
Sample-Efficient Reinforcement Learning with $\log\log(T)$ Switching Cost

Dan Qiao¹ Ming Yin^{1,2} Ming Min² Yu-Xiang Wang¹

Abstract

We study the problem of reinforcement learning (RL) with low (policy) switching cost — a problem well-motivated by real-life RL applications in which deployments of new policies are costly and the number of policy updates must be low. In this paper, we propose a new algorithm based on stage-wise exploration and adaptive policy elimination that achieves a regret of $\tilde{O}(\sqrt{H^4 S^2 A T})$ while requiring a switching cost of $O(HSA \log \log T)$. This is an exponential improvement over the best-known switching cost $O(H^2 S A \log T)$ among existing methods with $\tilde{O}(\text{poly}(H, S, A)\sqrt{T})$ regret. In the above, S, A denotes the number of states and actions in an H -horizon episodic Markov Decision Process model with unknown transitions, and T is the number of steps. As a byproduct of our new techniques, we also derive a reward-free exploration algorithm with a switching cost of $O(HSA)$. Furthermore, we prove a pair of information-theoretical lower bounds which say that (1) Any no-regret algorithm must have a switching cost of $\Omega(HSA)$; (2) Any $\tilde{O}(\sqrt{T})$ regret algorithm must incur a switching cost of $\Omega(HSA \log \log T)$. Both our algorithms are thus optimal in their switching costs.

1. Introduction

In many real-world reinforcement learning (RL) tasks, it is costly to run fully adaptive algorithms that update the exploration policy frequently. Instead, collecting data in large batches using the current policy deployment is usually cheaper. For instance, in recommendation systems (Afsar et al., 2021), the system is able to collect millions of new data points in minutes, while the deployment of a new policy

often takes weeks, as it involves significant extra cost and human effort. It is thus infeasible to change the policy after collecting every new data point as a typical RL algorithm would demand. A practical alternative is to schedule a large batch of experiments in parallel and only decide whether to change the policy after the whole batch is complete. Similar constraints arise in other RL applications such as those in healthcare (Yu et al., 2021), database optimization (Krishnan et al., 2018), computer networking (Xu et al., 2018) and new material design (Raccuglia et al., 2016).

In those scenarios, the agent needs to minimize the number of policy switching while maintaining (nearly) the same regret bounds as its fully-adaptive counterparts. On the empirical side, Matsushima et al. (2020) cast this problem via the notion *deployment efficiency* and designed algorithms with high deployment efficiency for both online and offline tasks. On the theoretical side, Bai et al. (2019) first brought up the definition of *switching cost* that measures the number of policy updates. They designed Q learning-based algorithm with regret of $\tilde{O}^*(\sqrt{T})$ and switching cost of $O^*(\log T)$ ¹. Later, Zhang et al. (2020c) improved both the regret bound and switching cost bound. However, the switching cost remains order $O^*(\log T)$. In addition, both algorithms need to monitor the data stream to decide whether the policy should be switched at each episode. In contrast, for an A -armed bandit problem, Cesa-Bianchi et al. (2013) created *arm elimination* algorithm that achieves the optimal $\tilde{O}(\sqrt{AT})$ regret and a near constant switching cost bound of $O(A \log \log T)$. Meanwhile, the arm elimination algorithm predefined when to change policy before the algorithm starts, which could render parallel implementation. To adapt this feature from multi-armed bandit to RL problem, one straightforward way is to consider each deterministic policy (A^{SH} policies in total) as an arm. Applying the same algorithm for the RL setting, one ends up with the switching cost to be $O(A^{SH} \log \log T)$ and the regret bound of order $O(\sqrt{A^{SH} T})$. Clearly, such an adaptation is far from satisfactory as the exponential dependence on H, S makes the algorithm inefficient. This motivates us to consider the following question:

Question 1.1. Is it possible to design an algorithm for online RL problem with $O^*(\log \log T)$ switching cost and

¹Department of Computer Science, UC Santa Barbara
²Department of Statistics and Applied Probability, UC Santa Barbara. Correspondence to: Dan Qiao <danqiao@ucsb.edu>, Yu-Xiang Wang <yuxiangw@cs.ucsb.edu>.

¹Here $O^*(\cdot)$ and \tilde{O}^* omit a $\text{poly}(H, S, A)$ terms, this will be a notation we use throughout.

<i>Algorithms for regret minimization</i>	<i>Regret</i>	<i>Switching cost</i>
UCB2-Bernstein (Bai et al., 2019)	$\tilde{O}(\sqrt{H^3 SAT})$	Local: $O(H^3 SA \log T)$
UCB-Advantage (Zhang et al., 2020c)	$\tilde{O}(\sqrt{H^2 SAT})$	Local: $O(H^2 SA \log T)$
Algorithm 1 in (Gao et al., 2021) *	$\tilde{O}(\sqrt{d^3 H^3 T})$	Global: $O(dH \log T)$
APEVE (Our Algorithm 1)	$\tilde{O}(\sqrt{H^4 S^2 AT})$	Global: $O(HSA \log \log T)$
Explore-First w. LARFE (Our Algorithm 4)	$\tilde{O}(T^{2/3} H^{4/3} S^{2/3} A^{1/3})$	Global: $O(HSA)$
Lower bound (Our Theorem 4.2)	if $\tilde{O}(\sqrt{T})$ (“Optimal regret”)	Global: $\Omega(HSA \log \log T)$
Lower bound (Our Theorem 4.3)	if $o(T)$ (“No regret”)	Global: $\Omega(HSA)$
<i>Algorithms for reward-free exploration</i>	<i>Sample (episode) complexity</i>	<i>Switching cost</i>
Algorithm 2&3 in (Jin et al., 2020a)	$\tilde{O}(\frac{H^5 S^2 A}{\epsilon^2})$	Global: $\tilde{O}(\frac{H^7 S^4 A}{\epsilon})^\ddagger$
RF-UCRL (Kaufmann et al., 2021)	$\tilde{O}(\frac{H^4 S^2 A}{\epsilon^2})$	Global: $\tilde{O}(\frac{H^4 S^2 A}{\epsilon^2})$
RF-Express (Ménard et al., 2021)	$\tilde{O}(\frac{H^3 S^2 A}{\epsilon^2})$	Global: $\tilde{O}(\frac{H^3 S^2 A}{\epsilon^2})$
SSTP (Zhang et al., 2020b)	$\tilde{O}(\frac{S^2 A}{\epsilon^2})^*$	Global: $\tilde{O}(SA \log(\frac{S^2 A}{\epsilon^2}))^\ddagger$
Algorithm 3&4 in (Huang et al., 2022)	$\tilde{O}(dH(\frac{d^{3c_K} H^{6c_K+1}}{\epsilon^{2c_K}})^{\frac{1}{c_K-1}})$	Global: $c_K dH + 1$
LARFE (Our Algorithm 4)	$\tilde{O}(\frac{H^5 S^2 A}{\epsilon^2})$	Global: $O(HSA)$

Table 1. Comparison of our results (in blue) to existing work regarding problem type, regret/sample complexity, and switching cost. Note that some of the works are under linear MDP, where d is the dimension of feature map. When the feature map is the canonical basis (Jin et al., 2020b), linear MDP recovers tabular MDP and $d = SA$. *: This result is generalized by Wang et al. (2021), whose algorithm has a same switching cost bound under this regret bound. †: In (Jin et al., 2020a), there are $\tilde{O}(\frac{H^7 S^4 A}{\epsilon})$ episodes of data collected using EULER, which can lead to the same number of switching cost in the worst case. *: This result is derived under stationary MDP with total reward bounded by 1. ‡: We translate the use of trigger set in Algorithm 3 (Zhang et al., 2020b) to a worst case switching cost bound.

