,h}) \geq 1 - \delta$. Moreover, $\mathbb{P}_{\mathcal{M}}(\widehat{\pi} \in \Pi_{s,a,h}) \leq \delta$ since (s,a,h) is not visited by any ε-optimal policy. Therefore, Lemma 8 implies that

$$\mathbb{E}[n_h^\tau(s,a)] \geq \frac{2\sigma^2}{\Delta^2} \mathrm{kl}(\mathbb{P}_{\mathcal{M}}(\widehat{\pi} \in \Pi_{s,a,h}), \mathbb{P}_{\widetilde{\mathcal{M}}}(\widehat{\pi} \in \Pi_{s,a,h})) \geq \frac{2\sigma^2}{\Delta^2} \mathrm{kl}(\delta, 1-\delta) \geq \frac{2\sigma^2}{\Delta^2} \log(1/3\delta).$$

This holds for any $\Delta > \overline{\Delta}_h(s,a) + \varepsilon$ and the proof is concluded by taking the limit. □

[5] Recall that we use $\widehat{\pi}$ to denote the policy returned by the recommendation rule of the algorithm

Lemma 10 (Lower bound for non-unique ε-optimal pairs)**.** *For any $h \in [H]$ and any ε-optimal pair $(s,a) \in \mathcal{Z}_h^\varepsilon$, if $|\mathcal{Z}_h^\varepsilon| > 1$,*

$$\mathbb{E}[n_h^\tau(s,a)] \geq \frac{\sigma^2 \log(1/4\delta)}{4\varepsilon^2}.$$

Proof. Take any pair (s,a) in $\mathcal{Z}_h^\varepsilon$. We distinguish two cases.

Case 1: $\mathbb{P}_{\mathcal{M}}(\widehat{\pi} \in \Pi_{s,a,h}) \leq 1/2$**.** We can build the same alternative MDP $\widetilde{\mathcal{M}}$ as in the proof of Lemma 9, for which we have $\mathbb{P}_{\widetilde{\mathcal{M}}}(\widehat{\pi} \in \Pi_{s,a,h}) \geq 1-\delta$. Thus, using Lemma 8,

$$\begin{aligned}
\mathbb{E}[n_h^\tau(s,a)] &\geq \frac{2\sigma^2}{\Delta^2}\mathrm{kl}(\mathbb{P}_{\mathcal{M}}(\widehat{\pi} \in \Pi_{s,a,h}), \mathbb{P}_{\widetilde{\mathcal{M}}}(\widehat{\pi} \in \Pi_{s,a,h})) \\
&\geq \frac{2\sigma^2}{\Delta^2}\mathrm{kl}(1/2, 1-\delta) \\
&= \frac{2\sigma^2}{\Delta^2}\mathrm{kl}(1/2, \delta) \\
&\geq \frac{\sigma^2}{\Delta^2}\log(1/4\delta),
\end{aligned}$$

where we used the fact that $\mathrm{kl}(x,y) = \mathrm{kl}(1-x, 1-y)$ and $\mathrm{kl}(x,y) \geq x\log(1/y) - \log(2)$. By taking the limit $\Delta \to \overline{\Delta}_h(s,a) + \varepsilon$ and using $\overline{\Delta}_h(s,a) \leq \varepsilon$, we thus conclude that $\mathbb{E}[n_h^\tau(s,a)] \geq \frac{\sigma^2 \log(1/4\delta)}{(\overline{\Delta}_h(s,a)+\varepsilon)^2} \geq \frac{\sigma^2 \log(1/4\delta)}{4\varepsilon^2}$.

Case 2: $\mathbb{P}_{\mathcal{M}}(\widehat{\pi} \in \Pi_{s,a,h}) > 1/2$**.** Consider the alternative MDP $\widetilde{\mathcal{M}} := (\mathcal{S}, \mathcal{A}, \{f_h, \widetilde{\nu}_h\}_{h\in[H]}, s_1, H)$ which is equivalent to $\mathcal{M}$ except that the reward is modified only at (s,a,h) as $\widetilde{\nu}_h(s,a) = \mathcal{N}(r_h(s,a) - \Delta, \sigma^2)$ with $\Delta > 2\varepsilon - \overline{\Delta}_h(s,a)$, while $\widetilde{\nu}_{h'}(s',a') = \nu_{h'}(s',a')$ on all other state-action-stage triplets. Then, for $\pi^0 \in \underset{\pi \in \Pi \setminus \Pi_{s,a,h}}{\arg\max}\, V_1^\pi(s_1)$,

$$\begin{aligned}
\widetilde{V}_1^{\pi^0}(s_1) = V_1^{\pi^0}(s_1) &\geq V_1^\star(s_1) - \varepsilon \pm \max_{\pi \in \Pi_{s,a,h}} V_1^\pi(s_1) \\
&= \overline{\Delta}_h(s,a) - \varepsilon + \max_{\pi \in \Pi_{s,a,h}} V_1^\pi(s_1) \\
&= \overline{\Delta}_h(s,a) - \varepsilon + \max_{\pi \in \Pi_{s,a,h}} \widetilde{V}_1^\pi(s_1) + \Delta > \max_{\pi \in \Pi_{s,a,h}} \widetilde{V}_1^\pi(s_1) + \varepsilon,
\end{aligned}$$

where the first inequality is due to the fact that, since $\mathcal{Z}_h^\varepsilon > 1$, there exists at least one ε-optimal policy which does not visit (s,a) at step h (i.e., which belongs to $\Pi \setminus \Pi_{s,a,h}$). This implies that $\mathbb{P}_{\widetilde{\mathcal{M}}}(\widehat{\pi} \in \Pi_{s,a,h}) \leq \delta$. Thus, using Lemma 8,

$$\mathbb{E}[n_h^\tau(s,a)] \geq \frac{2\sigma^2}{\Delta^2}\mathrm{kl}(\mathbb{P}_{\mathcal{M}}(\widehat{\pi} \in \Pi_{s,a,h}), \mathbb{P}_{\widetilde{\mathcal{M}}}(\widehat{\pi} \in \Pi_{s,a,h})) \geq \frac{2\sigma^2}{\Delta^2}\mathrm{kl}(1/2, \delta) \geq \frac{\sigma^2}{\Delta^2}\log(1/4\delta).$$

By taking the limit $\Delta \to 2\varepsilon - \overline{\Delta}_h(s,a)$, we thus conclude that $\mathbb{E}[n_h^\tau(s,a)] \geq \frac{\sigma^2 \log(1/4\delta)}{(2\varepsilon - \overline{\Delta}_h(s,a))^2} \geq \frac{\sigma^2 \log(1/4\delta)}{4\varepsilon^2}$. □

Lemma 11 (Lower bound for unique ε-optimal pairs)**.** *For any $h \in [H]$ and any ε-optimal pair $(s,a) \in \mathcal{Z}_h^\varepsilon$, if $|\mathcal{Z}_h^\varepsilon| = 1$,*

$$\mathbb{E}[n_h^\tau(s,a)] \geq \frac{2\sigma^2 \log(1/3\delta)}{(\overline{\Delta}_{\min}^h + \varepsilon)^2},$$

where $\overline{\Delta}_{\min}^h := \min_{(s',a'):\overline{\Delta}_h(s',a')>0} \overline{\Delta}_h(s',a')$.

Proof. Note that, since $(s,a) \in \mathcal{Z}_h^\varepsilon$ and $|\mathcal{Z}_h^\varepsilon| = 1$, then $\Pi^\varepsilon \cap (\Pi \setminus \Pi_{s,a,h}) = \emptyset$ (i.e., all ε-optimal policies visit (s,a,h)). Therefore, $\mathbb{P}_{\mathcal{M}}(\widehat{\pi} \in \Pi_{s,a,h}) \geq 1-\delta$. We now use a construction similar to the one in Case 2 of the proof of Lemma 10.

Consider the alternative MDP $\widetilde{\mathcal{M}} := (\mathcal{S}, \mathcal{A}, \{f_h, \widetilde{\nu}_h\}_{h\in[H]}, s_1, H)$ which is equivalent to $\mathcal{M}$ except that the reward is modified only at (s,a,h) as $\widetilde{\nu}_h(s,a) = \mathcal{N}(r_h(s,a) - \Delta, \sigma^2)$ with $\Delta > \max_{\pi\in\Pi_{s,a,h}} V_1^\pi(s_1) - \max_{\pi\in\Pi\setminus\Pi_{s,a,h}} V_1^\pi(s_1) + \varepsilon$, while $\widetilde{\nu}_{h'}(s',a') = \nu_{h'}(s',a')$ on all other state-action-stage triplets. Then, for $\pi^0 \in \operatorname*{arg\,max}_{\pi\in\Pi\setminus\Pi_{s,a,h}} V_1^\pi(s_1)$,

$$\begin{aligned}
\widetilde{V}_1^{\pi^0}(s_1) = V_1^{\pi^0}(s_1) &= \max_{\pi\in\Pi\setminus\Pi_{s,a,h}} V_1^\pi(s_1) \pm \max_{\pi\in\Pi_{s,a,h}} V_1^\pi(s_1) \\
&= \max_{\pi\in\Pi\setminus\Pi_{s,a,h}} V_1^\pi(s_1) - \max_{\pi\in\Pi_{s,a,h}} V_1^\pi(s_1) + \max_{\pi\in\Pi_{s,a,h}} \widetilde{V}_1^\pi(s_1) + \Delta \\
&> \max_{\pi\in\Pi_{s,a,h}} \widetilde{V}_1^\pi(s_1) + \varepsilon.
\end{aligned}$$

This implies that $\mathbb{P}_{\widetilde{\mathcal{M}}}(\widehat{\pi} \in \Pi_{s,a,h}) \le \delta$. Thus, applying Lemma 8,

$$\mathbb{E}[n_h^\tau(s,a)] \ge \frac{2\sigma^2}{\Delta^2}\mathrm{kl}(\mathbb{P}_{\mathcal{M}}(\widehat{\pi} \in \Pi_{s,a,h}), \mathbb{P}_{\widetilde{\mathcal{M}}}(\widehat{\pi} \in \Pi_{s,a,h})) \ge \frac{2\sigma^2}{\Delta^2}\mathrm{kl}(1-\delta,\delta) \ge \frac{2\sigma^2}{\Delta^2}\log(1/3\delta).$$

Now note that, since an optimal policy belongs to $\Pi_{s,a,h}$,

$$\begin{aligned}
\max_{\pi\in\Pi_{s,a,h}} V_1^\pi(s_1) - \max_{\pi\in\Pi\setminus\Pi_{s,a,h}} V_1^\pi(s_1) &= V_1^\star(s_1) - \max_{\pi\in\Pi\setminus\Pi_{s,a,h}} V_1^\pi(s_1) \\
&= V_1^\star(s_1) - \max_{s'\in\mathcal{S}_h} \max_{a'\in\mathcal{A}_h(s'):(s',a')\neq(s,a)} \max_{\pi\in\Pi_{s',a',h}} V_1^\pi(s_1) \\
&= \min_{(s',a'):\overline{\Delta}_h(s',a')>0} \overline{\Delta}_h(s',a') = \overline{\Delta}_{\min}^h.
\end{aligned}$$

By taking the limit $\Delta \to \overline{\Delta}_{\min}^h + \varepsilon$, we conclude that $\mathbb{E}[n_h^\tau(s,a)] \ge \frac{2\sigma^2\log(1/3\delta)}{(\overline{\Delta}_{\min}^h+\varepsilon)^2}$. □

We are now ready to prove the main theorem.

Proof of Theorem 1. The first statement follows easily from Lemma 9, Lemma 10, and Lemma 11. In fact, for $(s,a) \notin \mathcal{Z}_h^\varepsilon$, Lemma 9 yields

$$\mathbb{E}[n_h^\tau(s,a)] \ge \frac{2\sigma^2\log(1/3\delta)}{(\overline{\Delta}_h(s,a)+\varepsilon)^2} \ge \frac{\sigma^2\log(1/3\delta)}{2\max(\overline{\Delta}_h(s,a),\varepsilon)^2} = \frac{\sigma^2\log(1/3\delta)}{2\max(\overline{\Delta}_h(s,a),\overline{\Delta}_{\min}^h,\varepsilon)^2}$$

since $\overline{\Delta}_h(s,a) \le \overline{\Delta}_{\min}^h$. For $(s,a) \in \mathcal{Z}_h^\varepsilon$ when $|\mathcal{Z}_h^\varepsilon| > 1$, the result follows trivially from Lemma 10 by noting that $\overline{\Delta}_{\min}^h = 0$ and $\overline{\Delta}_h(s,a) \le \varepsilon$. For $(s,a) \in \mathcal{Z}_h^\varepsilon$ when $|\mathcal{Z}_h^\varepsilon| = 1$, using Lemma 11 with $\overline{\Delta}_h(s,a) \le \varepsilon$,

$$\mathbb{E}[n_h^\tau(s,a)] \ge \frac{2\sigma^2\log(1/3\delta)}{(\overline{\Delta}_{\min}^h+\varepsilon)^2} \ge \frac{\sigma^2\log(1/3\delta)}{2\max(\overline{\Delta}_{\min}^h,\varepsilon)^2} = \frac{\sigma^2\log(1/3\delta)}{2\max(\overline{\Delta}_h(s,a),\overline{\Delta}_{\min}^h,\varepsilon)^2}.$$

To prove the second statement, note that the visitation counts $n_h^\tau(s,a)$ form a feasible flow. Therefore, given local lower bounds $\underline{c}_h(s,a)$ for each state-action-stage triplet (s,a,h), the minimum expected stopping when satisfying all such lower bounds is exactly the value of the minimum flow $\varphi^\star(\underline{c})$. □

C.2 Worst-case lower bound

In order to derive a worst-case lower bound, we build a deterministic variant of the hard MDP instance introduced by [17] for the time-inhomogeneous stochastic setting. A major complication is that stochasticity plays a crucial role for obtaining the right minimax dependence on the horizon H in the latter context. Here we present a different analysis where we shall achieve the optimal dependence by leveraging the theory of minimum flows.