$\tilde{O}(\text{poly}(H, S, A)\sqrt{T})$ regret bound while it can decide when to change policy before the process starts?

Our contributions. In this paper, we answer the above question affirmatively by contributing the new low switching algorithm APEVE (Algorithm 1). Furthermore, the framework of APEVE naturally adapts to the more challenging *low switching reward-free* setting and we end up with LARFE (Algorithm 4) as a byproduct. Our concrete contributions are summarized as follows. To the best of our knowledge, all of the results are the first of its kinds.

- A new policy elimination algorithm APEVE (Algorithm 1) that achieves $O(HSA \log \log T)$ switching costs (Theorem 4.1). This provides an exponential improvement over the existing algorithms that require an $O^*(\log T)$ switching cost to achieve $\tilde{O}^*(\sqrt{T})$ regret.²
- A matching global switching cost lower bound of $\Omega(HSA \log \log T)$ for any algorithm with $\tilde{O}^*(\sqrt{T})$ regret bound (Theorem 4.2). This certifies the policy switching of APEVE is near-optimal for sample-efficient RL. As a byproduct, we provide a global

switching cost lower bound of $\Omega(HSA)$ for any no-regret algorithm (Theorem 4.3).

- We also propose a new low-adaptive algorithm LARFE for *reward-free* exploration (Algorithm 4). It comes with an optimal global switching cost of $O(HSA)$ for deterministic policies (Theorem 5.1) and allows the identification of an ϵ -optimal policy *simultaneously* for all (unknown, possibly data-dependent) reward design.

Why $\log \log T$ switching cost matters? The improvement from $\log T$ to $\log \log T$ could make a big difference in practical applications. Take $T = 1e5$ as an example, $\log T \approx 11.5$ and $\log \log T \approx 2.4$. This represents a nearly 5x improvement in a reasonably-sized exploration dataset one can collect. 80% savings in the required resources could distinguish between what is practical and what is not, and will certainly allow for more iterations. On the other hand, the total number of atoms in the observable universe $\approx 10^{82}$ and $\log \log 10^{82} \approx 5.24$. This reveals $\log \log T$ could be cast as *constant* quantity in practice, since it is impossible to run $T > 10^{82}$ steps for any experiment in real-world applications.

Related work. There is a large and growing body of literature on the statistical theory of reinforcement learning that we will not attempt to thoroughly review. Detailed comparisons with existing work on RL with low-switching

²To be rigorous, we point out there are different notions for switching cost, e.g. local switching cost (Zhang et al., 2020c) and global switching cost (ours). However, we are the first to achieve $\log \log T$ switching cost with \sqrt{T} regret, regardless of its type.

cost (Bai et al., 2019; Zhang et al., 2020c; Gao et al., 2021; Wang et al., 2021) and reward-free exploration (Jin et al., 2020a; Kaufmann et al., 2021; Ménard et al., 2021; Zhang et al., 2020b; Huang et al., 2022) are given in Table 1. For a slightly more general context of this work, please refer to Appendix A and the references therein. Notably, all existing algorithms with a $\tilde{O}^*(\sqrt{T})$ regret incurs a switching cost of $O^*(\log T)$. In terms of lower bounds, our $\Omega(HSA)$ lower bound is stronger than that of Bai et al. (2019) as it operates on the *global* switching cost rather than the *local* switching cost.

The only existing algorithm with $o(\log T)$ switching cost comes from the concurrent work of Huang et al. (2022) who studied the problem of *deployment-efficient* reinforcement learning under the linear MDP model, where they require a constant switching cost. Huang et al. (2022) obtained only sample complexity bounds for pure exploration, which makes their result incompatible to our regret bounds. When compared with our results in the reward-free RL setting in the tabular setting (taking $d = SA$) their algorithm has a comparable $O(HSA)$ switching cost, but incurs a larger sample complexity in H, S, A and ϵ .

Lastly, the low-switching cost setting is often confused with its cousin — the low adaptivity setting (Perchet et al., 2016; Gao et al., 2019) (also known as *batched* RL³). Low-adaptivity requires decisions about policy changes to be made at only a few (often predefined) checkpoints but does not constrain the number of policy changes. Low-adaptive algorithms often do have low-switching cost, but lower bounds on rounds of adaptivity do not imply lower bounds for our problem. We note that our algorithms are low-adaptive, because they schedule the batch sizes of each policy ahead of time and require no adaptivity during the batches. This feature makes our algorithm more practical relative to (Bai et al., 2019; Zhang et al., 2020c; Gao et al., 2021; Wang et al., 2021) which uses adaptive switching (see, e.g., Huang et al., 2022, for a more elaborate discussion). In Section 4 we will revisit this problem and highlight the optimality of our algorithm in this alternative setting, as a byproduct.

A remark on technical novelty. The design of our algorithms involves substantial technical innovation over Bai et al. (2019); Zhang et al. (2020c); Gao et al. (2021); Wang et al. (2021). The common idea behind these $O(\log T)$ switching cost algorithms is the doubling schedule of batches in updating the policies, which originates from the UCB2 algorithm (Auer et al., 2002) for bandits. The change from UCB to UCB2 is mild enough such that existing “optimism”-based algorithms for strategic exploration designed without switching cost constraints can be adapted.

³Note that this is different from Batch RL, which is synonymous to Offline RL.

In contrast, algorithms with $O(\log \log T)$ switching cost deviates from “optimism” even in bandits problem (Cesa-Bianchi et al., 2013), thus require fresh new ideas in solving exploration when extended to RL.

The generalization of the arm elimination schedule for bandits (Cesa-Bianchi et al., 2013) to RL is nontrivial because there is an exponentially large set of deterministic policies but we need a sample efficient algorithm with polynomial dependence on H, S, A (also see the discussion before Question 1.1). Part of our solution is inspired by the reward-free exploration approach (Jin et al., 2020a), which learns to visit each (h, s, a) as much as possible by designing special rewards. However, this approach itself requires an exploration oracle, and no existing RL algorithms has $o(\log T)$ switching cost (otherwise our problem is solved). We address this problem by breaking up the exploration into stages and iteratively update a carefully constructed “absorbing MDP” that can be estimated with multiplicative error bounds. Finally, our lower bound construction is new and simple, as it essentially shows that tabular MDPs are as hard as multi-armed bandits with $\Omega(HSA)$ arms in terms of the switching cost. These techniques might be of independent interests beyond the context of this paper.

2. Problem Setup

Episodic reinforcement learning. We consider finite-horizon episodic *Markov Decision Processes* (MDP) with non-stationary transitions. The model is defined by a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, r, H, d_1 \rangle$ (Sutton & Barto, 1998), where $\mathcal{S} \times \mathcal{A}$ is the discrete state-action space and $S := |\mathcal{S}|, A := |\mathcal{A}|$ are finite. A non-stationary transition kernel has the form $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times [H] \mapsto [0, 1]$ with $P_h(s'|s, a)$ representing the probability of transition from state s , action a to next state s' at time step h . In addition, r is a known⁴ expected (immediate) reward function which satisfies $r_h(s, a) \in [0, 1]$. H is the length of the horizon and d_1 is the initial state distribution. In this work, we assume there is a fixed initial state s_1 .⁵ A policy can be seen as a series of mapping $\pi = (\pi_1, \dots, \pi_H)$, where each π_h maps each state $s \in \mathcal{S}$ to a probability distribution over actions, i.e. $\pi_h : \mathcal{S} \rightarrow \Delta(\mathcal{A})$, where $\Delta(\mathcal{A})$ is the set of probability distributions over the actions, $\forall h \in [H]$. A random trajectory $(s_1, a_1, r_1, \dots, s_H, a_H, r_H, s_{H+1})$ is generated by the following rule: s_1 is fixed, $a_h \sim \pi_h(\cdot|s_h), r_h = r(s_h, a_h), s_{h+1} \sim P(\cdot|s_h, a_h), \forall h \in [H]$.

Q-values, Bellman (optimality) equations. Given a policy π and any $h \in [H]$, the value function $V_h^\pi(\cdot) \in \mathbb{R}^S$ and Q-value function $Q_h^\pi(\cdot, \cdot) \in \mathbb{R}^{S \times A}$ are defined as: $V_h^\pi(s) =$

⁴This is due to the fact that the uncertainty of reward function is dominated by that of transition kernel in RL.

⁵The generalized case where d_1 is an arbitrary distribution can be recovered from this setting by adding one layer to the MDP.

$\mathbb{E}_\pi[\sum_{t=h}^H r_t | s_h = s]$, $Q_h^\pi(s, a) = \mathbb{E}_\pi[\sum_{t=h}^H r_t | s_h, a_h = s, a]$, $\forall s, a \in \mathcal{S}, \mathcal{A}$. Then Bellman (optimality) equation follows $\forall h \in [H]$:

$$Q_h^\pi(s, a) = r_h(s, a) + P_h(\cdot | s, a) V_{h+1}^\pi, \quad V_h^\pi = \mathbb{E}_{a \sim \pi_h} [Q_h^\pi]$$

$$Q_h^*(s, a) = r_h(s, a) + P_h(\cdot | s, a) V_{h+1}^*, \quad V_h^* = \max_a Q_h^*(\cdot, a)$$

In this work, we will consider different MDPs with respective transition kernels and reward functions. We define the value function for policy π under MDP (\tilde{r}, \tilde{P}) as below

$$V^\pi(\tilde{r}, \tilde{P}) = \mathbb{E}_\pi[\sum_{h=1}^H \tilde{r}_h | \tilde{P}].$$

Also, the notation $\mathbb{P}_\pi[\cdot | \tilde{P}]$ means the conditional probability under policy π and MDP \tilde{P} , the notation $\mathbb{E}_\pi[\cdot | \tilde{P}]$ means the conditional expectation under policy π and MDP \tilde{P} .

Regret. We measure the performance of online reinforcement learning algorithms by the regret. The regret of an algorithm is defined as

$$\text{Regret}(K) := \sum_{k=1}^K [V_1^*(s_1) - V_1^{\pi_k}(s_1)],$$

where π_k is the policy it employs at episode k . Let K be the number of episodes that the agent plan to play and total number of steps is $T := KH$.

Switching cost. We adopt the global switching cost (Bai et al., 2019), which simply measures how many times the algorithm changes its policy:

$$N_{\text{switch}} := \sum_{k=1}^{K-1} \mathbb{1}\{\pi_k \neq \pi_{k+1}\}.$$

Global switching costs are more natural than local switching costs⁶ as they measure the number of times a deployed policy (which could then run asynchronously in a distributed fashion for an extended period of time) can be changed. Bai et al. (2019)’s bound on local switching cost is thus viewed by them as a conservative surrogate of the global counterpart. Similar to Bai et al. (2019), our algorithm also uses deterministic policies only.

3. Algorithms and Explanation

Our algorithm generalizes the arm-elimination algorithm of Cesa-Bianchi et al. (2013) for bandits to a policy-elimination algorithm for RL. The high-level idea of our policy elimination algorithm is the following. We maintain a *version space* ϕ of remaining policies and iteratively refine the estimated values of all policies in ϕ while using these values to

⁶ $N_{\text{switch}}^{\text{local}} = \sum_{k=1}^{K-1} |\{(h, s) \in [H] \times \mathcal{S} : \pi_k^h(s) \neq \pi_{k+1}^h(s)\}|$

Algorithm 1 Adaptive Policy Elimination by Value Estimation (APEVE)

- 1: **Require:** Number of episodes for exploration K , r is the known deterministic reward. Universal constant C . Failure probability δ .
 - 2: **Initialize:** $T^{(k)} = K^{1-\frac{1}{2k}}$, $k \leq K_0 = O(\log \log K)$, $\phi^1 := \{\text{the set of all the deterministic policies}\}$, $\iota = \log(2HAK/\delta)$.
 - 3: **for** $k = 1, 2, \dots, K_0$ **do**
 - 4: \diamond **Number of episodes in k -th stage:**
 - 5: **if** $2(\sum_{i=1}^k T^{(i)}) \geq K$ **then**
 - 6: $T^{(k)} = \frac{K-2(\sum_{i=1}^{k-1} T^{(i)})}{2}$. (o.w. $T^{(k)} = K^{1-\frac{1}{2k}}$)
 - 7: **end if**
 - 8: \diamond **Crude exploration using Algorithm 2:**
 - 9: $\mathcal{F}^k, P^{\text{int},k} = \text{Crude Exploration}(\phi^k, T^{(k)})$.
 - 10: \diamond **Estimating \hat{P}^k using Algorithm 3:**
 - 11: $\hat{P}^k = \text{Fine Exploration}(\mathcal{F}^k, P^{\text{int},k}, T^{(k)}, \phi^k)$.
 - 12: \diamond **Adaptive policy elimination from ϕ^k :**
 - 13: $U^k = \emptyset$
 - 14: **for** $\pi \in \phi^k$ **do**
 - 15: **if** $V^\pi(r, \hat{P}^k) \leq \sup_{\hat{\pi} \in \phi^k} V^{\hat{\pi}}(r, \hat{P}^k) - 2C(\sqrt{\frac{H^5 S^2 A \iota}{T^{(k)}}} + \frac{S^3 A^2 H^5 \iota}{T^{(k)}})$ **then**
 - 16: Update $U^k \leftarrow U^k \cup \{\pi\}$.
 - 17: **end if**
 - 18: **end for**
 - 19: $\phi^{k+1} \leftarrow \phi^k \setminus U^k$.
 - 20: **end for**
-

eliminate those policies that are *certifiably* suboptimal. The hope is that towards the end of the algorithms, all policies that are not eliminated are already nearly optimal.