An example of our hard instance is shown in Figure 4. Fix some $S \ge 2$, $A \ge 2$, and $H \ge 3\log_2(S)$. As common in existing worst-case lower bounds for MDPs [33, 17], we arrange $S-1$ states in a full binary tree. As such, we will assume that $S - 1 = \sum_{i=0}^{d-1} 2^i = 2^d - 1$ for some integer $d \ge 1$ which represents the depth of the tree. The condition $H \ge 3\log_2(S)$ is to make sure that there are enough

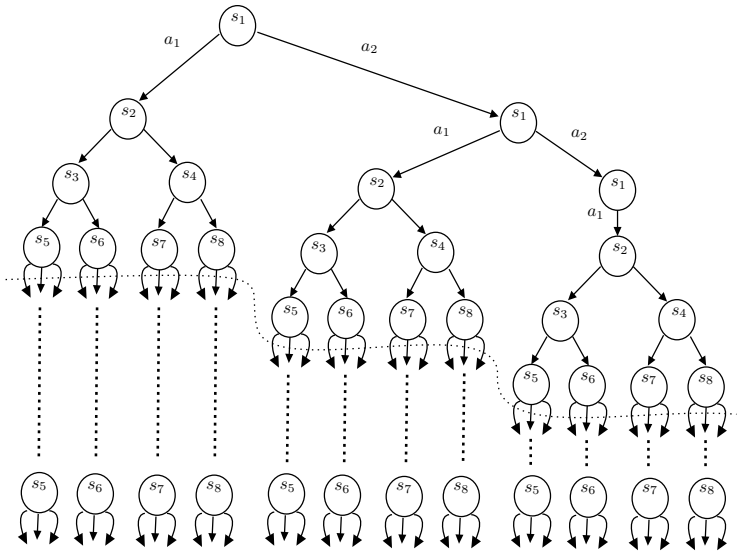

Figure 4: Example of hard instance with $S = 8$, $A = 3$, and $\overline{H} = 3$. The mean reward is zero everywhere, and its distribution is Gaussian with unit variance.

stages to build the binary tree. These assumptions are made only to simplify the exposition, while our result can be easily generalized to any number of states and stages by considering non-complete binary trees (see also Appendix D of [17]).

The starting state s_1 has two available actions, a_1 and a_2. Action a_1 makes the agent transition to the root s_2 of the binary tree containing all other $S - 1$ states. When taking action a_2, the agent remains in state s_1. Such an action is only available up to stage $\bar{H} - 1$, where the value of $\bar{H}$ will be specified later. In stage $\bar{H}$, only a_1 is available and the agent must thus transition to state s_2. In other words, the agent can reach the root of the binary tree s_2 from stage $h = 2$ (when taking a_1 at stage 1) to stage $h = \bar{H} + 1$ (when taking action a_2 for all $h = 1, \dots, \bar{H} - 1$ and a_1 in stage $\bar{H}$). In the leaf states of the binary tree, all A actions are available whose effect effect is to keep the agent in the same state until the final stage.

The intuition behind the hardness of this MDP instance is as follows. In order to learn a near-optimal policy, the agent must figure out which leaf state-action pair of the tree to reach (roughly SA choices) and at which stage to reach it (exactly $\bar{H}$ choices). In graph-theoretical words, any flow leaving the initial state must pass through one of the tree leaves at one stage from $1 + d$ to $\bar{H} + d$. If we "cut" the tree at those state-action-stage triplets (see the dashed line in Figure 4), we have that the value of the flow (and thus τ) can be written as the sum of visits to $\Omega(SA\bar{H})$ triplets. By constructing variants of this hard MDP where we raise or lower the reward of some of these triplets by roughly $\varepsilon/\overline{H}$, we can prove (Lemma 12 stated in Appendix C) that each of these triplets needs to be explored roughly $\Omega(\frac{\bar{H}}{\varepsilon^2}\log(1/\delta))$ times. Summing them up and choosing $\bar{H} \geq \Omega(H)$ yields that τ must be at least $\Omega(\frac{SAH^2}{\varepsilon^2}\log(1/\delta))$.

Theorem 7. *For any $S, A \geq 2$ and $H \geq 3\log_2(S)$ such that $S = 2^d$ for some integer $d \geq 1$, there exists an MDP $\mathcal{M}$ with S states, A actions, and H stages such that any algorithm which is (ε, δ)-PAC on the class $\mathfrak{M}_1$ must satisfy*

$$\mathbb{E}_{\mathcal{M}}[\tau] \geq \frac{SAH^2}{72\varepsilon^2}\log(1/4\delta).$$

Now we state a result that is key in the proof of Theorem 7.

Lemma 12. *Let $\mathcal{M}$ be any MDP with $r_h(s, a) = 0$ for all s, a, h. Then, for any $1 \leq \bar{h} \leq \bar{H} \leq H$ and any policy $\pi \in \Pi$,*

$$\sum_{h=\bar{h}}^{\bar{H}} \mathbb{E}[n_h^\tau(s_h^\pi, a_h^\pi)] \geq \frac{(\bar{H} - \bar{h} + 1)^2}{4\varepsilon^2}\log(1/4\delta).$$

Proof. Fix some $1 \le \bar{h} \le \bar{H} \le H$ and policy $\pi \in \Pi$. Define the event

$$E^{\pi} := \left\{ \sum_{h=\bar{h}}^{\bar{H}} \mathbb{1}\left((s_h^{\hat{\pi}}, a_h^{\hat{\pi}}) = (s_h^{\pi}, a_h^{\pi}) \right) \ge \frac{\bar{H} - \bar{h} + 1}{2} \right\}.$$

We distinguish two cases.

Case 1: $\mathbb{P}_{\mathcal{M}}(E^{\pi}) \le 1/2$**.** Consider the alternative MDP $\widetilde{\mathcal{M}} := (\mathcal{S}, \mathcal{A}, \{f_h, \widetilde{\nu}_h\}_{h\in[H]}, s_1, H)$ which is equivalent to $\mathcal{M}$ except that the reward is modified as

$$\widetilde{\nu}_h(s,a) = \begin{cases} \mathcal{N}(2\varepsilon/(\bar{H} - \bar{h} + 1), 1) & \text{if } (s,a) = (s_h^{\pi}, a_h^{\pi}), \bar{h} \le h \le \bar{H}, \\ \nu_h(s,a) & \text{otherwise.} \end{cases}$$

Note that, for any policy $\pi' \in \Pi$, since the mean reward in $\widetilde{\mathcal{M}}$ is non-zero only for state-action pairs visited by π in stages from $\bar{h}$ to $\bar{H}$,

$$\widetilde{V}_1^{\pi'}(s_1) = \frac{2\varepsilon}{\bar{H} - \bar{h} + 1} \sum_{h=\bar{h}}^{\bar{H}} \mathbb{1}\left((s_h^{\pi'}, a_h^{\pi'}) = (s_h^{\pi}, a_h^{\pi}) \right).$$

This implies that $\widetilde{V}_1^{\pi}(s_1) = \widetilde{V}_1^{\star}(s_1) = 2\varepsilon$. Moreover, $\widetilde{V}_1^{\pi'}(s_1) < \varepsilon$ if $\sum_{h=\bar{h}}^{\bar{H}} \mathbb{1}\left((s_h^{\pi'}, a_h^{\pi'}) = (s_h^{\pi}, a_h^{\pi}) \right) < \frac{\bar{H}-\bar{h}+1}{2}$. Therefore, the event E^{π} must have $\mathbb{P}_{\widetilde{\mathcal{M}}}(E^{\pi}) \ge 1 - \delta$, otherwise the returned policy would not be ε-optimal in $\widetilde{\mathcal{M}}$. Thus, applying Lemma 8,

$$\sum_{h=\bar{h}}^{\bar{H}} \mathbb{E}[n_h^{\tau}(s_h^{\pi}, a_h^{\pi})] \ge \frac{(\bar{H} - \bar{h} + 1)^2}{2\varepsilon^2} \mathrm{kl}(\mathbb{P}_{\mathcal{M}}(E^{\pi}), \mathbb{P}_{\widetilde{\mathcal{M}}}(E^{\pi})) \ge \frac{(\bar{H} - \bar{h} + 1)^2}{4\varepsilon^2} \log(1/4\delta),$$

where the second inequality uses the same steps as in the proof of Case 1 of Lemma 10.

Case 2: $\mathbb{P}_{\mathcal{M}}(E^{\pi}) > 1/2$**.** We use a similar construction as in Case 1, except that this time we build a new MDP by lowering the reward at pairs visited by π. Consider the alternative MDP $\widetilde{\mathcal{M}} := (\mathcal{S}, \mathcal{A}, \{f_h, \widetilde{\nu}_h\}_{h\in[H]}, s_1, H)$ which is equivalent to $\mathcal{M}$ except that the reward is modified as

$$\widetilde{\nu}_h(s,a) = \begin{cases} \mathcal{N}(-\Delta, 1) & \text{if } (s,a) = (s_h^{\pi}, a_h^{\pi}), \bar{h} \le h \le \bar{H}, \\ \nu_h(s,a) & \text{otherwise.} \end{cases}$$

Here $\Delta > 2\varepsilon/(\bar{H} - \bar{h} + 1)$. Note that $\widetilde{V}_1^{\star}(s_1) = 0$, which is attained by any policy not visiting state-action pairs visited by π from $\bar{h}$ to $\bar{H}$. Moreover, for any $\pi' \in \Pi$, $\widetilde{V}_1^{\pi'}(s_1) < -\varepsilon$ if $\sum_{h=\bar{h}}^{\bar{H}} \mathbb{1}\left((s_h^{\pi'}, a_h^{\pi'}) = (s_h^{\pi}, a_h^{\pi}) \right) \ge \frac{\bar{H}-\bar{h}+1}{2}$ and thus π' is not ε-optimal for $\widetilde{\mathcal{M}}$. Therefore, $\mathbb{P}_{\widetilde{\mathcal{M}}}(E^{\pi}) \le \delta$. Thus, applying Lemma 8,

$$\sum_{h=\bar{h}}^{\bar{H}} \mathbb{E}[n_h^{\tau}(s_h^{\pi}, a_h^{\pi})] \ge \frac{2}{\Delta^2} \mathrm{kl}(\mathbb{P}_{\mathcal{M}}(E^{\pi}), \mathbb{P}_{\widetilde{\mathcal{M}}}(E^{\pi})) \ge \frac{1}{\Delta^2} \log(1/4\delta),$$

where the second inequality uses the same steps as in the proof of Case 2 of Lemma 10. This holds for any $\Delta > 2\varepsilon/(\bar{H} - \bar{h} + 1)$ and the proof is concluded by taking the limit. □

We now prove the main theorem.

Proof of Theorem 7. Let us consider the hard instance described above and exemplified in Figure 4. Let us enumerate by $(\bar{s}_i)_{i=1}^n$ the leaf states of the binary tree. For any $i \in [n], a \in [A], j \in [H]$, let us denote by π^{iaj} the policy which chooses action a_1 in s_1 at stage j (and thus action a_2 in all stages before), which traverses the tree from stage $j+1$ to stage $j+d$, reaching the leaf state $\bar{s}_i$ and playing always action a thereafter.

Now fix some $j \in [\bar{H}]$. Applying Lemma 12 on the segment of stages from $j+d$ to $j+d+\bar{H}$,

$$\frac{1}{nA}\sum_{i=1}^{n}\sum_{a=1}^{A}\sum_{h=j+d}^{j+d+\bar{H}}\mathbb{E}[n_h^\tau(s_h^{\pi^{iaj}}, a_h^{\pi^{iaj}})] \geq \frac{(\bar{H}+1)^2}{4\varepsilon^2}\log(1/4\delta).$$

Since this holds for any $j \in [\bar{H}]$,

$$\sum_{j=1}^{\bar{H}}\sum_{i=1}^{n}\sum_{a=1}^{A}\sum_{h=j+d}^{j+d+\bar{H}}\mathbb{E}[n_h^\tau(s_h^{\pi^{iaj}}, a_h^{\pi^{iaj}})] \geq \frac{nA\bar{H}(\bar{H}+1)^2}{4\varepsilon^2}\log(1/4\delta).$$

Now note that the lefthand side can be equivalently written as

$$\sum_{j=1}^{\bar{H}}\sum_{i=1}^{n}\sum_{a=1}^{A}\sum_{h=j+d}^{j+d+\bar{H}}\mathbb{E}[n_h^\tau(s_h^{\pi^{iaj}}, a_h^{\pi^{iaj}})] = \sum_{l=0}^{\bar{H}}\sum_{j=1}^{\bar{H}}\sum_{i=1}^{n}\sum_{a=1}^{A}\mathbb{E}[n_{j+d+l}^\tau(s_{j+d+l}^{\pi^{iaj}}, a_{j+d+l}^{\pi^{iaj}})].$$

Each element of the outer sum over $l = 0, \dots, \bar{H}$ is exactly equal to $\mathbb{E}[\tau]$ since it is the value of a cut at depth l below each unrolled binary tree (e.g., the one shown in Figure 4 is for $l = 0$). Therefore,

$$\mathbb{E}[\tau] \geq \frac{nA\bar{H}(\bar{H}+1)}{4\varepsilon^2}\log(1/4\delta).$$

Now note that $n = 2^{d-1} = S/2$. Then, we only need to choose $\bar{H}$. Clearly, we need that $2\bar{H}+d \leq H$ to have all the segments above within the horizon, i.e., $H \geq 2\bar{H} + \log_2(S)$. If we choose $\bar{H} = H/3$, then $H \geq 3\log_2(s)$. With this choice we get

$$\mathbb{E}[\tau] \geq \frac{SAH^2}{72\varepsilon^2}\log(1/4\delta).$$

□

D Sample Complexity Bounds (Proofs of Section 4)

D.1 Good event

We consider the following concentration event

$$\mathcal{G} := \left\{\forall t \in \mathbb{N}, \forall h \in [H], s \in \mathcal{S}_h, a \in \mathcal{A}_h(s) : |\hat{r}_h^t(s,a) - r_h(s,a)| \leq b_h^t(s,a)\right\},$$

where we recall that the bonuses $b_h^t(s,a)$ are defined in (5).

Lemma 13. *The good event $\mathcal{G}$ holds with probability at least $1-\delta$.*

Proof. The proof easily follows from Hoeffding's inequality for sub-Gaussian distributions and a union bound:

$$\begin{aligned}
\mathbb{P}(\neg\mathcal{G}) &\leq \mathbb{P}\left(\exists t \in \mathbb{N}, h \in [H], s \in \mathcal{S}_h, a \in \mathcal{A}_h(s) : |\hat{r}_h^t(s,a) - r_h(s,a)| > b_h^t(s,a)\right) \\
&\leq \sum_{h\in[H]}\sum_{s\in\mathcal{S}_h}\sum_{a\in\mathcal{A}_h(s)}\mathbb{P}\left(\exists t \in \mathbb{N} : |\hat{r}_h^t(s,a) - r_h(s,a)| > b_h^t(s,a)\right) \\
&\leq \sum_{h\in[H]}\sum_{s\in\mathcal{S}_h}\sum_{a\in\mathcal{A}_h(s)}\mathbb{P}\left(\exists n \in \mathbb{N}^* : |\hat{r}_{h,n}(s,a) - r_h(s,a)| > \sqrt{\frac{2\sigma^2\log\left(\frac{4n^2N}{\delta}\right)}{n}}\right) \\
&\leq N\sum_{n=1}^{\infty}\frac{\delta}{2Nn^2} \leq \delta,
\end{aligned}$$

where $\hat{r}_{h,n}(s,a)$ denotes the mean reward estimate after n samples at (s,a,h). □

D.2 Properties of Algorithm 1

D.2.1 Correctness (Proof of Theorem 2)

For a given subset of policies $\Pi' \subseteq \Pi$, we define $\mathcal{S}_h(\Pi') := \{s \in \mathcal{S}_h \mid \exists \pi \in \Pi' : s_h^\pi = s\}$ as the set of states which are visited by some policy in Π' at stage h. Thus, $\mathcal{S}_h = \mathcal{S}_h(\Pi)$ is the set of all states reachable at stage h, while $\mathcal{S}_h(\Pi^\star)$ is the subset of states visited by optimal policies.

Lemma 14. *Under event $\mathcal{G}$, for any $t \in \mathbb{N}$, $h \in [H]$, $s \in \mathcal{S}_h(\Pi^\star)$, and $a \in \arg\max_{a \in \mathcal{A}_h(s)} Q_h^\star(s,a)$, we have $a \in \mathcal{A}_h^t(s)$, i.e., a is never eliminated.*

Proof. Take any $h \in [H], s \in \mathcal{S}_h(\Pi^\star), a \in \arg\max_{a \in \mathcal{A}_h(s)} Q_h^\star(s,a)$. Let us prove the result by induction. It clearly holds for $t = 0$ since the sets of active actions are initialized with the full sets of actions. Suppose it holds for $t-1$ with $t \geq 1$. Since this implies that $a \in \mathcal{A}_h^{t-1}(s)$, it is enough to show that

$$\max_{\pi \in \Pi_{s,a,h} \cap \Pi^{t-1}} \overline{V}_1^{t,\pi}(s_1) \geq \max_{\pi \in \Pi} \underline{V}_1^{t,\pi}(s_1)$$

to guarantee that $a \in \mathcal{A}_h^t(s)$. Then, for some optimal policy $\pi^\star \in \Pi^\star$,

$$\max_{\pi \in \Pi_{s,a,h} \cap \Pi^{t-1}} \overline{V}_1^{t,\pi}(s_1) \geq \overline{V}_1^{t,\pi^\star}(s_1) \geq V_1^{\pi^\star}(s_1) = \max_{\pi \in \Pi} V_1^\pi(s_1) \geq \max_{\pi \in \Pi} \underline{V}_1^{t,\pi}(s_1),$$

where the first inequality holds since there exists an optimal policy that visits (s,a) at stage h whose actions (at all visited states) are active by the inductive hypothesis, while the second and the third one are due to event $\mathcal{G}$. This concludes the proof. □

Lemma 15. *Under event $\mathcal{G}$, if the algorithm stops at the end of time $\tau \geq 1$ and returns a policy $\widehat{\pi}$, then $V_1^{\widehat{\pi}}(s_1) \geq V_1^\star(s_1) - \varepsilon$.*

Proof. We have two possible cases. First, suppose the algorithm stops with the first stopping rule. Under event $\mathcal{G}$, we know that some optimal policy $\pi^\star \in \Pi^\star$ is always active, hence $\pi^\star \in \Pi^\tau$. Then,

$$\begin{aligned}
V_1^\star(s_1) - V_1^{\widehat{\pi}}(s_1) = V_1^{\pi^\star}(s_1) - V_1^{\widehat{\pi}}(s_1) &\leq \overline{V}_1^{\pi^\star,\tau}(s_1) - \underline{V}_1^{\widehat{\pi},\tau}(s_1) \\
&\leq \max_{\pi \in \Pi^\tau} \overline{V}_1^{\pi,\tau}(s_1) - \underline{V}_1^{\widehat{\pi},\tau}(s_1) \\
&= \overline{V}_1^{\widehat{\pi},\tau}(s_1) - \underline{V}_1^{\widehat{\pi},\tau}(s_1) \\
&\leq \max_{\pi \in \Pi^\tau} \left(\overline{V}_1^{\pi,\tau}(s_1) - \underline{V}_1^{\pi,\tau}(s_1) \right) \leq \varepsilon,
\end{aligned}$$

where the first inequality holds by event $\mathcal{G}$, the second inequality holds since $\pi^\star \in \Pi^\tau$, the equality is by definition of the recommendation rule, and the last inequality is due to the stopping rule.

In the second case, if the algorithm stops with the second stopping rule, then Lemma 14 ensures that for all states visited by some optimal policy exactly the (necessarily unique) optimal action is left active. Therefore, the recommended policy plays only optimal actions in states that are visited by an optimal policy, which implies that the policy itself is optimal. □

Proof of Theorem 2. This is a simple combination of Lemma 15, which shows that the algorithm is ε-correct on event $\mathcal{G}$, and Lemma 13, which guarantees that $\mathcal{G}$ holds with probability at least $1 - \delta$. □

D.2.2 Diameter vs Gaps (Proof of Lemma 1)

We prove Lemma 1 stated in Section 4, an important result for Algorithm 1 which relates the diameter of active policies to the sub-optimality gaps of non-eliminated actions. Here we prove that the result holds under the good event $\mathcal{G}$, which in turns holds with probability at least $1 - \delta$.

Proof of Lemma 1. Suppose the good event $\mathcal{G}$ holds and let t be any episode at the end of which the algorithm did not stop. We derive separately a gap-dependent bound for sub-optimal state-action pairs and an ε-dependent bound for all state-action pairs.

Gap-dependent bound (suboptimal state-action pairs) Let $h \in [H], s \in \mathcal{S}_h, a \in \mathcal{A}_h(s)$ be such that $\overline{\Delta}_h(s,a) > 0$ (i.e., this is a sub-optimal state-action pair at stage h) and $a \in \mathcal{A}_h^t(s)$ (i.e., the action has not been eliminated at the end of episode t). Then, by definition of the elimination rule,

$$\max_{\pi \in \Pi_{s,a,h} \cap \Pi^{t-1}} \overline{V}_1^{t,\pi}(s_1) \geq \max_{\pi \in \Pi} \underline{V}_1^{t,\pi}(s_1).$$

Using the good event, this implies that, for any optimal policy $\pi^\star$,

$$\begin{aligned}\max_{\pi \in \Pi_{s,a,h} \cap \Pi^{t-1}} \left(V_1^\pi(s_1) + 2\sum_{h=1}^H b_h^t(s_h^\pi, a_h^\pi) \right) &\geq \max_{\pi \in \Pi} \left(V_1^\pi(s_1) - 2\sum_{h=1}^H b_h^t(s_h^\pi, a_h^\pi) \right) \\ &\geq V_1^\star(s_1) - 2\sum_{h=1}^H b_h^t(s_h^{\pi^\star}, a_h^{\pi^\star}).\end{aligned}$$

Thus,

$$2 \max_{\pi \in \Pi^{t-1}} \sum_{h=1}^H b_h^t(s_h^\pi, a_h^\pi) + 2\sum_{h=1}^H b_h^t(s_h^{\pi^\star}, a_h^{\pi^\star}) \geq V_1^\star(s_1) - \max_{\pi \in \Pi_{s,a,h}} V_1^\pi(s_1) = \overline{\Delta}_h(s,a).$$

Since all state-action pairs along each optimal trajectory are active under the good event (Lemma 14), $\sum_{h=1}^H b_h^t(s_h^{\pi^\star}, a_h^{\pi^\star}) \leq \max_{\pi \in \Pi^{t-1}} \sum_{h=1}^H b_h^t(s_h^\pi, a_h^\pi)$. Therefore, expanding the definition of the bonuses,

$$\overline{\Delta}_h(s,a) \leq 4 \max_{\pi \in \Pi^{t-1}} \sum_{h=1}^H b_h^t(s_h^\pi, a_h^\pi).$$

ε-dependent bound If the algorithm did not stop at the end of episode t, by the first stopping rule,

$$\frac{\varepsilon}{2} \leq \max_{\pi \in \Pi^t} \sum_{h=1}^H b_h^t(s_h^\pi, a_h^\pi) \leq \max_{\pi \in \Pi^{t-1}} \sum_{h=1}^H b_h^t(s_h^\pi, a_h^\pi).$$

Gap-dependent bound (unique optimal trajectory) Finally, let us consider the special case where the optimal trajectory $(s_h^\star, a_h^\star)_{h \in [H]}$ is unique. The derivation above holds for any state-action pair not belonging to an optimal trajectory (i.e., with positive gap). In this case, it can be trivially extended to optimal state-action pairs. Since the algorithm did not stop at the end of episode t, it must be that at least some sub-optimal state-action pair is active (otherwise there would be at most one active action in each state and the stopping condition would be verified). That is, there exist $h \in [H], s \in \mathcal{S}_h, a \in \mathcal{A}_h(s)$ with $\overline{\Delta}_h(s,a) \geq \overline{\Delta}_{\min} > 0$ such that $a \in \mathcal{A}_h^t(s)$. Using the same derivation as above, we obtain

$$\overline{\Delta}_{\min} \leq 4 \max_{\pi \in \Pi^{t-1}} \sum_{h=1}^H b_h^t(s_h^\pi, a_h^\pi).$$

□

D.3 Maximum-coverage algorithm (Proof of Theorem 3)

D.3.1 Static maximum-coverage sampling

For the purpose of the analysis, we introduce a variant of the maximum-coverage sampling rule, that we refer to as *static maximum-coverage*. As we will see, static maximum-coverage and (standard) maximum-coverage are very related, in the sense that in each period k there exists a function $C^k : 2^\Pi \to [0, \infty)$, called a coverage function, such that static-maximum coverage directly maximizes the function C^k, while maximum-coverage greedily builds a set of policies that maximizes it. The pseudo-code of static maximum-coverage is given in Algorithm 3.

In words, static maximum-coverage precomputes a set of policies of minimum size, which we call a *minimum policy cover*[6], that guarantees at least one visit to all active under-sampled (s,a,h),

[6] A similar concept of "policy cover" was considered by [3], where the authors propose an algorithm that incrementally builds a set of policies exploring the whole state-action space in the context of policy gradient methods.

Algorithm 3 Static maximum-coverage sampling

function STATICMAXCOVERAGE()
  Let $k_t \leftarrow \min_{h\in[H],s\in\mathcal{S}_h,a\in\mathcal{A}_h^{t-1}(s)} n_h^{t-1}(s,a)+1$ and $\bar{t}_{k_t} \leftarrow \inf_{l\in\mathbb{N}}\{l : k_l = k_t\}$
  if $k_t > k_{t-1}$ **then**
    Let $\underline{c}_h^{k_t}(s,a) \leftarrow \mathbb{1}\left(a \in \mathcal{A}_h^{\bar{t}_{k_t}-1}(s), n_h^{\bar{t}_{k_t}-1}(s,a) < k_t\right)$
    Compute η^{k_t}, an integer minimum flow on $\mathcal{G}(\mathcal{M})$ with lower bounds $\underline{c}^{k_t}$
    Extract a minimum policy cover Π_{cover}^t from η^{k_t} using Algorithm 2
  end if
  if $t \bmod 2 = 1$ **then**
    Choose π^t arbitrarily from Π_{cover}^t and remove it: $\Pi_{\text{cover}}^{t+1} \leftarrow \Pi_{\text{cover}}^t \setminus \{\pi^t\}$
    return π^t
  else
    Let $\Pi_{\text{cover}}^{t+1} \leftarrow \Pi_{\text{cover}}^t$
    return $\pi^t \leftarrow$ MAXDIAMETER()
  end if
end function

function MAXDIAMETER()
  return $\pi^t \leftarrow \arg\max_{\pi\in\Pi^{t-1}} \sum_{h=1}^H b_h^{t-1}(s_h^\pi, a_h^\pi)$
end function

i.e., all those such that $a \in \mathcal{A}_h^{\bar{t}_{k_t}-1}(s)$ and $n_h^{\bar{t}_{k_t}-1}(s,a) < k_t$. It is easy to see that (see also Appendix B.3) this problem can be reduced to finding a minimum flow with lower bound function $\underline{c}_h^{k_t}(s,a) := \mathbb{1}\left(a \in \mathcal{A}_h^{\bar{t}_{k_t}-1}(s), n_h^{\bar{t}_{k_t}-1}(s,a) < k_t\right)$. Stated otherwise, we require the resulting flow to have a value of at least 1 for every active (s,a,h) that has less than k visits. Once a minimum (integer) flow η^k has been computed, a policy cover can be easily extracted using Algorithm 2. Finally, once a minimum policy cover $\Pi_{\text{cover}}^{\bar{t}_k}$ for period k has been extracted, our sampling rule simply switches between playing a policy in this set to ensure good coverage of the whole MDP and playing the policy prescribed by the maximum-diameter sampling rule.

D.3.2 Main Theorem

We state the following theorem, which simultaneously upper bound the sample complexity of max-coverage sampling and its static version.

Theorem 8 (Formal statement of Theorem 3)**.** *With probability at least $1-\delta$, the sample complexity of Algorithm 1 combined with either the static maximum-coverage (in which case $C_H := 1$) or the maximum-coverage (in which case $C_H := \log(H)+1$) sampling rule is bounded by*

$$\tau \le 2C_H \left(\max_{(s,a,h)} \log\left(g_h(s,a)\right) + 1\right) \varphi^\star(g),$$

where $g : \mathcal{E} \to [0,\infty)$ is the lower bound function defined by

$$g_h(s,a) := \frac{32\sigma^2 H^2}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2}\left(\log\left(\frac{4N^3}{\delta}\right) + L_h(s,a)\right) + 2$$

with $L_h(s,a) := 8\log\left(\frac{16\sigma H \log\left(\frac{4N^3}{\delta}\right)}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)}\right)$. Moreover, with the same probability,

$$\tau \le \frac{256\sigma^2 SAH^2}{\varepsilon^2}\left(\log\left(\frac{4SAH}{\delta}\right) + 4\log\left(\frac{512\sigma^2 SAH^2 \log\left(\frac{4SAH}{\delta}\right)}{\varepsilon^2}\right)\right) + 8SAH + 2.$$

In the next sub-sections, we prove Theorem 8.

D.3.3 Decomposition into periods

Recall that k_t, defined in Algorithm 1 for both the static maximum-coverage and the maximum-coverage sampling rules, is the "target" number of visits at time t. We shall refer to the set of consecutive time steps $\{t \in \mathbb{N} : k_t = k\}$ as the k-th *period*. This is intuitively the set of time steps where the sampling rule is trying to make all active triplets reach k visits. Let $d_k := \sum_{t=1}^{\tau} \mathbb{1}(k_t = k)$ be the duration of the k-th period. Note that the period could be empty (e.g., this might happen when some under-sampled triplets are eliminated), in which case we have $d_k = 0$. The sample complexity can thus be decomposed as

$$\tau = \sum_{k=1}^{k_\tau} \sum_{t=1}^{\tau} \mathbb{1}(k_t = k) = \sum_{k=1}^{k_\tau} d_k.$$

Our goal in this section is to bound the duration of each period. In particular, while the duration of the k-th period can be trivially bounded by twice the size of the minimum policy cover $\Pi^{\bar{t}_k}_{\text{cover}}$ for the static maximum-coverage sampling rule, we shall see that a similar bound holds also for maximum-coverage.