As we explained earlier, the challenge is to estimate the value function of all A^{HS} policies using $\text{poly}(H, S, A)$ samples. This uniform convergence problem typically involves estimating the transition kernels, but it requires solving an exploration problem to even visit a particular state-action pair once. In addition, some states cannot be visited frequently by any policy. To address these issues, we need to construct a surrogate MDP (known as an ‘‘absorbing MDP’’) with an absorbing state. This absorbing MDP replaces these troublesome states with an absorbing state s^\dagger , such that all remaining states can be visited sufficiently frequently by some policy in ϕ . Moreover, its value function uniformly approximates the original MDP for all policies of interest. This reduces the problem to estimating the transition kernel of the absorbing MDP.

Adaptive policy elimination. The overall workflow of our algorithm — Adaptive Policy Elimination by Value Estimation (APEVE) — is given in Algorithm 1 and illustrated graphically in Figure 1. It first divides a budget of K episodes into a sequence of stages with increasing length

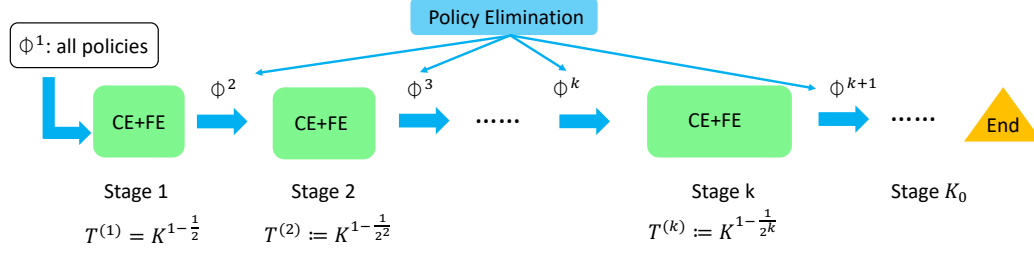


Figure 1. A visualization to explicate the procedures of APEVE (Algorithm 1). In particular, the policy elimination procedures are conducted stage by stage, with increasing size $2T^{(k)} = 2K^{1-\frac{1}{2^k}}$ at each stage. The Crude Exploration (CE) and Fine Exploration (FE) procedures within the stage apply Algorithm 2 and Algorithm 3. APEVE (Algorithm 1) is conducted to maintain a decreasing policy set ϕ^k . The number of episodes satisfies $2 \sum_{k=1}^{K_0} T^{(k)} = K$.

$2T^{(k)} := 2K^{1-1/(2^k)}$ for $k = 1, 2, 3, \dots$. By Lemma G.1, the total number of stages $K_0 = O(\log \log K)$.

Each stage involves three steps.

Step 1. Crude exploration Explore each h, s, a layer-by-layer *from scratch* based on the current version-space ϕ . Construct an absorbing MDP \tilde{P} and a *crude* intermediate estimate (P^{int}) of \tilde{P} .

Step 2. Fine exploration Explore each h, s, a with the crude estimate of the absorbing MDP. Construct a more refined estimate (\hat{P}) of the absorbing MDP's \tilde{P} .

Step 3. Policy elimination Evaluate all policies in ϕ using \hat{P} . Update the version-space ϕ by eliminating all policies whose value upper confidence bound (UCB) is smaller than the *mode* (max over $\pi \in \phi$) of the lower confidence bound (LCB).

As the algorithm proceeds, under the high probability event that our confidence bounds are valid, the optimal policy will not be eliminated. After each stage, the performance of all policies that remain will be better than the LCB of the optimal policy, which itself will get closer to the actual valuation function as we collect more data.

Next, we break down the key components in ‘‘Crude Exploration’’ and ‘‘Fine Exploration’’ and explain how they work.

Layerwise ‘‘Exploration’’ in Algorithm 2. Our goal is to learn an accurate enough⁷ estimate of $P_h(s'|s, a)$ for any tuple (h, s, a, s') and it suffices to visit this tuple $O(H^2\iota)$ times.⁸ Therefore, we try to visit each tuple as much as possible using policies from the input policy set ϕ . However, it is possible that some (h, s, a, s') tuples are hard to visit by any policy in the remaining policy set. To address this problem, we use a set \mathcal{F} to store all the tuples that have not

been visited enough times such that for the tuples not in \mathcal{F} , we can get an accurate enough estimate, while for the tuples in \mathcal{F} , we will prove that they have little influence on the value function.

In the algorithm, we apply the trick of layerwise exploration. During the exploration of the h -th layer, we use the intermediate MDP P^{int} to construct $\pi_{h,s,a}$ that can visit (h, s, a) with the largest probability under P^{int} . Then we run each $\pi_{h,s,a}$ for the same number of episodes. Using the data set \mathcal{D} we collect, the h -th layer of P^{int} (i.e. P_h^{int}) is updated using Algorithm 5.

Given infrequent tuples \mathcal{F} , the absorbing MDP is constructed as in Definition 3.1. In the construction, we first let $\tilde{P} = P$, then for $(h, s, a, s') \in \mathcal{F}$, we move the probability of $\tilde{P}_h(s'|s, a)$ to $\tilde{P}_h(s^\dagger|s, a)$.

Definition 3.1 (The absorbing MDP \tilde{P}). Given \mathcal{F} and P , $\forall (h, s, a, s') \notin \mathcal{F}$, let $\tilde{P}_h(s'|s, a) = P_h(s'|s, a)$. For any $(h, s, a, s') \in \mathcal{F}$, $\tilde{P}_h(s'|s, a) = 0$. For any $(h, s, a) \in [H] \times \mathcal{S} \times \mathcal{A}$, define $\tilde{P}_h(s^\dagger|s^\dagger, a) = 1$ and

$$\tilde{P}_h(s^\dagger|s, a) = 1 - \sum_{s' \in \mathcal{S}: (h, s, a, s') \notin \mathcal{F}} \tilde{P}_h(s'|s, a).$$

According to the construction in Algorithm 5, P^{int} is the empirical estimate of \tilde{P} . We will show that with high probability, for $(h, s, a, s') \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, either $(1 - \frac{1}{H})P_h^{int}(s'|s, a) \leq \tilde{P}_h(s'|s, a) \leq (1 + \frac{1}{H})P_h^{int}(s'|s, a)$ or $P_h^{int}(s'|s, a) = \tilde{P}_h(s'|s, a) = 0$. Based on this property, we can prove that $\pi_{h,s,a}$'s are efficient in exploration. Algorithm 5 and detailed explanation are deferred to Appendix B.

In Algorithm 2, the reward function $1_{h,s,a}$ is defined under the original MDP while P^{int} is a transition kernel of the absorbing MDP. In addition, $\pi_{h,s,a}$ is a policy under the absorbing MDP and we need to run it under the original MDP. The transition between the original MDP and the absorbing MDP is deferred to Appendix C.

Fine exploration by Algorithm 3. The idea behind Al-

⁷ $\frac{1}{H}$ -multiplicatively accurate, details in Definition E.4

⁸ Detailed proof in Lemma E.3

Algorithm 2 Crude Exploration (for constructing infrequent tuples \mathcal{F} and a reference transition P^{int})

- 1: **Input:** Policy set ϕ . Number of episodes T .
- 2: **Initialize:** $T_0 = \frac{T}{HSA}$, $C_1 = 6$, $\mathcal{F} = \emptyset$, $\mathcal{D} = \emptyset$, $\iota = \log(2HAK/\delta)$. $1_{h,s,a}$ is a reward function r' where $r'_{h'}(s', a') = \mathbb{1}[(h', s', a') = (h, s, a)]$. s^\dagger is an additional absorbing state. P^{int} is a transition kernel over the extended space $\mathcal{S} \cup \{s^\dagger\} \times \mathcal{A}$, initialized arbitrarily.
- 3: **Output:** Infrequent tuples \mathcal{F} . Intermediate transition kernel P^{int} .
- 4: **for** $h = 1, 2, \dots, H$ **do**
- 5: \diamond **Construct and run policies to visit each state-action:**
- 6: **for** $(s, a) \in \mathcal{S} \times \mathcal{A}$ **do**
- 7: $\pi_{h,s,a} = \operatorname{argmax}_{\pi \in \phi} V^\pi(1_{h,s,a}, P^{int})$.
- 8: Run $\pi_{h,s,a}$ for T_0 episodes, and add the trajectories into data set \mathcal{D} .
- 9: **end for**
- 10: **for** $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ **do**
- 11: $N_h(s, a, s') = \text{count of } (h, s, a, s') \text{ in } \mathcal{D}$.
- 12: **end for**
- 13: \diamond **Use \mathcal{F} to store the infrequent tuples:**
- 14: $\mathcal{F} = \mathcal{F} \cup \{(h, s, a, s') | N_h(s, a, s') \leq C_1 H^2 \iota\}$.
- 15: \diamond **Update the intermediate transition kernel using Algorithm 5:**
- 16: $P^{int} = \text{EstimateTransition}(\mathcal{D}, \mathcal{F}, s^\dagger, h, P^{int})$
- 17: \diamond **Clear the data set:**
- 18: Reset data set $\mathcal{D} = \emptyset$.
- 19: **end for**
- 20: **Return:** $\{\mathcal{F}, P^{int}\}$.

gorithm 3 is that with high probability, we can use P^{int} to construct policies to visit each tuple (h, s, a) with the guarantee that $\sup_{a \in \mathcal{A}, \pi \in \phi} \frac{V^\pi(1_{h,s,a}, \hat{P})}{\mu_h(s,a)} \leq 12HSA$, where μ is the distribution of our data. This inequality is similar to the result of Theorem 3.3 in (Jin et al., 2020a), which means that we can get a similar result to Lemma 3.6 in (Jin et al., 2020a) that $V^\pi(r', \hat{P})$ is an accurate estimate of $V^\pi(r', P)$ simultaneously for all $\pi \in \phi$ and any reward function r' .

For each (h, s, a) , the algorithm finds the policy $\pi_{h,s,a}$ from ϕ that visits (h, s, a) with the largest probability under P^{int} . Then each $\pi_{h,s,a}$ is run for the same number of episodes over all (h, s, a) . At last, \hat{P} is calculated as an empirical estimate of \tilde{P} by using Algorithm 5 and the data set \mathcal{D} .

4. Main Results of APEVE

In this section, we will state our main results, which formalizes the algorithmic ideas we explained in the previous section.

Theorem 4.1 (Regret and switching cost of Algorithm 1).

Algorithm 3 Fine Exploration (further exploration for accurate transition estimation)

- 1: **Input:** Infrequent tuples \mathcal{F} . Intermediate transition kernel P^{int} . Number of episodes T . Policy set ϕ .
- 2: **Initialize:** $T_0 = \frac{T}{HSA}$, $\mathcal{D} = \emptyset$. $1_{h,s,a}$ is a reward function r' where $r'_{h'}(s', a') = \mathbb{1}[(h', s', a') = (h, s, a)]$. $\hat{P} = P^{int}$.
- 3: **Output:** Empirical estimate \hat{P} .
- 4: \diamond **Construct and run policies to visit each state-action:**
- 5: **for** $(h, s, a) \in [H] \times \mathcal{S} \times \mathcal{A}$ **do**
- 6: $\pi_{h,s,a} = \operatorname{argmax}_{\pi \in \phi} V^\pi(1_{h,s,a}, P^{int})$.
- 7: Run $\pi_{h,s,a}$ for T_0 episodes, and add the trajectories into data set \mathcal{D} .
- 8: **end for**
- 9: \diamond **Construct an empirical estimate for \tilde{P} using Algorithm 5:**
- 10: **for** $h \in [H]$ **do**
- 11: $\hat{P} = \text{EstimateTransition}(\mathcal{D}, \mathcal{F}, s^\dagger, h, \hat{P})$.
- 12: **end for**
- 13: **Return** \hat{P} .

With probability $1 - \delta$, Algorithm 1 will have regret bounded by $O(\sqrt{H^5 S^2 AK} \cdot \log(\frac{2}{\delta} HAK) \cdot \log \log K + S^3 A^2 H^5 K^{\frac{1}{4}} \cdot \log(\frac{2}{\delta} HAK)) = \tilde{O}(\sqrt{H^4 S^2 AT})$. Furthermore, the global switching cost of Algorithm 1 is $O(HSA \log \log T)$ while the timestep for policy switching can be decided before the algorithm starts.

Recall that the number of episodes $K = T/H$ where T is the number of steps. This theorem says that Algorithm 1 obtains a regret bound that is optimal in T while changing (deterministic) policies for only $O(HSA \log \log T)$ times.

The proof of Theorem 4.1 is sketched in Section 6 with pointers to more detailed arguments to the full proof in the appendix. Now we discuss a few interesting aspects of the result.

Near optimal switching cost. Our algorithm achieves a switching cost that improves over existing work with \sqrt{T} regret. We also prove the following information-theoretic limit which says that the global switching cost of APEVE (Algorithm 1) is optimal up to constant.

Theorem 4.2 (Lower bound for global switching cost under optimal regret bound). *If $S \leq A^{\frac{H}{2}}$, for any algorithm with near-optimal $\tilde{O}^*(\sqrt{T})$ regret bound, the global switching cost is at least $\Omega(HSA \log \log T)$.*

As a byproduct, our proof technique naturally leads to the following lower bound for global switching cost for any algorithm with no regret.

Theorem 4.3 (Lower bound for global switching cost under sub-linear regret bound). *If $S \leq A^{\frac{H}{2}}$, for any algorithm*

with sub-linear regret bound, the global switching cost is at least $\Omega(HSA)$.

Both proofs are deferred to Appendix H. Theorem 4.3 is a stronger conclusion than the existing $\Omega(HSA)$ lower bound for local switching cost (Bai et al., 2019) because global switching cost is smaller than local switching cost. An $\Omega(HSA)$ lower bound on local switching cost can only imply an $\Omega(A)$ lower bound on global switching cost.