Static maximum-coverage sampling The following bound can be easily derived from the definition of the sampling rule.

Theorem 9 (Period duration for static maximum-coverage)**.** *When using the static maximum-coverage sampling rule, for any non-empty period $k \in \mathbb{N}$,*

$$d_k \le 2\varphi^\star(\underline{c}^k).$$

Proof. If k is a non-empty period, there exists a time $\bar{t}_k$ at which the period starts where a minimum policy cover $\Pi^{\bar{t}_k}_{\text{cover}}$ is computed. The size of this cover is exactly the value $\varphi^\star(\underline{c}^k)$ of a minimum flow computed with lower bound function $\underline{c}^k_h(s,a) := \mathbb{1}\left(a \in \mathcal{A}_h^{\bar{t}_k - 1}(s), n_h^{\bar{t}_k - 1}(s,a) < k\right)$. The stated bound easily follows from the fact that a new policy in the cover is played every two episodes and that the period necessarily ends when all policies in the cover have been played. □

Maximum-coverage sampling Let $\bar{t}_k := \inf_{t \in \mathbb{N}}\{t : k_t = k\}$ be the first time step in the period as defined in Algorithm 1 (which exists if the period is non-empty). We start by proving the following result, which allows us to better characterize the duration of period k in terms of the first time where all active triplets at the *beginning* of the period receive at least k visits.

Lemma 16. *For any non-empty period $k \in \mathbb{N}$, almost surely*

$$d_k \le \tilde{d}_k := \inf_{t \in \mathbb{N}}\{t : \min_{h \in [H], s \in \mathcal{S}_h, a \in \mathcal{A}_h^{\bar{t}_k - 1}(s)} n_h^{t-1}(s,a) \ge k\} - \bar{t}_k < \infty.$$

Proof. Since period k ends when all active pairs are visited at least k times,

$$\begin{aligned} d_k = \inf_{t \in \mathbb{N}}\{t : k_t > k\} - \bar{t}_k &= \inf_{t \ge \bar{t}_k}\{t : \min_{h \in [H], s \in \mathcal{S}_h, a \in \mathcal{A}_h^{t-1}(s)} n_h^{t-1}(s,a) \ge k\} - \bar{t}_k. \\ &\le \inf_{t \in \mathbb{N}}\{t : \min_{h \in [H], s \in \mathcal{S}_h, a \in \mathcal{A}_h^{\bar{t}_k - 1}(s)} n_h^{t-1}(s,a) \ge k\} - \bar{t}_k. \end{aligned}$$

where the inequality holds since $\mathcal{A}_h^{t-1}(s) \subseteq \mathcal{A}_h^{\bar{t}_k - 1}(s)$ for each $h \in [H], s \in \mathcal{S}_h$, and $t \ge \bar{t}_k$. To see why $\tilde{d}_k < \infty$, note that, by definition, the sampling rule visits at least one undersampled triplet (s,a,h) (i.e., such that $n_h^{t-1}(s,a) < k$) every two steps. Since there are at most N triplets that need to be visited, we get that $\tilde{d}_k \le 2N < \infty$ almost surely. □

Reduction to submodular maximization Let us define the set function $C^k : 2^\Pi \to [0, \infty)$ as

$$C^k(\Pi') := \sum_{h=1}^{H} \sum_{s \in \mathcal{S}_h} \sum_{a \in \mathcal{A}_h^{\bar{t}_k - 1}(s)} \mathbb{1}\left(n_h^{\bar{t}_k - 1}(s,a) < k, \exists \pi \in \Pi' : (s_h^\pi, a_h^\pi) = (s,a)\right) \quad \forall \Pi' \subseteq \Pi.$$

Moreover, let $\bar{\Pi}_i^k := \{\pi^t \mid t = \bar{t}_k, \dots, \bar{t}_k + i - 1\}$ be the set containing the first i policies played by the maximum-coverage sampling rule in period k. We note that the first policy selection strategy (the one called at odd steps) is essentially a greedy algorithm approximating the maximization of C^k. In fact, maximizing C^k corresponds to finding a set of policies that visit all active triplets at time $\bar{t}_k - 1$ that have less than k visits (which, by definition of period, means that they have exactly $k-1$ visits). Instead of directly maximizing the set function C^k, such policy selection strategy greedily builds the set $\bar{\Pi}_i^k$ by adding, at each round where it is used, the policy visiting the most of these undervisited triplets. Moreover, we note that C^k is a *coverage function*, a kind of function which is known to be monotone submodular and for which greedy maximization is very efficient [36]. Let us prove some of its important properties.

First, we relate the maximization of C^k to the computation of a minimum flow with lower bound function $\underline{c}_h^k(s,a) \leftarrow \mathbb{1}\left(a \in \mathcal{A}_h^{\bar{t}_k-1}(s), n_h^{\bar{t}_k-1}(s,a) < k\right)$, i.e., the same as the one used by the static maximum-coverage sampling rule. Let $N_k := \sum_{h\in[H]} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h^{\bar{t}_k-1}(s)} \mathbb{1}\left(n_h^{\bar{t}_k-1}(s,a) < k\right)$ be the total number of triplets that need to be visited in period k.

Proposition 1 (Maximization vs minimum flow)**.** *For each $v \geq \varphi^\star(\underline{c}^k)$,*

$$\max_{\Pi'\subseteq\Pi:|\Pi'|\leq v} C^k(\Pi') = \max_{\Pi'\subseteq\Pi} C^k(\Pi') = N_k.$$

Proof. Clearly, $C^k(\Pi') \leq N_k$ for all $\Pi' \subseteq \Pi$, which is attained when all undervisited state-action-stage triplets are visited at least once. When the cardinality of Π' can be at least $\varphi^\star(\underline{c}^k)$, we can choose Π' to include a set of $\varphi^\star(\underline{c}^k)$ policies realizing a minimum 1-flow (i.e., a minimum policy cover as the one computed by static maximum-coverage sampling in period k). These, by definition, cover all undervisited triplets, and thus attain the maximal value N_k. □

Proposition 2 (Monotonicity)**.** *For each $\Pi' \subseteq \Pi'' \subseteq \Pi$, $C^k(\Pi') \leq C^k(\Pi'')$.*

Proof. This is trivial: since Π'' contains Π', it must visit at least all the triplets visited by Π'. □

Proposition 3 (Sub-modularity)**.** *Function f is sub-modular, i.e., for every $\Pi' \subseteq \Pi'' \subseteq \Pi$ and $\bar{\pi} \in \Pi \setminus \Pi''$,*

$$C^k(\Pi' \cup \{\bar{\pi}\}) - C^k(\Pi') \geq C^k(\Pi'' \cup \{\bar{\pi}\}) - C^k(\Pi'').$$

Proof. Note that

$$\begin{aligned}
&C^k(\Pi'\cup\{\bar{\pi}\}) - C^k(\Pi') \\
&\quad := \sum_{h=1}^{H} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h^{\bar{t}_k-1}(s), n_h^{\bar{t}_k-1}(s,a)<k} \mathbb{1}\left((s_h^{\bar{\pi}}, a_h^{\bar{\pi}}) = (s,a), \neg\exists \pi \in \Pi' : (s_h^{\pi}, a_h^{\pi}) = (s,a)\right) \\
&\quad = \sum_{h=1}^{H} \mathbb{1}\left(\neg\exists \pi \in \Pi' : (s_h^{\pi}, a_h^{\pi}) = (s_h^{\bar{\pi}}, a_h^{\bar{\pi}}), a_h^{\bar{\pi}} \in \mathcal{A}_h^{\bar{t}_k-1}(s_h^{\bar{\pi}}), n_h^{\bar{t}_k-1}(s_h^{\bar{\pi}}, a_h^{\bar{\pi}}) < k\right) \\
&\quad \geq \sum_{h=1}^{H} \mathbb{1}\left(\neg\exists \pi \in \Pi'' : (s_h^{\pi}, a_h^{\pi}) = (s_h^{\bar{\pi}}, a_h^{\bar{\pi}}), a_h^{\bar{\pi}} \in \mathcal{A}_h^{\bar{t}_k-1}(s_h^{\bar{\pi}}), n_h^{\bar{t}_k-1}(s_h^{\bar{\pi}}, a_h^{\bar{\pi}}) < k\right) \\
&\quad = C^k(\Pi'' \cup \{\bar{\pi}\}) - C^k(\Pi''),
\end{aligned}$$

where the inequality holds since $\Pi' \subseteq \Pi''$. □

Proposition 4 (Greedy maximization)**.** *Let $\bar{\Pi}_i^k$ be the set containing the first $i \geq 0$ policies computed by the maximum-coverage sampling rule in period k. Then, for any positive integer v,*

$$C^k(\bar{\Pi}_i^k) \geq (1 - e^{-\lfloor (i+1)/2 \rfloor / v}) \max_{\Pi'\subseteq\Pi:|\Pi'|\leq v} C^k(\Pi').$$

Proof. This is a simple extension to Theorem 1.5 of [31], which in turns is a slight generalization of a well-known result by [36]. We report the proof for completeness since we have to deal explicitly with time steps where the maximum-diameter rule (which is not a greedy maximizer of C^k) is used.

Fix some positive integers i, v. If i is such that $\bar{t}_k + i - 1$ is odd (i.e., the first sampling rule is used at step $\bar{t}_k + i - 1$), then using Equation 3 to 7 in the proof of Theorem 1.5 of [31],

$$C^\star := \max_{\Pi' \subseteq \Pi : |\Pi'| \leq v} C^k(\Pi') \leq C^k(\bar{\Pi}^k_{i-1}) + v(C^k(\bar{\Pi}^k_i) - C^k(\bar{\Pi}^k_{i-1})).$$

In particular, note that their inequality 6 holds since, by definition of our first sampling rule,

$$\pi^{\bar{t}_k + i - 1} \in \arg\max_{\pi \in \Pi} \left(C^k(\bar{\Pi}^k_{i-1} \cup \{\pi\}) - C^k(\bar{\Pi}^k_{i-1}) \right).$$

Rearranging, we get

$$C^\star - C^k(\bar{\Pi}^k_i) \leq (1 - 1/v)(C^\star - C^k(\bar{\Pi}^k_{i-1})).$$

On the other hand, if i is such that $\bar{t}_k + i - 1$ is even (i.e., the maximum-diameter sampling rule is used at step $\bar{t}_k + i - 1$), then, by monotonicity of C^k,

$$C^\star - C^k(\bar{\Pi}^k_i) \leq C^\star - C^k(\bar{\Pi}^k_{i-1}).$$

Therefore, unrolling this recursion from $i \geq 0$ and using that $C^k(\bar{\Pi}^k_0) = C^k(\emptyset) = 0$,

$$C^\star - C^k(\bar{\Pi}^k_i) \leq (1 - 1/v)^{\lfloor (i+1)/2 \rfloor} C^\star.$$

Using that $1 - x \leq e^{-x}$ and rearranging concludes the proof. □

Theorem 10 (Period duration for maximum-coverage)**.** *When using the maximum-coverage sampling rule, for any non-empty period $k \in \mathbb{N}$,*

$$d_k \leq 2\varphi^\star(\underline{c}^k)(\log(H) + 1).$$

Proof. Let $\underline{i} := \sup_{i \in \mathbb{N}} \{i : C^k(\bar{\Pi}^k_i) \leq N_k - \varphi^\star(\underline{c}^k)\}$ be the last iteration in period k at which at least $\varphi^\star(\underline{c}^k)$ triplets still need to be visited by the algorithm. Then, by Proposition 4 combined with Proposition 1,

$$N_k - \varphi^\star(\underline{c}^k) \geq C^k(\bar{\Pi}^k_{\underline{i}}) \geq (1 - e^{-\lfloor (\underline{i}+1)/2 \rfloor / \varphi^\star(\underline{c}^k)}) \max_{\Pi' \subseteq \Pi : |\Pi'| \leq \varphi^\star(\underline{c}^k)} C^k(\Pi') = (1 - e^{-\lfloor (\underline{i}+1)/2 \rfloor / \varphi^\star(\underline{c}^k)}) N_k.$$

Thus,

$$\lfloor (\underline{i} + 1)/2 \rfloor \leq \varphi^\star(\underline{c}^k) \log(N_k / \varphi^\star(\underline{c}^k)) \leq \varphi^\star(\underline{c}^k) \log(H),$$

where the second inequality holds since $\varphi^\star(\underline{c}^k) \geq \max_{h \in [H]} \sum_{s \in \mathcal{S}_h} \sum_{a \in \mathcal{A}_h^{\bar{t}_k - 1}(s)} \mathbb{1}\left(n_h^{\bar{t}_k - 1}(s, a) < k \right)$ by Lemma 6 and $N_k \leq H \max_{h \in [H]} \sum_{s \in \mathcal{S}_h} \sum_{a \in \mathcal{A}_h^{\bar{t}_k - 1}(s)} \mathbb{1}\left(n_h^{\bar{t}_k - 1}(s, a) < k \right)$. This implies that $\underline{i} \leq 2\varphi^\star(\underline{c}^k) \log(H) - 1$ if $\underline{i}$ is odd, while $\underline{i} \leq 2\varphi^\star(\underline{c}^k) \log(H)$ if $\underline{i}$ is even. Finally, note that $\tilde{d}_k \leq \underline{i} + 2\varphi^\star(\underline{c}^k)$ since at iteration $\underline{i} + 1$ less than $\varphi^\star(\underline{c}^k)$ triplets are missing and the algorithm visits at least a new one every two rounds. Then, the proof is concluded by Lemma 16. □

D.3.4 Elimination periods

We now bound the period indexes at which sub-optimal state-action pairs are eliminated. All results in this section hold for both the static maximum-coverage and the maximum-coverage sampling rule.

Lemma 17 (Cover property)**.** *For any non-empty period $k \in \mathbb{N}$, $h \in [H]$, $s \in \mathcal{S}_h$, and any action $a \in \mathcal{A}_h^{\bar{t}_k - 1}(s)$ that is active when the period begins,*

$$n_h^{\bar{t}_k - 1}(s, a) \geq k - 1.$$

Proof. This is trivial from the definition of period: $\bar{t}_k$ is the first time with $k_{\bar{t}_k} = k$ and $k_{\bar{t}_k} := \min_{h \in [H], s \in \mathcal{S}_h, a \in \mathcal{A}_h^{\bar{t}_k - 1}(s)} n_h^{\bar{t}_k - 1}(s, a) + 1$. □

Lemma 18 (Elimination periods)**.** *Recall that k_τ is the period in which Algorithm 1 stops and define*

$$\kappa_{s,a,h} := \inf_{k\in\mathbb{N}} \left\{ k : a \notin \mathcal{A}_h^{\bar{t}_{k+1}-1}(s) \right\} \wedge k_\tau$$

as the period at the end of which (s,a,h) is eliminated. Then, under the good event $\mathcal{G}$, for any $h \in [H], s \in \mathcal{S}_h, a \in \mathcal{A}_h(s)$,

$$\kappa_{s,a,h} \le \overline{\kappa}_{s,a,h} := \frac{32\sigma^2 H^2}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2} \left(\log\left(\frac{4N^3}{\delta}\right) + L_h(s,a) \right) + 1$$

where

$$L_h(s,a) := 8 \log\left(\frac{16\sigma H \log\left(\frac{4N^3}{\delta}\right)}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)} \right).$$

Proof. Take any $k \in \mathbb{N}, h \in [H], s \in \mathcal{S}_h, a \in \mathcal{A}_h(s)$ such that $a \in \mathcal{A}_h^{\bar{t}_{k+1}-1}(s)$. Under $\mathcal{G}$, we have,

$$\begin{aligned}
\max\left(\frac{\overline{\Delta}_h(s,a)}{4}, \frac{\overline{\Delta}_{\min}}{4}, \frac{\varepsilon}{2} \right) &\overset{(a)}{\le} \max_{\pi \in \Pi^{\bar{t}_{k+1}-2}} \sum_{h=1}^{H} b_h^{\bar{t}_{k+1}-1}(s_h^\pi, a_h^\pi) \\
&\overset{(b)}{\le} \max_{\pi \in \Pi^{\bar{t}_{k+1}-2}} \sum_{h=1}^{H} \sqrt{\frac{\beta(n_h^{\bar{t}_{k+1}-1}(s_h^\pi, a_h^\pi), \delta)}{n_h^{\bar{t}_{k+1}-1}(s_h^\pi, a_h^\pi)}} \\
&\overset{(c)}{\le} \max_{\pi \in \Pi^{\bar{t}_{k+1}-2}} \sum_{h=1}^{H} \sqrt{\frac{\beta(\bar{t}_{k+1}-1, \delta)}{n_h^{\bar{t}_{k+1}-1}(s_h^\pi, a_h^\pi)}} \\
&\overset{(d)}{\le} \max_{\pi \in \Pi^{\bar{t}_{k+1}-2}} \sum_{h=1}^{H} \sqrt{\frac{\beta(Nk, \delta)}{n_h^{\bar{t}_{k+1}-1}(s_h^\pi, a_h^\pi)}} \\
&\overset{(e)}{\le} H\sqrt{\frac{\beta(Nk,\delta)}{k}},
\end{aligned}$$

where (a) uses Lemma 1, (b) follows by expanding the definition of the bonuses, (c) uses that any active state-action-stage triplet cannot be visited more than t times and the end of round t, (d) uses that $\bar{t}_{k+1} - 1 \le \sum_{k'=1}^{k} d_{k'} \le Nk$ since a trivial bound on the duration of each period is N, and (e) uses Lemma 17. Note that

$$\beta(Nk,\delta) = 2\sigma^2 \log\left(\frac{4N^3k^2}{\delta}\right) = 2\sigma^2 \log\left(\frac{4N^3}{\delta}\right) + 4\sigma^2 \log(k)$$

Therefore, we have that, if (s,a,h) is active at the end of period k,

$$k \le 2\sigma^2 H^2 \frac{\log\left(\frac{4N^3}{\delta}\right) + 2\log(k)}{\max\left(\frac{\overline{\Delta}_h(s,a)}{4}, \frac{\overline{\Delta}_{\min}}{4}, \frac{\varepsilon}{2}\right)^2} \le 32\sigma^2 H^2 \frac{\log\left(\frac{4N^3}{\delta}\right) + 2\log(k)}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2}.$$

Using Lemma 19 with $C = \frac{32\sigma^2H^2\log\left(\frac{4N^3}{\delta}\right)}{\max\left(\overline{\Delta}_h(s,a),\overline{\Delta}_{\min},\varepsilon\right)^2}$ and $B = \frac{64\sigma^2H^2}{\max\left(\overline{\Delta}_h(s,a),\overline{\Delta}_{\min},\varepsilon\right)^2}$, while noting that

$$\log(B^2 + 2C) \le 2\log(4C) \le 4\log\left(\frac{16\sigma H \log\left(\frac{4N^3}{\delta}\right)}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)} \right) = \frac{L_h(s,a)}{2},$$

we obtain

$$k \le \frac{32\sigma^2 H^2}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2} \left(\log\left(\frac{4N^3}{\delta}\right) + L_h(s,a) \right).$$

The proof is concluded by noting that $\kappa_{s,a,h}$ is smaller than the first integer $k \in \mathbb{N}$ not satisfying the inequality above. □

D.3.5 Sample complexity

We prove first the instance-dependent bound and then the worst-case one.

Theorem 11. *Under the good event $\mathcal{G}$, the sample complexity of Algorithm 1 when combined with either the static maximum-coverage (in which case $C_H := 1$) or the maximum-coverage (in which case $C_H := \log(H) + 1$) sampling rule is bounded by*

$$\tau \leq 2C_H(\log(\overline{\kappa}) + 1)\varphi^\star(g),$$

where $g : \mathcal{E} \to [0, \infty)$ is defined as $g_h(s, a) = \overline{\kappa}_{s,a,h} + 1$, with $\overline{\kappa}_{s,a,h}$ being the upper bound on the elimination period of (s, a, h) from Lemma 18, and $\overline{\kappa} := \max_{h\in[H],s\in\mathcal{S}_h,a\in\mathcal{A}_h(s)} \overline{\kappa}_{s,a,h}$.

Proof. Using the decomposition into periods introduced in Section D.3.3 followed by Theorem 9 (for static maximum-coverage sampling) or Theorem 10 (for maximum-coverage sampling),

$$\tau = \sum_{k=1}^{k_\tau} \sum_{t=1}^{\tau} \mathbb{1}\left(k_t = k\right) = \sum_{k=1}^{k_\tau} d_k \leq 2C_H \sum_{k=1}^{k_\tau} \varphi^\star(\underline{c}^k),$$

where we recall that $\underline{c}_h^k(s, a) := \mathbb{1}\left(a \in \mathcal{A}_h^{\bar{t}_k-1}(s), n_h^{\bar{t}_k-1}(s, a) < k\right)$. Let $\mathbf{1}^k : \mathcal{E} \to [0, 1]$ be another lower bound function such that $\mathbf{1}_h^k(s, a) = \mathbb{1}\left(a \in \mathcal{A}_h^{\bar{t}_k-1}(s)\right)$. Then,

$$\sum_{k=1}^{k_\tau} \varphi^\star(\underline{c}^k) \leq \sum_{k=1}^{k_\tau} \varphi^\star(\mathbf{1}^k),$$

where the inequality is due to Lemma 5 and $\underline{c}_h^k(s, a) \leq \mathbf{1}_h^k(s, a)$ for all s, a, h. Let $k \geq 1$ and $\mathcal{C}^k$ be any maximum cut for the minimum flow problem with lower bounds $\mathbf{1}^k$. Then, by Theorem 6,

$$\varphi^\star(\mathbf{1}^k) = \psi(\mathcal{C}^k, \mathbf{1}^k) = \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C}^k)} \mathbf{1}_h^k(s, a) = \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C}^k)} \mathbb{1}\left(a \in \mathcal{A}_h^{\bar{t}_k-1}(s)\right),$$

where we recall that, since $\mathcal{C}^k$ is a maximum cut, it has no backward arc and thus its value is simply the sum of lower bounds on its forward arcs. Plugging this back into our sample complexity bound,

$$\begin{aligned}
\tau &\leq 2C_H \sum_{k=1}^{k_\tau} \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C}^k)} \mathbb{1}\left(a \in \mathcal{A}_h^{\bar{t}_k-1}(s)\right) \\
&= 2C_H \sum_{k=1}^{k_\tau} \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C}^k)} \mathbb{1}\left(k - 1 \leq \kappa_{s,a,h}\right) \\
&= 2C_H \sum_{k=1}^{k_\tau} \frac{1}{k} \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C}^k)} k\mathbb{1}\left(k - 1 \leq \kappa_{s,a,h}\right) \\
&\leq 2C_H \sum_{k=1}^{k_\tau} \frac{1}{k} \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C}^k)} (\kappa_{s,a,h} + 1) \\
&\leq 2C_H \sum_{k=1}^{k_\tau} \frac{1}{k} \max_{\mathcal{C}\in\mathfrak{C}} \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C})} (\kappa_{s,a,h} + 1) \\
&\leq 2C_H(\log(k_\tau) + 1) \max_{\mathcal{C}\in\mathfrak{C}} \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C})} (\kappa_{s,a,h} + 1) \\
&\leq 2C_H(\log(\overline{\kappa}) + 1) \max_{\mathcal{C}\in\mathfrak{C}} \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C})} (\overline{\kappa}_{s,a,h} + 1),
\end{aligned}$$

where in the last inequality we applied Lemma 18. Now note that the maximization in the last line computes a maximum cut for the problem with lower bound function $g : \mathcal{E} \to [0, \infty)$ defined by $g_h(s, a) = \overline{\kappa}_{s,a,h} + 1$. Therefore, the statement follows by applying Theorem 6. □

Theorem 12. *[Worst-case bound] Under the good event $\mathcal{G}$, the sample complexity of Algorithm 1 when combined with either the static maximum-coverage or the maximum-coverage sampling rule is bounded by*

$$\tau \leq \frac{256\sigma^2 SAH^2}{\varepsilon^2}\left(\log\left(\frac{4SAH}{\delta}\right) + 4\log\left(\frac{512\sigma^2 SAH^2 \log\left(\frac{4SAH}{\delta}\right)}{\varepsilon^2}\right)\right) + 8SAH + 2.$$

Proof. The proof is an easy extension of the one of Theorem 13 (part 2) where we only need to handle the fact that the maximum-diameter sampling rule is called once every two episodes. We report the full steps for completeness.

Take any time T at the end of which the algorithm did not stop. Let $\bar{t}$ be the first time step where all active triplets (s,a,h) have at least one visit. Note that $\bar{t} \leq N$ almost surely since max-coverage sampling visits at least one new triplet at each episode (see Appendix D.3.3) and the same holds for max-diameter. Then, for any $\bar{t} \leq t \leq T$ such that t is odd,

$$\frac{\varepsilon}{2} \leq \max_{\pi\in\Pi^t} \sum_{h=1}^{H} b_h^t(s_h^\pi, a_h^\pi) = \sum_{h=1}^{H} b_h^t(s_h^{\pi^{t+1}}, a_h^{\pi^{t+1}}) \leq \sum_{h=1}^{H} \sqrt{\frac{\beta(t,\delta)}{n_h^t(s_h^{\pi^{t+1}}, a_h^{\pi^{t+1}}) \vee 1}},$$

where the first inequality follows from the first stopping rule, the equality uses the fact that the second sampling rule is used at time $t+1$, and the last inequality uses the definition of the bonuses together with $n_h^t(s,a) \leq t$ and $n_h^t(s_h^{\pi^{t+1}}, a_h^{\pi^{t+1}}) \geq 1$ by definition of $\bar{t}$. Let $\bar{n}_h^t(s,a) := \sum_{\bar{t}\leq l\leq t:(l \bmod 2)=0} \mathbb{1}\left((s_h^l, a_h^l) = (s,a)\right) + 1$ be the number of visits to (s,a,h) restricted to even steps (i.e., those by the second sampling rule). Note that $n_h^t(s,a) \geq \bar{n}_h^t(s,a)$ for all $t \geq \bar{t}$. Then, we have the following sequence of inequalities (explained below):

$$\begin{aligned}
\frac{\varepsilon}{2}\lfloor (T-\bar{t}+1)/2 \rfloor &\overset{(a)}{\leq} \sum_{\bar{t}\leq t\leq T:(t \bmod 2)=1} \sum_{h=1}^{H} \sqrt{\frac{\beta(t,\delta)}{n_h^t(s_h^{\pi^{t+1}}, a_h^{\pi^{t+1}}) \vee 1}} \\
&\overset{(b)}{\leq} \sum_{\bar{t}\leq t\leq T:(t \bmod 2)=1} \sum_{h=1}^{H} \sqrt{\frac{\beta(t,\delta)}{\bar{n}_h^t(s_h^{\pi^{t+1}}, a_h^{\pi^{t+1}})}} \\
&\overset{(c)}{=} \sum_{h=1}^{H} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} \sum_{\bar{t}\leq t\leq T:(t \bmod 2)=1} \mathbb{1}\left((s_h^{\pi^{t+1}}, a_h^{\pi^{t+1}}) = (s,a)\right) \sqrt{\frac{\beta(t,\delta)}{\bar{n}_h^t(s,a)}} \\
&\overset{(d)}{\leq} \sqrt{\beta(T,\delta)} \sum_{h=1}^{H} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} \sum_{\bar{t}\leq t\leq T:(t \bmod 2)=1} \mathbb{1}\left((s_h^{\pi^{t+1}}, a_h^{\pi^{t+1}}) = (s,a)\right) \sqrt{\frac{1}{\bar{n}_h^t(s,a)}} \\
&\overset{(e)}{\leq} \sqrt{\beta(T,\delta)} \sum_{h=1}^{H} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} \sum_{i=1}^{\bar{n}_h^T(s,a)} \sqrt{\frac{1}{i}} \\
&\overset{(f)}{\leq} 2\sqrt{\beta(T,\delta)} \sum_{h=1}^{H} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} \sqrt{\bar{n}_h^T(s,a)} \\
&\overset{(g)}{\leq} 2\sqrt{\beta(T,\delta) N \sum_{h=1}^{H} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} \bar{n}_h^T(s,a)} \\
&\overset{(h)}{\leq} 2\sqrt{\beta(T,\delta) NH \lfloor T/2 \rfloor},
\end{aligned}$$

where (a) is by summing both sides of the inequality derived at the beginning over all odd t from $\bar{t}$ to T, (b) uses that $n_h^t(s,a) \geq \bar{n}_h^t(s,a)$ for all $s,a,h,t \geq \bar{t}$, (c) is trivial, (d) uses the monotonicity of $\beta(\cdot,\delta)$, (e) uses the standard pigeon-hole principle, (f) uses the inequality $\sum_{i=1}^{m} \sqrt{1/i} \leq 2\sqrt{m}$, (g) uses Cauchy-Schwartz inequality, and (h) uses that the total number of even episodes up to time T is $\lfloor T/2 \rfloor$. Therefore, we obtain the inequality

$$\frac{\varepsilon T}{4} \leq \sqrt{4\sigma^2 NHT\left(\log\left(\frac{4N}{\delta}\right) + 2\log(T)\right)} + \varepsilon N,$$

where we used that $\bar{t} - 1 \leq N$. Taking the square of both sides and using $(x+y)^2 \leq 2(x^2+y^2)$,

$$\frac{\varepsilon^2 T^2}{16} \leq 8\sigma^2 NHT \left(\log\left(\frac{4N}{\delta}\right) + 2\log(T) \right) + 2\varepsilon^2 N^2, \tag{6}$$

Up to constants, this is the same inequality we obtain in (8) for the proof of Theorem 13. By repeating exactly the same steps as for Theorem 13, we obtain

$$T \leq \frac{256\sigma^2 NH}{\varepsilon^2} \left(\log\left(\frac{4N}{\delta}\right) + 4\log\left(\frac{512\sigma^2 NH \log\left(\frac{4N}{\delta}\right)}{\varepsilon^2} \right) \right) + 8N.$$

The proof is concluded by noting that τ cannot be larger than the bound above plus two (since the maximum-diameter rule is called only every two steps) and that $N \leq SAH$. □

Proof of Theorem 3. The proof simply combines Theorem 11 and Theorem 12 together with the fact that the good event $\mathcal{G}$ holds with probability at least $1-\delta$ (Lemma 13). □

D.4 Maximum-diameter sampling

We now state the main Theorem which gives guarantees on the sample complexity of EPRL when it is coupled with Maximum Diameter sampling (Line 17 in Algorithm 1).