Near-optimal adaptivity. Interestingly, our algorithm also enjoys low-adaptivity besides low-switching cost in the batched RL setting (Perchet et al., 2016; Gao et al., 2019), because the length of each batch can be determined ahead of time and we do not require monitoring within each batch. APEVE (Algorithm 1) runs with $O(H \log \log T)$ batches. In Appendix J, we present APEVE+ (Algorithm 6), which further improves the batch complexity to $O(H + \log \log T)$ while maintaining the same regret $\tilde{O}(\sqrt{H^4 S^2 AT})$. These results nearly matches the existing lower bound $\Omega(\frac{H}{\log T} + \log \log T)$ due to Theorem B.3 in (Huang et al., 2022) (for the $\frac{H}{\log T}$ term) and Corollary 2 in (Gao et al., 2019) (for the $\log \log T$ term).

Dependence on H, S, A in the regret. As we explained earlier our regret bound is optimal in T . However, there is a gap of $\sqrt{H^2 S}$ when compared to the information-theoretic limit of $\Omega(\sqrt{H^2 SAT})$ that covers all algorithms (including those without switching cost constraints). We believe our analysis is tight and further improvements on H, S will require new algorithmic ideas. It is an intriguing open problem whether any algorithm with $\log \log T$ switching cost need to have $\Omega(S^2)$ dependence.

Computational efficiency. One weakness of our APEVE (Algorithm 1) is that it is not computationally efficient. APEVE needs to explicitly go over each element of the version spaces — the sets of remaining policies — to implement policy elimination. It remains an interesting open problem to design a polynomial-time algorithm for RL with optimal switching cost. A promising direction to achieve computational efficiency is to avoid explicitly representing the version spaces, or to reduce to “optimization oracles”. We leave a full exploration of these ideas to a future work.

5. Low Adaptive Reward-Free Exploration

In this section, we further consider the new setting of *low adaptive reward-free exploration*. Specifically, due to its nature that Crude Exploration (Algorithm 2) and Fine Exploration (Algorithm 3) do not use any information about the reward function r , these two algorithms can be leveraged in reward-free setting. LARFE (Algorithm 4) is an algorithm that tackles reward-free exploration while maintaining the low switching cost at the same time.

Algorithm 4 Low Adaptive Reward-Free Exploration (LARFE)

- 1: **Input:** Episodes for crude exploration N_0 , episodes for fine exploration N . Failure probability δ .
 - 2: **Initialize:** $\phi^1 = \{\text{the set of all deterministic policies}\}$, $\iota = \log(2HA(N_0 + N)/\delta)$.
 - 3: **Output:** $\hat{\pi}^r$ for any reward function r .
 - 4: \diamond **Crude exploration using Algorithm 2:**
 - 5: $\mathcal{F}, P^{int} = \text{Crude Exploration}(\phi^1, N_0)$.
 - 6: \diamond **Estimate \hat{P} using Algorithm 3:**
 - 7: $\hat{P} = \text{Fine Exploration}(\mathcal{F}, P^{int}, N, \phi^1)$.
 - 8: \diamond **For any reward function, output the optimal policy under empirical MDP by value iteration:**
 - 9: $\hat{\pi}^r = \operatorname{argmax}_{\pi \in \phi^1} V^\pi(r, \hat{P})$ for any r .
 - 10: **Return** $\{\hat{\pi}^r\}$.
-

In LARFE, we use Crude Exploration (Algorithm 2) to construct the infrequent tuples \mathcal{F} and the intermediate MDP P^{int} . Then the algorithm uses Fine Exploration (Algorithm 3) to get an empirical estimate \hat{P} of the absorbing MDP \hat{P} . At last, for any reward function r , the algorithm outputs the optimal policy under the empirical MDP, which can be done efficiently by value iteration.

Theorem 5.1 provides the switching cost and sample complexity of Algorithm 4 (whose proof is deferred to Appendix I).

Theorem 5.1. *The global switching cost of Algorithm 4 is bounded by $2HSA$. There exists a constant $c > 0$ such that, for any $\epsilon > 0$ and any $\delta > 0$, if the number of total episodes K satisfies that*

$$K > c \cdot \left(\frac{H^5 S^2 A \cdot \iota'}{\epsilon^2} + \frac{S^3 A H^5 \cdot \iota'}{\epsilon} \right),$$

where $\iota' = \log(\frac{HSA}{\epsilon\delta})$, then there exists a choice of N_0 and N such that $N_0 + N = K$ and with probability $1 - \delta$, for any reward function r , Algorithm 4 will output a policy $\hat{\pi}^r$ that is ϵ -optimal.

Take-away of Theorem 5.1. First of all, one key feature of LARFE is that the global switching cost is always bounded by $O(HSA)$ and this holds true for any K (i.e. independent of the PAC guarantee). Second, as a comparison to Jin et al. (2020a) regarding reward-free exploration, their episodes needed is $O(\frac{H^5 S^2 A \iota'}{\epsilon^2} + \frac{S^4 A H^7 (\iota')^3}{\epsilon})$. Our result matches this in the main term and does better in the lower order term. In addition, our algorithm achieves near optimal switching cost while the use of EULER in Jin et al. (2020a) can have switching cost equal to the number of episodes N_0 . This means that Crude Exploration (Algorithm 2) is efficient in the sense of sample complexity and switching cost when doing exploration.

Explore-First with LARFE. Given the number of episodes

K and the corresponding number of steps $T = KH$, we can apply LARFE (Algorithm 4) for the first K_0 episodes, then run the greedy policy $\hat{\pi}^r$ returned by LARFE (with r to be the real reward) for the remaining episodes. Then the regret is bounded by $HK_0 + \tilde{O}(K \cdot \sqrt{\frac{H^5 S^2 A}{K_0}})$ with high probability. By selecting $K_0 = K^{\frac{2}{3}} H S^{\frac{2}{3}} A^{\frac{1}{3}}$, the regret can be bounded by $\tilde{O}(K^{\frac{2}{3}} H^2 S^{\frac{2}{3}} A^{\frac{1}{3}}) = \tilde{O}(T^{\frac{2}{3}} H^{\frac{4}{3}} S^{\frac{2}{3}} A^{\frac{1}{3}})$, as shown in Table 1. We highlight that Explore-First w. LARFE matches the lower bound given by Theorem 4.3.

Optimal switching cost in Pure Exploration. Since any best policy identification (i.e., Pure Exploration) algorithm with polynomial sample complexity can be used to construct a no-regret learning algorithm with an Explore-First strategy, Theorem 4.3 implies that $\Omega(HSA)$ is a switching cost lower bound for the pure exploration problem too, thus also covering the task-agnostic / reward-free extensions. LARFE implies that one can achieve nearly optimal sample complexity ($\tilde{O}^*(1/\epsilon^2)$) while achieving the best possible switching cost of $O(HSA)$.

Separation of Regret Minimization and Pure Exploration in RL. Note that achieving a near-optimal $\tilde{O}^*(\sqrt{T})$ regret requires an additional factor of $\log \log T$ in the switching cost (Theorem 4.2). This provides an interesting separation of the hardness between low-adaptive regret minimization and low-adaptive pure exploration in RL.

6. Proof Overview

Due to the space constraint, we could only sketch the proof of Theorem 4.1 as the $\log \log T$ switching cost is our major contribution. The analysis involves two main parts: the switching cost bound and the regret bound. The switching cost bound directly results from the schedule of Algorithm 1.