Theorem 13. *With probability at least $1-\delta$, the sample complexity of Algorithm 1 combined with the maximum-diameter sampling rule is bounded as*

$$\tau \leq \sum_{h\in[H]} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} \frac{32\sigma^2 H^2}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2} \left(\log\left(\frac{4N}{\delta}\right) + L \right) + N + 1,$$

where $L := 8\log\left(\frac{16N\sigma H \log\left(\frac{4N}{\delta}\right)}{\varepsilon} \right)$. *Moreover, with the same probability,*

$$\tau \leq \frac{128\sigma^2 SAH^2}{\varepsilon^2} \left(\log\left(\frac{4SAH}{\delta}\right) + 4\log\left(\frac{256\sigma^2 SAH^2 \log\left(\frac{4SAH}{\delta}\right)}{\varepsilon^2} \right) \right) + 2SAH + 1.$$

Proof. We first derive the instance-dependent bound and then focus on the worst-case one separately.

Part 1. Instance-dependent bound. We use the following "target trick" to obtain a sample complexity which scales as the sum of inverse gaps. Instead of bounding the number of times each state-action pair is visited, we imagine that each played policy "targets" some state-action pair and bound the number of times each state-action pair is targeted. Formally, we say that the policy π^t played at time t targets (s,a) at stage h if the following event occurs:

$$G^t_{s,a,h} := \left\{ h \in \arg\min_{l\in[H]} n_l^{t-1}(s_l^{\pi^t}, a_l^{\pi^t}), s_h^{\pi^t} = s, a_h^{\pi^t} = a \right\}.$$

Intuitively, we say that policy π^t targets the state-action pair (along its trajectory) that has been visited the least so far. Then, since at each time step at least one state-action-stage triplet is targeted,

$$\tau \leq \sum_{h=1}^{H} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} Z_h^\tau(s,a), \tag{7}$$

where $Z_h^\tau(s,a) := \sum_{t=1}^{\tau} \mathbb{1}\left(G^t_{s,a,h}\right)$ is the number of times (s,a,h) is targeted up to the stopping time. Thus, we shall focus on bounding $Z_h^t(s,a)$ for some fixed time t. Note that $Z_h^t(s,a) \leq n_h^t(s,a)$ since a targeted state-action-stage triplet is necessarily visited at time t. Moreover, $n_h^t(s,a)$ can be much larger than $Z_h^t(s,a)$ since (s,a,h) could be visited even without being the target.

Bounding $Z_h^t(s,a)$ Let t be any episode at which the algorithm did not stop. For any (s,a,h),

$$\max\left(\frac{\overline{\Delta}_h(s,a)}{4}, \frac{\overline{\Delta}_{\min}}{4}, \frac{\varepsilon}{2}\right) \overset{(a)}{\leq} \max_{\pi\in\Pi^{t-1}} \sum_{h=1}^{H} b_h^t(s_h^\pi, a_h^\pi) \overset{(b)}{\leq} \max_{\pi\in\Pi^{t-1}} \sum_{h=1}^{H} b_h^{t-1}(s_h^\pi, a_h^\pi)$$
$$\overset{(c)}{=} \sum_{h=1}^{H} b_h^{t-1}(s_h^{\pi^t}, a_h^{\pi^t}) \overset{(d)}{\leq} \sum_{h=1}^{H} \sqrt{\frac{\beta(t,\delta)}{n_h^{t-1}(s_h^{\pi^t}, a_h^{\pi^t})}},$$

where (a) is from Lemma 1, (b) from the monotonicity of the bonuses, (c) from the definition of the maximum-diameter sampling rule, and (d) from the definition of the bonuses. Now we distinguish two cases. If $G_{s,a,h}^t$ holds, then

$$\forall l \in [H] : n_l^{t-1}(s_l^{\pi^t}, a_l^{\pi^t}) \geq n_h^{t-1}(s,a) \geq Z_h^{t-1}(s,a).$$

Plugging this into the inequality above and rearranging, we obtain

$$\max\left(\frac{\overline{\Delta}_h(s,a)}{4}, \frac{\overline{\Delta}_{\min}}{4}, \frac{\varepsilon}{2}\right) \leq H\sqrt{\frac{\beta(t,\delta)}{Z_h^{t-1}(s,a)}} \implies Z_h^t(s,a) \leq \frac{16H^2\beta(t,\delta)}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2} + 1.$$

On the other hand, in case $G_{s,a,h}^t$ does not hold, then $Z_h^t(s,a) = Z_h^{t-1}(s,a)$ and we can recursively apply the reasoning above to obtain the same bound on $Z_h^t(s,a)$.

Bounding τ Evaluating this bound at $t = \tau - 1$, plugging it into (7), and expanding the definition of the threshold β, we obtain

$$\tau \leq \sum_{h\in[H]} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} \left(\frac{16H^2\beta(\tau-1,\delta)}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2} + 1\right) + 1$$
$$\leq \sum_{h\in[H]} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} \frac{32\sigma^2H^2}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2} \left(\log\left(\frac{4N}{\delta}\right) + 2\log(\tau)\right) + N + 1.$$

We conclude by applying Lemma 19 with $B = \sum_{h\in[H]} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} \frac{64\sigma^2H^2}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2}$ and $C = \sum_{h\in[H]} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} \frac{32\sigma^2H^2}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2} \log\left(\frac{4N}{\delta}\right) + N + 1$, while noting that

$$\log(B^2 + 2C) \leq \log\left(2\left(\frac{64N\sigma^2H^2\log\left(\frac{4N}{\delta}\right)}{\varepsilon^2}\right)^2 + 2N + 2\right)$$
$$\leq \log\left(4\left(\frac{64N\sigma^2H^2\log\left(\frac{4N}{\delta}\right)}{\varepsilon^2}\right)^2\right) \leq 4\log\left(\frac{16N\sigma H\log\left(\frac{4N}{\delta}\right)}{\varepsilon}\right).$$

Part 2. Worst-case bound. Take any time T at the end of which the algorithm did not stop. Let $\bar{t}$ be the first time step where all active triplets (s,a,h) have at least one visit. Note that $\bar{t} \leq N$ almost surely by definition of the sampling rule: since an active unvisited triplet has infinite confidence interval, the algorithm must visit at least a new one of such triplets in each episode. Then, for any $\bar{t} \leq t \leq T$,

$$\frac{\varepsilon}{2} \leq \max_{\pi\in\Pi^t} \sum_{h=1}^{H} b_h^t(s_h^\pi, a_h^\pi) = \sum_{h=1}^{H} b_h^t(s_h^{\pi^{t+1}}, a_h^{\pi^{t+1}}) \leq \sum_{h=1}^{H} \sqrt{\frac{\beta(t,\delta)}{n_h^t(s_h^{\pi^{t+1}}, a_h^{\pi^{t+1}}) \vee 1}},$$

where the first inequality follows from the first stopping rule, the equality uses the definition of the maximum-diameter sampling rule, and the last inequality uses the definition of the bonuses together

with $n_h^t(s,a) \le t$ and $n_h^t(s_h^{\pi^{t+1}}, a_h^{\pi^{t+1}}) \ge 1$ since $t \ge \bar{t}$. Then,

$$\begin{aligned}
\frac{\varepsilon}{2}(T-\bar{t}+1) &\overset{(a)}{\le} \sum_{t=\bar{t}}^{T}\sum_{h=1}^{H}\sqrt{\frac{\beta(t,\delta)}{n_h^t(s_h^{\pi^{t+1}}, a_h^{\pi^{t+1}}) \vee 1}} \\
&\overset{(b)}{=} \sum_{h=1}^{H}\sum_{s\in\mathcal{S}_h}\sum_{a\in\mathcal{A}_h(s)}\sum_{t=\bar{t}}^{T}\mathbb{1}\left((s_h^{\pi^{t+1}}, a_h^{\pi^{t+1}}) = (s,a)\right)\sqrt{\frac{\beta(t,\delta)}{n_h^t(s,a)\vee 1}} \\
&\overset{(c)}{\le} \sqrt{\beta(T,\delta)}\sum_{h=1}^{H}\sum_{s\in\mathcal{S}_h}\sum_{a\in\mathcal{A}_h(s)}\sum_{t=\bar{t}}^{T}\mathbb{1}\left((s_h^{\pi^{t+1}}, a_h^{\pi^{t+1}}) = (s,a)\right)\sqrt{\frac{1}{n_h^t(s,a)\vee 1}} \\
&\overset{(d)}{\le} \sqrt{\beta(T,\delta)}\sum_{h=1}^{H}\sum_{s\in\mathcal{S}_h}\sum_{a\in\mathcal{A}_h(s)}\sum_{i=2}^{n_h^T(s,a)}\sqrt{\frac{1}{i-1}} \\
&\overset{(e)}{\le} 2\sqrt{\beta(T,\delta)}\sum_{h=1}^{H}\sum_{s\in\mathcal{S}_h}\sum_{a\in\mathcal{A}_h(s)}\sqrt{n_h^T(s,a)-1} \\
&\overset{(f)}{\le} 2\sqrt{\beta(T,\delta)N\sum_{h=1}^{H}\sum_{s\in\mathcal{S}_h}\sum_{a\in\mathcal{A}_h(s)}n_h^T(s,a)} \\
&\overset{(g)}{=} 2\sqrt{\beta(T,\delta)NHT},
\end{aligned}$$

where (a) is by summing both sides of the inequality derived at the beginning over all t from 0 to T, (b) is trivial, (c) uses the monotonicity of $\beta(\cdot,\delta)$, (d) uses the standard pigeon-hole principle, (e) uses the inequality $\sum_{i=1}^{m}\sqrt{1/i} \le 2\sqrt{m}$, (f) uses Cauchy-Schwartz inequality, and (g) uses that the total number of samples after T episodes is TH. Therefore, we obtain the inequality,

$$\frac{\varepsilon T}{2} \le 2\sqrt{2\sigma^2 NHT\left(\log\left(\frac{4N}{\delta}\right) + 2\log(T)\right)} + \frac{\varepsilon}{2}N,$$

where we used $\bar{t} - 1 \le N$. Taking the square of both sides and using $(x+y)^2 \le 2(x^2+y^2)$,

$$\frac{\varepsilon^2 T^2}{4} \le 16\sigma^2 NHT\left(\log\left(\frac{4N}{\delta}\right) + 2\log(T)\right) + \frac{\varepsilon^2}{2}N^2 \tag{8}$$

This implies that the lhs is below twice the maximum between the two terms in the rhs. Suppose the first term in (8) is the maximum. Then,

$$\frac{\varepsilon^2 T^2}{4} \le 32\sigma^2 NHT\left(\log\left(\frac{4N}{\delta}\right) + 2\log(T)\right). \tag{9}$$

Using $\log(T) \le \sqrt{T}$, a crude bound on T is

$$\frac{\varepsilon^2 T^2}{4} \le 32\sigma^2 NHT\left(\log\left(\frac{4N}{\delta}\right) + 2\sqrt{T}\right) \le 64\sigma^2 NHT^{3/2}\log\left(\frac{4N}{\delta}\right),$$

which implies that

$$T \le \left(\frac{256\sigma^2 NH\log\left(\frac{4N}{\delta}\right)}{\varepsilon^2}\right)^2.$$

Plugging this into the log term in (9) and solving for T,

$$T \le \frac{128\sigma^2 NH}{\varepsilon^2}\left(\log\left(\frac{4N}{\delta}\right) + 4\log\left(\frac{256\sigma^2 NH\log\left(\frac{4N}{\delta}\right)}{\varepsilon^2}\right)\right).$$

Suppose now that the second term in (8) is the maximum. Then, we directly get $T \le 2N$.Then, T must be below the maximum of the two obtained bounds and hence below their sum,

$$T \le \frac{128\sigma^2 NH}{\varepsilon^2}\left(\log\left(\frac{4N}{\delta}\right) + 4\log\left(\frac{256\sigma^2 NH\log\left(\frac{4N}{\delta}\right)}{\varepsilon^2}\right)\right) + 2N.$$

This holds for any T at the end of which the algorithm did not stop. Therefore, τ cannot be larger than the bound above plus one. The proof is concluded by noting that $N \le SAH$. □

D.5 Auxiliary Results

Lemma 19. *Let $B, C \geq 1$. If $k \leq B\log(k) + C$, then*

$$k \leq B\log(B^2 + 2C) + C.$$

Proof. Since $\log(k) \leq \sqrt{k}$ for any $k \geq 1$, we have that $k \leq B\sqrt{k} + C$. Solving this second-order inequality, we get the crude bound $\sqrt{k} \leq \frac{B}{2} + \sqrt{\frac{B^2}{4} + C}$, which in turns yields $k \leq B^2 + 2C$ using that $(x+y)^2 \leq 2(x^2+y^2)$ for $x, y \geq 0$. The statement follows by plugging this bound into the logarithm. □

E Refined Results for Tree-based MDPs

In this appendix, we show that all our results can be refined for the specific class of deterministic MDPs represented by a tree, i.e., where each reachable state has exactly one incoming arc (except for the initial state which has none). This implies that there exists a unique path to reach each state $s \in \mathcal{S}_h$ at stage $h > 1$ from the root.

E.1 Instance-dependent lower bound

In the case of tree-based MDPs, one can derive a lower bound with an improved H factor. The intuition behind this result is the following: While in general MDPs the policies going through different triplets (s, a, h) and (s', a', h) may share some common state-action pairs at any further stage $l \geq h$, such phenomenon does not occur in tree-based MDPs. This makes the learning problem more difficult, as learning whether (s, a, h) is optimal or not does not gives us side-information about (s', a', h). Throughout this section, we will be using the same notation as Section 3.

Theorem 14. *Suppose that $\mathcal{M}$ is tree-based. Then:*

$$\mathbb{E}[\tau] \geq \max_h \sum_{s \in \mathcal{S}_h, a \in \mathcal{A}_h(s)} \frac{\sigma^2(H-h+1)\log(1/4\delta)}{4\max(\overline{\Delta}_h(s,a), \overline{\Delta}^h_{\min}, \varepsilon)^2},$$

where $\overline{\Delta}^h_{\min} := \min_{(s',a'):\overline{\Delta}_h(s',a')>0} \overline{\Delta}_h(s', a')$.

The proof of this theorem relies on the following lemma which is refined version of Lemmas 9, 10 and 11 in the case of tree-based MDPs. Before we state the lemma, we define for any triplet (s, a, h) the set

$$E(s,a,h) = \left\{(s', a', l) : \; l \in [|h, H|], \; s' \in \mathcal{S}_l, \; a' \in \mathcal{A}_l(s), \; \exists \pi \in \Pi_{s,a,h}, \; s^\pi_l = s', a^\pi_l = a'\right\}.$$

In words, $E(s, a, h)$ is the set of triplets at stages $l \geq h$ that are visited by the policies in the set $\Pi_{s,a,h}$.

Lemma 20. *Suppose that $\mathcal{M}$ is tree-based and fix any stage $h \in [H]$. We have:*

1. *For suboptimal pairs $(s, a) \notin \mathcal{Z}^\varepsilon_h$:*

$$\sum_{(s',a',l) \in E(s,a,h)} \mathbb{E}[n^\tau_l(s', a')] \geq \frac{2\sigma^2(H-h+1)^2}{(\overline{\Delta}_h(s,a) + \varepsilon)^2} \log(1/2.4\delta).$$

2. *For non-unique optimal pairs $(s, a) \in \mathcal{Z}^\varepsilon_h$ and $|\mathcal{Z}^\varepsilon_h| > 1$:*

$$\sum_{(s',a',l) \in E(s,a,h)} \mathbb{E}[n^\tau_l(s', a')] \geq \frac{\sigma^2(H-h+1)^2}{4\varepsilon^2} \log(1/4\delta).$$

3. *For unique optimal pairs $(s, a) \in \mathcal{Z}^\varepsilon_h$ and $|\mathcal{Z}^\varepsilon_h| = 1$:*

$$\sum_{(s',a',l) \in E(s,a,h)} \mathbb{E}[n^\tau_l(s', a')] \geq \frac{2\sigma^2(H-h+1)^2}{(\overline{\Delta}^h_{\min} + \varepsilon)^2} \log(1/4\delta),$$

where $\overline{\Delta}^h_{\min} := \min_{(s',a'):\overline{\Delta}_h(s',a')>0} \overline{\Delta}_h(s', a')$.

Proof. We distinguish four cases.

Case 1: $(s,a) \notin \mathcal{Z}_h^\varepsilon$**.** Consider the alternative MDP $\widetilde{\mathcal{M}} := (\mathcal{S}, \mathcal{A}, \{f_h, \widetilde{\nu}_h\}_{h\in[H]}, s_1, H)$ which is equivalent to $\mathcal{M}$ except that the reward is f only at the pairs $(s',a',l) \in E(s,a,h)$ as $\widetilde{\nu}_l(s',a') = \mathcal{N}(r_l(s',a') + \Delta, \sigma^2)$ with $\Delta > \frac{\overline{\Delta}_h(s,a)+\varepsilon}{H-h+1}$, while the reward distribution remains the same on all other state-action-stage triplets. Note that the values of policies in $\Pi \setminus \Pi_{s,a,h}$ remain unchanged. On the other hand, for all $\pi \in \arg\max_{\pi\in\Pi_{s,a,h}} V_1^\pi(s_1)$,

$$\widetilde{V}_1^\pi(s_1) = V_1^\pi(s_1) + (H-h+1)\Delta > V_1^\pi(s_1) + \overline{\Delta}_h(s,a) + \varepsilon = V_1^\star(s_1) + \varepsilon \geq \max_{\pi\notin\Pi_{s,a,h}} \widetilde{V}_1^\pi(s_1) + \varepsilon,$$

where the first equality is because we increased the mean reward by Δ at the pairs $(s_l^\pi, a_l^\pi)_{l\in[|h,H|]}$ and the second equality comes from the definition of $\overline{\Delta}_h(s,a)$ and the fact that $\pi \in \arg\max_{\pi\in\Pi_{s,a,h}} V_1^\pi(s_1)$. Now from the inequality above we deduce that $\mathbb{P}_{\widetilde{\mathcal{M}}}(\widehat{\pi} \in \Pi_{s,a,h}) \geq 1-\delta$. On the other hand, since $(s,a) \notin \mathcal{Z}_h^\varepsilon$, $\mathbb{P}_{\mathcal{M}}(\widehat{\pi} \in \Pi_{s,a,h}) \leq \delta$. Therefore Lemma 1 from [28] implies that:

$$\begin{aligned}\sum_{(s',a',l)\in E(s,a,h)} \mathbb{E}[n_l^\tau(s',a')] &\geq \frac{2}{\Delta^2}\mathrm{kl}(\mathbb{P}_{\mathcal{M}}(\widehat{\pi} \in \Pi_{s,a,h}), \mathbb{P}_{\widetilde{\mathcal{M}}}(\widehat{\pi} \in \Pi_{s,a,h})) \\ &\geq \frac{2\sigma^2}{\Delta^2}\mathrm{kl}(\delta, 1-\delta) \geq \frac{2\sigma^2}{\Delta^2}\log(1/2.4\delta).\end{aligned}$$

This holds for any $\Delta > \frac{\overline{\Delta}_h(s,a)+\varepsilon}{H-h+1}$ and the first statement is obtained by taking the limit.

Case 2: $(s,a) \in \mathcal{Z}_h^\varepsilon$, $|\mathcal{Z}_h^\varepsilon| > 1$ **and** $\mathbb{P}_{\mathcal{M}}(\widehat{\pi} \in \Pi_{s,a,h}) \leq 1/2$**.** We consider the same $\widetilde{\mathcal{M}}$ from the previous case. We still have that $\mathbb{P}_{\widetilde{\mathcal{M}}}(\widehat{\pi} \in \Pi_{s,a,h}) \geq 1-\delta$. Using Lemma 1 from [28] we get

$$\sum_{(s',a',l)\in E(s,a,h)} \mathbb{E}[n_l^\tau(s',a')] \geq \frac{2}{\Delta^2}\mathrm{kl}(\mathbb{P}_{\mathcal{M}}(\widehat{\pi} \in \Pi_{s,a,h}), \mathbb{P}_{\widetilde{\mathcal{M}}}(\widehat{\pi} \in \Pi_{s,a,h})) \geq \frac{2\sigma^2}{\Delta^2}\mathrm{kl}(1/2, 1-\delta).$$

By taking the limit $\Delta \to \frac{\overline{\Delta}_h(s,a)+\varepsilon}{H-h+1}$ we get:

$$\begin{aligned}\sum_{(s',a',l)\in E(s,a,h)} \mathbb{E}[n_l^\tau(s',a')] &\geq \frac{2\sigma^2}{(\overline{\Delta}_h(s,a)+\varepsilon)^2}\mathrm{kl}(1/2, 1-\delta) \\ &= \frac{2\sigma^2(H-h+1)^2}{\left(\overline{\Delta}_h(s,a)+\varepsilon\right)^2}\mathrm{kl}(1/2, \delta) \\ &\geq \frac{\sigma^2(H-h+1)^2}{4\varepsilon^2}\log(1/4\delta),\end{aligned}$$

where we used the fact that $\mathrm{kl}(x,y) = \mathrm{kl}(1-x, 1-y)$, $\mathrm{kl}(x,y) \geq x\log(1/y) - \log(2)$ and $\overline{\Delta}_h(s,a) \leq \varepsilon$.

Case 3: $(s,a) \in \mathcal{Z}_h^\varepsilon$, $|\mathcal{Z}_h^\varepsilon| > 1$ **and** $\mathbb{P}_{\mathcal{M}}(\widehat{\pi} \in \Pi_{s,a,h}) \geq 1/2$**.** Consider the alternative MDP $\widetilde{\mathcal{M}} := (\mathcal{S}, \mathcal{A}, \{f_h, \widetilde{\nu}_h\}_{h\in[H]}, s_1, H)$ which is equivalent to $\mathcal{M}$ except that the reward is modified only at the pairs $(s',a',l) \in E(s,a,h)$ as $\widetilde{\nu}_l(s',a') = \mathcal{N}(r_l(s',a') - \Delta, \sigma^2)$ with $\Delta > \frac{2\varepsilon-\overline{\Delta}_h(s,a)}{H-h+1}$, while the reward distribution remains the same on all other state-action-stage triplets. Note that the values of policies in $\Pi \setminus \Pi_{s,a,h}$ remain unchanged. On the other hand, for all $\pi \in \Pi_{s,a,h}$,

$$\begin{aligned}\widetilde{V}_1^\pi(s_1) &= V_1^\pi(s_1) - (H-h+1)\Delta \\ &< V_1^\pi(s_1) + \overline{\Delta}_h(s,a) - 2\varepsilon \\ &\leq V_1^\star(s_1) - 2\varepsilon \\ &\leq \max_{\pi\notin\Pi_{s,a,h}} V_1^\pi(s_1) - \varepsilon = \max_{\pi\notin\Pi_{s,a,h}} \widetilde{V}_1^\pi(s_1) - \varepsilon,\end{aligned}$$

where the first equality is because we decreased the mean reward by Δ at the pairs $(s_l^\pi, a_l^\pi)_{l\in[|h,H|]}$ and the last inequality is due to the fact that since $|\mathcal{Z}_h^\varepsilon| > 1$, there exists at least one ε-optimal policy

which does not visit (s,a) at step h (i.e., which belongs to $\Pi \setminus \Pi_{s,a,h}$). From the inequality above we deduce that $\mathbb{P}_{\widetilde{\mathcal{M}}}(\widehat{\pi} \in \Pi_{s,a,h}) \leq \delta$. Applying Lemma 1 from [28] to $\mathcal{M}$ and $\widetilde{\mathcal{M}}$ gives:

$$\begin{aligned}\sum_{(s',a',l)\in E(s,a,h)} \mathbb{E}[n_l^\tau(s',a')] &\geq \frac{2\sigma^2}{\Delta^2}\mathrm{kl}(\mathbb{P}_{\mathcal{M}}(\widehat{\pi} \in \Pi_{s,a,h}), \mathbb{P}_{\widetilde{\mathcal{M}}}(\widehat{\pi} \in \Pi_{s,a,h})) \\ &\geq \frac{2\sigma^2}{\Delta^2}\mathrm{kl}(1/2, \delta).\end{aligned}$$

By taking the limit $\Delta \to \frac{2\varepsilon - \overline{\Delta}_h(s,a)}{H-h+1}$ we get:

$$\begin{aligned}\sum_{(s',a',l)\in E(s,a,h)} \mathbb{E}[n_l^\tau(s',a')] &\geq \frac{2\sigma^2(H-h+1)^2}{\left(2\varepsilon - \overline{\Delta}_h(s,a)\right)^2}\mathrm{kl}(1/2, \delta) \\ &\geq \frac{\sigma^2(H-h+1)^2}{4\varepsilon^2}\log(1/4\delta),\end{aligned}$$

where we used the fact that $\mathrm{kl}(x,y) \geq x\log(1/y) - \log(2)$ and $\overline{\Delta}_h(s,a) \leq \varepsilon$. Cases 2 and 3 combined prove the second statement of the lemma.

Case 4: $(s,a) \in \mathcal{Z}_h^\varepsilon$, $|\mathcal{Z}_h^\varepsilon| = 1$**.** Consider the alternative MDP $\widetilde{\mathcal{M}} := (\mathcal{S}, \mathcal{A}, \{f_h, \widetilde{\nu}_h\}_{h\in[H]}, s_1, H)$ which is equivalent to $\mathcal{M}$ except that the reward is modified only at the pairs $(s',a',l) \in E(s,a,h)$ as $\widetilde{\nu}_l(s',a') = \mathcal{N}(r_l(s',a') - \Delta, \sigma^2)$ with $\Delta > \frac{\varepsilon + \overline{\Delta}_{\min}^h}{H-h+1}$, while the reward distribution remains the same on all other state-action-stage triplets. Note that the values of policies in $\Pi \setminus \Pi_{s,a,h}$ remain unchanged. On the other hand, for all $\pi \in \Pi_{s,a,h}$,

$$\begin{aligned}\widetilde{V}_1^\pi(s_1) &= V_1^\pi(s_1) - (H-h+1)\Delta \\ &< V_1^\pi(s_1) - \overline{\Delta}_{\min}^h - \varepsilon \\ &\leq V_1^\star(s_1) - \overline{\Delta}_{\min}^h - \varepsilon \\ &= \max_{\pi\notin\Pi_{s,a,h}} V_1^\pi(s_1) - \varepsilon = \max_{\pi\notin\Pi_{s,a,h}} \widetilde{V}_1^\pi(s_1) - \varepsilon,\end{aligned}$$

where in the last equality we used the fact that since (s,a) is the only ε-optimal pair, $\{(s',a') : \overline{\Delta}_h(s',a') = 0\} = \{(s,a)\}$ and therefore $\overline{\Delta}_{\min}^h = \min_{(s',a')\neq(s,a)} \overline{\Delta}_h(s',a') = V_1^\star(s_1) - \max_{\pi\notin\Pi_{s,a,h}} V_1^\pi(s_1)$. From the inequality above we deduce that $\mathbb{P}_{\widetilde{\mathcal{M}}}(\widehat{\pi} \in \Pi_{s,a,h}) \leq \delta$. On the other hand, since (s,a) is the only ε-optimal pair in $\mathcal{M}$, $\mathbb{P}_{\mathcal{M}}(\widehat{\pi} \in \Pi_{s,a,h}) \geq 1-\delta$. Using Lemma 1 from [28] to $\mathcal{M}$ and $\widetilde{\mathcal{M}}$ gives:

$$\begin{aligned}\sum_{(s',a',l)\in E(s,a,h)} \mathbb{E}[n_l^\tau(s',a')] &\geq \frac{2\sigma^2}{\Delta^2}\mathrm{kl}(\mathbb{P}_{\mathcal{M}}(\widehat{\pi} \in \Pi_{s,a,h}), \mathbb{P}_{\widetilde{\mathcal{M}}}(\widehat{\pi} \in \Pi_{s,a,h})) \\ &\geq \frac{2\sigma^2}{\Delta^2}\mathrm{kl}(1-\delta, \delta) \geq \frac{2\sigma^2}{\Delta^2}\log(1/2.4\delta).\end{aligned}$$

By taking the limit $\Delta \to \frac{\varepsilon + \overline{\Delta}_{\min}^h}{H-h+1}$ we get:

$$\sum_{(s',a',l)\in E(s,a,h)} \mathbb{E}[n_l^\tau(s',a')] \geq \frac{2\sigma^2(H-h+1)^2}{(\varepsilon + \overline{\Delta}_{\min}^h)^2}\log(1/2.4\delta).$$

This proves the last statement of the lemma. □

We are now ready to prove Theorem 14.