Upper bound for switching cost. First of all, we have the conclusion that the global switching cost of Algorithm 1 is bounded by $O(HSA \log \log T)$. This is because the global switching cost of both Algorithm 2 and Algorithm 3 are bounded by HSA and the fact that the number of stages satisfy $K_0 = O(\log \log T)$.

However, such an elimination schedule requires the algorithm to run the same deterministic policy for a long period of time before being able to switch to another policy, which is the main technical challenge to the regret analysis.

Regret analysis. At the heart of the regret analysis is to construct a uniform off-policy evaluation bound that covers all remaining deterministic policies. The remaining policy set at the beginning of stage k is ϕ^k . Assume we can estimate all $V^\pi(r, P)$ ($\pi \in \phi^k$) to ϵ_k accuracy with high probability, then we can eliminate all policies that are at least $2\epsilon_k$ sub-optimal in the sense of estimated value function. Therefore, the optimal policy will not be eliminated and all the policies

remaining will be at most $4\epsilon_k$ sub-optimal with high probability. Summing up the regret of all stages, we have with high probability, the total regret is bounded by

$$\text{Regret}(K) \leq 2HT^{(1)} + \sum_{k=2}^{K_0} 2T^{(k)} \times 4\epsilon_{k-1}. \quad (1)$$

The following lemma gives an bound of ϵ_{k-1} using the model-based plug-in estimator with our estimate \hat{P} of the absorbing MDP.

Lemma 6.1. *There exists a constant C , such that with probability $1 - \delta$, it holds that for any k and $\pi \in \phi^k$,*

$$|V^\pi(r, \hat{P}^k) - V^\pi(r, P)| \leq C \left(\sqrt{\frac{H^5 S^2 A \iota}{T^{(k)}}} + \frac{S^3 A^2 H^5 \iota}{T^{(k)}} \right).$$

The proof of Lemma 6.1 involves controlling both the ‘‘bias’’ and ‘‘variance’’ part of the estimate. The ‘‘bias’’ refers to the difference between the true MDP and the absorbing MDP, and the ‘‘variance’’ refers to the statistical error in estimating the surrogate value functions of the absorbing MDP using our estimate \hat{P}^k .

From the proof of (Jin et al., 2020a), we know that if we can visit each (h, s, a) frequently enough, which means the visitation probability is maximal up to a constant factor, then the empirical transition kernel is enough for a uniform approximation to $V^\pi(r, P)$. The absorbing MDP \tilde{P} is the key to guarantee the condition of frequent visitation.

For the ease of illustration, in the following discussion, we omit the stage number k and the discussion holds true for all k . Besides, in all of the following lemmas in this section, ‘‘with high probability’’ means with probability at least $1 - \delta$ and $\iota = \log(2HAK/\delta)$.

6.1. The ‘‘bias’’: difference between P and \tilde{P}

To analyze the difference between the true MDP P and the absorbing MDP \tilde{P} , we first sketch some properties of the intermediate transition kernel P^{int} .

Accuracy of P^{int} . It holds that if the visitation number of a tuple (h, s, a, s') is larger than $O(H^2 \iota)$, with high probability⁹,

$$\left(1 - \frac{1}{H}\right) P_h^{int}(s'|s, a) \leq \tilde{P}_h(s'|s, a) \leq \left(1 + \frac{1}{H}\right) P_h^{int}(s'|s, a). \quad (2)$$

According to the definition of \mathcal{F} in Algorithm 2 and the construction of \tilde{P} , P^{int} , we have Equation (2) is true for any $(h, s, a, s') \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{S}$. Then we have¹⁰ for any $(h, s, a) \in [H] \times \mathcal{S} \times \mathcal{A}$, $\pi \in \phi$,

$$\frac{1}{4} V^\pi(1_{h,s,a}, P^{int}) \leq V^\pi(1_{h,s,a}, \tilde{P}) \leq 3V^\pi(1_{h,s,a}, P^{int}).$$

⁹Proof using empirical Bernstein’s inequality in Lemma E.3

¹⁰Proof using multiplicative bound in Lemma E.5

Because $\pi_{h,s,a} = \operatorname{argmax}_{\pi \in \phi} V^\pi(1_{h,s,a}, P^{int})$,

$$V^{\pi_{h,s,a}}(1_{h,s,a}, \tilde{P}) \geq \frac{1}{12} \sup_{\pi \in \phi} V^\pi(1_{h,s,a}, \tilde{P}), \quad (3)$$

which shows that $\pi_{h,s,a}$ is efficient in visiting the tuple (h, s, a) .

Uniform bound on $|V^\pi(r', P) - V^\pi(r', \tilde{P})|$. Now we are ready to bound $\sup_{\pi \in \phi} \sup_{r'} |V^\pi(r', P) - V^\pi(r', \tilde{P})|$ by bounding $\sup_{\pi \in \phi} \mathbb{P}_\pi[\mathcal{B}]$, where the bad event \mathcal{B} is defined as¹¹ the event where a trajectory visits some tuple in \mathcal{F} . Then we have the key lemma showing that the infrequent tuples are hard to visit by any policy in ϕ .

Lemma 6.2. *With high probability, $\sup_{\pi \in \phi} \mathbb{P}_\pi[\mathcal{B}] \leq O(\frac{S^3 A^2 H^4 \epsilon_t}{T})$.*

With Lemma 6.2, we are able to bound the difference between P and \tilde{P} in the sense of value function.

Lemma 6.3. *With high probability, it holds that*

$$0 \leq V^\pi(r', P) - V^\pi(r', \tilde{P}) \leq O(\frac{S^3 A^2 H^5 \epsilon_t}{T}),$$

for any policy $\pi \in \phi$ and reward function r' .

Therefore, the ‘‘bias’’ term $\sup_{\pi \in \phi} |V^\pi(r, P) - V^\pi(r, \tilde{P})|$ can be bounded by the right hand side of Lemma 6.3 as a special case.

6.2. The ‘‘variance’’: difference between \tilde{P} and \hat{P}

Because of the fact that with high probability, Equation (3) holds, we have the following key lemma.

Lemma 6.4. *With high probability, for any policy $\pi \in \phi$ and any reward function r' ,*

$$|V^\pi(r', \hat{P}) - V^\pi(r', \tilde{P})| = O(\sqrt{\frac{H^5 S^2 A \epsilon_t}{T}}).$$

Therefore, the ‘‘variance’’ term $\sup_{\pi \in \phi} |V^\pi(r, \hat{P}) - V^\pi(r, \tilde{P})|$ can be bounded by the right hand side of Lemma 6.4 as a special case.

6.3. Put everything together

Combining the bounds of the ‘‘bias’’ term and the ‘‘variance’’ term, because of triangular inequality, we have the conclusion in Lemma 6.1 holds. Then the proof of regret bound is completed by plugging in $\epsilon_k = O(\sqrt{\frac{H^5 S^2 A \epsilon_t}{T^{(k)}}} + \frac{S^3 A^2 H^5 \epsilon_t}{T^{(k)}})$ in equation (1).