Proof of Theorem 14. Fix a stage $h \in [H]$. Since in a tree-based MDP the policies that visit different triplets at stage h do not cross paths later, then for any $(s,a,h) \neq (s',a',h)$: $E(s,a,h) \cap$

$E(s',a',h) = \emptyset$. Besides $\bigcup_{s\in\mathcal{S}_h, a\in\mathcal{A}_h(s)} E(s,a,h) \subset \{(s',a',l) :\ l \in [|h,H|],\ s' \in \mathcal{S}_l,\ a' \in \mathcal{A}_l(s)\}$. Therefore one can write:

$$\begin{aligned}\mathbb{E}[\tau] &= \frac{1}{H-h+1}\sum_{l=h}^{H}\sum_{s'\in\mathcal{S}_l, a'\in\mathcal{A}_l(s)} \mathbb{E}[n_l^\tau(s',a')] \\ &\geq \frac{1}{H-h+1}\sum_{s\in\mathcal{S}_h, a\in\mathcal{A}_h(s)}\ \sum_{(s',a',l)\in E(s,a,h)} \mathbb{E}[n_l^\tau(s',a')] \qquad (10)\end{aligned}$$

Combining inequality (10) with the bounds from Lemma 20 finishes the proof. □

E.2 Sample complexity of maximum-diameter sampling

Theorem 15. *With probability at least $1-\delta$, the sample complexity of Algorithm 1 combined with the maximum-diameter sampling rule (Line 17 of Algorithm 1) is bounded as*

$$\tau \leq \max_{h\in[H]} \sum_{s\in\mathcal{S}_h}\sum_{a\in\mathcal{A}_h(s)} \left(\frac{128\sigma^2H^2}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2}\left(\log\left(\frac{4N}{\delta}\right) + L\right) + 2\right),$$

where $L := 8\log\left(\frac{32\sigma NH\log\left(\frac{4N}{\delta}\right)}{\varepsilon}\right)$.

Proof. Suppose that event $\mathcal{G}$ holds and let t be any episode at which the algorithm did not stop. For any active (s,a,h), by Lemma 1 and the same decomposition as in the proof of Theorem 13,

$$\max\left(\frac{\overline{\Delta}_H(s,a)}{4}, \frac{\overline{\Delta}_{\min}}{4}, \frac{\varepsilon}{2}\right) \leq \sum_{h=1}^{H}\sqrt{\frac{\beta(t,\delta)}{n_h^{t-1}(s_h^{\pi^t}, a_h^{\pi^t})}} \leq H\sqrt{\frac{\beta(t,\delta)}{n_H^{t-1}(s_H^{\pi^t}, a_H^{\pi^t})}},$$

where the last inequality holds since in a tree-based MDP there exists a unique path to reach each leaf, which implies that $\forall h \in [H] : n_h^{t-1}(s_h^{\pi^t}, a_h^{\pi^t}) \geq n_H^{t-1}(s_H^{\pi^t}, a_H^{\pi^t})$. Summing this inequality over all episodes where (s,a) is visited at the final stage H starting from its second visit up to episode T,

$$\begin{aligned}\max\left(\frac{\overline{\Delta}_H(s,a)}{4}, \frac{\overline{\Delta}_{\min}}{4}, \frac{\varepsilon}{2}\right)(n_H^T(s,a)-1) &\leq H\sqrt{\beta(T,\delta)}\sum_{i=2}^{n_H^T(s,a)}\sqrt{\frac{1}{i-1}} \\ &\leq 2H\sqrt{\beta(T,\delta)(n_H^T(s,a)-1)}.\end{aligned}$$

Solving the inequality above,

$$n_H^T(s,a) \leq \frac{64H^2\beta(T,\delta)}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2} + 1.$$

Evaluating this bound at $T = \tau - 1$,

$$\begin{aligned}\tau = \sum_{s\in\mathcal{S}_H}\sum_{a\in\mathcal{A}_H(s)} n_H^\tau(s,a) &\leq \sum_{s\in\mathcal{S}_H}\sum_{a\in\mathcal{A}_H(s)}\left(\frac{64H^2\beta(\tau,\delta)}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2} + 2\right) \\ &= \sum_{s\in\mathcal{S}_H}\sum_{a\in\mathcal{A}_H(s)}\left(\frac{128\sigma^2H^2}{\max\left(\overline{\Delta}_H(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2}\left(\log\left(\frac{4N}{\delta}\right) + 2\log(\tau)\right) + 2\right).\end{aligned}$$

The proof is concluded by applying Lemma 19 with $B = \sum_{s\in\mathcal{S}_H}\sum_{a\in\mathcal{A}_H(s)} \frac{256\sigma^2H^2}{\max\left(\overline{\Delta}_H(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2}$ and $C = \sum_{s\in\mathcal{S}_H}\sum_{a\in\mathcal{A}_H(s)}\left(\frac{128\sigma^2H^2}{\max\left(\overline{\Delta}_H(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2}\log\left(\frac{4N}{\delta}\right) + 2\right)$, while noting that

$$\log(B^2 + 2C) \leq \log\left(2\left(\frac{256\sigma^2NH^2\log\left(\frac{4N}{\delta}\right)}{\varepsilon^2}\right)^2 + 2N\right) \leq 4\log\left(\frac{32\sigma NH\log\left(\frac{4N}{\delta}\right)}{\varepsilon}\right).$$

□

E.3 Sample complexity of maximum-coverage sampling

Theorem 16. *With probability at least $1-\delta$, the sample complexity of Algorithm 1 combined with either the maximum-coverage or the static maximum-coverage (Algorithm 3) sampling rule is bounded by*

$$\tau \leq 2 \max_{h\in[H]} \sum_{s\in\mathcal{S}_h} \sum_{a\in\mathcal{A}_h(s)} \left(\frac{32\sigma^2 H^2}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)^2} \left(\log\left(\frac{4N^3}{\delta}\right) + L_h(s,a) \right) + 2 \right),$$

where $L_h(s,a) := 8 \log\left(\frac{16\sigma H \log\left(\frac{4N^3}{\delta}\right)}{\max\left(\overline{\Delta}_h(s,a), \overline{\Delta}_{\min}, \varepsilon\right)} \right)$.

Before proving Theorem 16, we need to state an important result.

Lemma 21. *In a tree-based MDP, when using either the maximum-coverage or the static maximum-coverage sampling rule, the duration of any non-empty period $k \in \mathbb{N}$ can be bounded as*

$$d_k \leq 2 \sum_{s\in\mathcal{S}_H} \sum_{a\in\mathcal{A}_H(s)} \mathbb{1}\left(a \in \mathcal{A}_h^{\bar{t}_k - 1}(s)\right).$$

Proof. For static maximum-coverage sampling, Theorem 9 followed by Theorem 6 yields that

$$d_k \leq 2\varphi^\star(\underline{c}^k) = 2 \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C}^k)} \mathbb{1}\left(a \in \mathcal{A}_h^{\bar{t}_k-1}(s), n_h^{\bar{t}_k-1}(s,a) < k\right) \leq 2 \sum_{(s,a,h)\in\mathcal{E}(\mathcal{C}^k)} \mathbb{1}\left(a \in \mathcal{A}_h^{\bar{t}_k-1}(s)\right),$$

where $\mathcal{C}^k$ is a maximum cut for the minimum flow problem with lower bounds $\underline{c}^k$. Now note that any $(s,a,h) \in \mathcal{E}(\mathcal{C}^k)$ such that $a \in \mathcal{A}_h^{\bar{t}_k-1}(s)$ reaches a *distinct* leaf (s',a',H) such that $a' \in \mathcal{A}_H^{\bar{t}_k-1}(s')$. To see why, note that, if some active $(s,a,h) \in \mathcal{E}(\mathcal{C}^k)$ reaches no active triplet at the last stage H, then we can recursively prove that the sub-tree with root (s,a,h) has been eliminated, which implies that (s,a,h) has been eliminated as well. To see why these triplets are distinct, suppose that there exist two triplets $(s,a,h),(s',a',h') \in \mathcal{E}(\mathcal{C}^k)$ that reach the same leaf (s'',a'',H). Since, by definition of forward arcs of a cut, (s,a,h) and (s',a',h') cannot be on the same path, this implies that there exist two different paths to reach the same leaf from the root, which violates the tree-based assumption. This allows us to conclude that $\sum_{(s,a,h)\in\mathcal{E}(\mathcal{C}^k)} \mathbb{1}\left(a \in \mathcal{A}_h^{\bar{t}_k-1}(s)\right) \leq \sum_{s\in\mathcal{S}_H} \sum_{a\in\mathcal{A}_H(s)} \mathbb{1}\left(a \in \mathcal{A}_h^{\bar{t}_k-1}(s)\right)$, and the proof follows.

The reasoning for maximum-coverage sampling is similar. First note that, at each step of period k, the sampling rule must play a policy visiting a distinct leaf than those previously visited in the same period. In fact, since there is a unique path to reach each leaf, if the same leaf is visited twice, then at the second visit the value of the objective function would be zero, which cannot happen unless the period has already terminated. Moreover, once all leaves have been visited, by the reasoning above, we are sure that the algorithm has covered a maximum cut for the lower bound function $\underline{c}^k$. That is, all under-sampled triplets have been visited and the period terminates. This proves the stated bound. □

Proof of Theorem 16. Using Lemma 21 and following the same steps as in the proof of Theorem 3,

$$\begin{aligned}
\tau = \sum_{k=1}^{k_\tau} \sum_{t=1}^{\tau} \mathbb{1}\left(k_t = k\right) = \sum_{k=1}^{k_\tau} d_k &\leq 2 \sum_{k=1}^{k_\tau} \sum_{s\in\mathcal{S}_H} \sum_{a\in\mathcal{A}_H(s)} \mathbb{1}\left(a \in \mathcal{A}_h^{\bar{t}_k-1}(s)\right) \\
&= 2 \sum_{s\in\mathcal{S}_H} \sum_{a\in\mathcal{A}_H(s)} \sum_{k=1}^{k_\tau} \mathbb{1}\left(k-1 \leq \kappa_{s,a,h}\right) \\
&\leq 2 \sum_{s\in\mathcal{S}_H} \sum_{a\in\mathcal{A}_H(s)} (\kappa_{s,a,h}+1) \leq 2 \sum_{s\in\mathcal{S}_H} \sum_{a\in\mathcal{A}_H(s)} (\overline{\kappa}_{s,a,h}+1),
\end{aligned}$$

where in the last inequality we applied Lemma 18. □

F Experiment Details

For the implementation, we used **rl-berry** [16], an open-source python library for implementing and performing parallel Monte-Carlo simulations of RL algorithms. The code and instructions can be found in the supplementary material.

Computational aspects We run the experiment on an internal cluster made of 32 CPUs. To speed-up computations, we only perform eliminations every SA episodes for maximum-diameter and at the end of each phase for maximum-coverage. The total run time is 48 hours.

On the choice of baselines The only algorithms for PAC RL in Episodic MDPs that we are aware of are BPI-UCRL [29], BPI-UCBVI [35] and MOCA [46]. However, we note that BPI-UCRL and BPI-UCBVI only differ in the type of bonus that they use to build confidence regions on the transition probabilities. This means that in our setting of deterministic MDPs, both algorithms are actually equivalent. On the other hand, MOCA has a rather involved design with several unspecified numerical constants and we could not find any open-source implementation of it by the authors. This is why only BPI-UCRL appears in our comparisons.

BPI-UCRL Whereas EPRL uses confidence intervals on the value of every policy, BPI-UCRL [29] is based on confidence intervals for the optimal value function. Such confidence intervals were originally proposed for stochastic MDPs with known reward function, which require a confidence region for the unknown transition probabilities. In deterministic MDPs, one can easily build confidence intervals on the optimal values by relying on confidence intervals for the unknown mean rewards:

$$\overline{Q}_h^{t,\star}(s,a) := \hat{r}_h^t(s,a) + b_h^t(s,a) + \overline{V}_{h+1}^{t,\star}(f_h(s,a)), \quad \overline{V}_h^{t,\star}(s) := \max_b \overline{Q}_h^{t,\star}(s,b),$$

$$\underline{Q}_h^{t,\star}(s,a) := \hat{r}_h^t(s,a) - b_h^t(s,a) + \underline{V}_{h+1}^{t,\star}(f_h(s,a)), \quad \underline{V}_h^{t,\star}(s) := \max_{b'} \underline{Q}_h^{t,\star}(s,b') \ .$$

using the same exploration bonus as in (5). In BPI-UCRL, the (optimistic) samplig rule is

$$\pi_h^t(s) = \arg\max_{a \in \mathcal{A}_h(s)} \overline{Q}_h^{t-1,\star}(s,a).$$

The stopping rule is

$$\tau^{\text{BPI-UCRL}} = \inf\left\{ t \in \mathbb{N} : \overline{V}_1^{t,\star}(s_1) - \underline{V}_1^{t,\star}(s_1) \le \varepsilon \right\},$$

while the recommendation rule is the greedy policy with respect to $\underline{Q}_h^{t,\star}(s,a)$.

Value-based eliminations In our implementation, we used an additional elimination rule for both EPRL and BPI-UCRL, which we call value-based elimination: a is eliminated from $\mathcal{A}_h^t(s)$ if

$$\overline{Q}_h^{t-1,\star}(s,a) < \underline{V}_h^{t-1,\star}(s).$$

It is easy to justify that on our good event, this sampling rule does not eliminate any optimal action, hence the correctness is preserved. Moreover, adding these eliminations does not alter the sample complexity results obtained in Theorems 3 and 13 as they can only improve the sample complexity of the resulting algorithms.

Bonuses in practice The threshold $\beta(t,\delta) := 2\sigma^2 \log\left(4t^2 N/\delta\right)$ recommended by theory can be overly conservative. In practice, we found that a smaller threshold of $\beta(t,\delta) := 2\sigma^2 \log\left((t+1)/\delta\right)$ (i.e., ignoring the union bound) is still empirically correct, and used it in our experiments.