¹¹The detailed definition can be found in Definition E.7

7. Conclusion and Future Works

This work studies the well-motivated *low switching online reinforcement learning* problem. Under the non-stationary tabular RL setting, we design the algorithm *Adaptive Policy Elimination by Value Estimation* (APEVE) which achieves $\tilde{O}(\sqrt{H^4 S^2 A T})$ regret while switching its policy for at most $O(HSA \log \log T)$ times. Under the reward-free exploration setting, we design the *Low Adaptive Reward-Free Exploration* (LARFE), which achieves $\tilde{O}(\frac{H^5 S^2 A \epsilon_t}{\epsilon^2})$ sample complexity with switching cost at most $2HSA$. We also prove lower bounds showing that these switching costs are information-theoretically optimal among algorithms that achieve nearly optimal regret or sample complexity. These results nicely settled the open problem on the optimal low-switching RL raised by Bai et al. (2019) (and revisited by Zhang et al. (2020c); Gao et al. (2021)) for the tabular setting.

It remains open to address computational efficiency, characterize the optimal dependence on H, S, A in the regret bound, study RL with function approximation, as well as to make the the algorithm practical. We leave those as future works and invite the broader RL research community to join us in the quest. Ideas and techniques developed in this paper could be of independent interest in other problems.

Acknowledgments

The research is partially supported by NSF Awards #2007117 and #2003257. The authors would like to thank Yichen Feng and Mengye Liu for helpful discussion at an early stage of this project, as well as Tianchen Yu and Chi Jin for clarifying the proof of Lemma C.2 in Jin et al. (2020a) (Lemma F.4 in this paper). DQ would like to thank Fuheng Zhao for some helpful suggestions on writing.

References

- Afsar, M. M., Crump, T., and Far, B. Reinforcement learning based recommender systems: A survey. *arXiv preprint arXiv:2101.06286*, 2021.
- Agrawal, S. and Jia, R. Posterior sampling for reinforcement learning: worst-case regret bounds. In *Advances in Neural Information Processing Systems*, pp. 1184–1194, 2017.
- Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.
- Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 263–272. JMLR. org, 2017.

- Bai, Y., Xie, T., Jiang, N., and Wang, Y.-X. Provably efficient q-learning with low switching cost. *Advances in Neural Information Processing Systems*, 32, 2019.
- Brafman, R. I. and Tennenholtz, M. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct): 213–231, 2002.
- Cesa-Bianchi, N., Dekel, O., and Shamir, O. Online learning with switching costs and other adaptive adversaries. In *Advances in Neural Information Processing Systems*, pp. 1160–1168, 2013.
- Dann, C., Lattimore, T., and Brunskill, E. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 5713–5723, 2017.
- Dann, C., Li, L., Wei, W., and Brunskill, E. Policy certificates: Towards accountable reinforcement learning. In *International Conference on Machine Learning*, pp. 1507–1516. PMLR, 2019.
- Esfandiari, H., Karbasi, A., Mehrabian, A., and Mirrokni, V. Regret bounds for batched bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7340–7348, 2021.
- Gao, M., Xie, T., Du, S. S., and Yang, L. F. A provably efficient algorithm for linear markov decision process with low switching cost. *arXiv preprint arXiv:2101.00494*, 2021.
- Gao, Z., Han, Y., Ren, Z., and Zhou, Z. Batched multi-armed bandits problem. *Advances in Neural Information Processing Systems*, 32, 2019.
- Huang, J., Chen, J., Zhao, L., Qin, T., Jiang, N., and Liu, T.-Y. Towards deployment-efficient reinforcement learning: Lower bound and optimality. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=ccWaPG19Hq>.
- Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(4), 2010.
- Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pp. 4863–4873, 2018.
- Jin, C., Krishnamurthy, A., Simchowitz, M., and Yu, T. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pp. 4870–4879. PMLR, 2020a.
- Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pp. 2137–2143. PMLR, 2020b.
- Kaufmann, E., Ménard, P., Domingues, O. D., Jonsson, A., Leurent, E., and Valko, M. Adaptive reward-free exploration. In *Algorithmic Learning Theory*, pp. 865–891. PMLR, 2021.
- Kearns, M. and Singh, S. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2-3):209–232, 2002.
- Krishnan, S., Yang, Z., Goldberg, K., Hellerstein, J., and Stoica, I. Learning to optimize join queries with deep reinforcement learning. *arXiv preprint arXiv:1808.03196*, 2018.
- Matsushima, T., Furuta, H., Matsuo, Y., Nachum, O., and Gu, S. Deployment-efficient reinforcement learning via model-based offline optimization. In *International Conference on Learning Representations*, 2020.
- Maurer, A. and Pontil, M. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.
- Ménard, P., Domingues, O. D., Jonsson, A., Kaufmann, E., Leurent, E., and Valko, M. Fast active learning for pure exploration in reinforcement learning. In *International Conference on Machine Learning*, pp. 7599–7608. PMLR, 2021.
- Osband, I., Russo, D., and Van Roy, B. (more) efficient reinforcement learning via posterior sampling. *Advances in Neural Information Processing Systems*, 26, 2013.
- Perchet, V., Rigollet, P., Chassang, S., and Snowberg, E. Batched bandit problems. *The Annals of Statistics*, 44(2): 660–681, 2016.
- Raccuglia, P., Elbert, K. C., Adler, P. D., Falk, C., Wenny, M. B., Mollo, A., Zeller, M., Friedler, S. A., Schrier, J., and Norquist, A. J. Machine-learning-assisted materials discovery using failed experiments. *Nature*, 533(7601): 73–76, 2016.
- Simchi-Levi, D. and Xu, Y. Phase transitions and cyclic phenomena in bandits with switching constraints. *Advances in Neural Information Processing Systems*, 32, 2019.
- Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
- Wang, R., Du, S. S., Yang, L., and Salakhutdinov, R. R. On reward-free reinforcement learning with linear function approximation. *Advances in neural information processing systems*, 33:17816–17826, 2020.

- Wang, T., Zhou, D., and Gu, Q. Provably efficient reinforcement learning with linear function approximation under adaptivity constraints. *Advances in Neural Information Processing Systems*, 34, 2021.
- Xu, Z., Tang, J., Meng, J., Zhang, W., Wang, Y., Liu, C. H., and Yang, D. Experience-driven networking: A deep reinforcement learning based approach. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 1871–1879. IEEE, 2018.
- Yu, C., Liu, J., Nemati, S., and Yin, G. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021.
- Zanette, A. and Brunskill, E. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In *International Conference on Machine Learning*, pp. 7304–7312. PMLR, 2019.
- Zanette, A., Lazaric, A., Kochenderfer, M. J., and Brunskill, E. Provably efficient reward-agnostic navigation with linear value iteration. *Advances in Neural Information Processing Systems*, 33:11756–11766, 2020.
- Zhang, X., Singla, A., et al. Task-agnostic exploration in reinforcement learning. *Advances in Neural Information Processing Systems*, 2020a.
- Zhang, Z., Du, S. S., and Ji, X. Nearly minimax optimal reward-free reinforcement learning. *arXiv preprint arXiv:2010.05901*, 2020b.
- Zhang, Z., Zhou, Y., and Ji, X. Almost optimal model-free reinforcement learning via reference-advantage decomposition. *Advances in Neural Information Processing Systems*, 33:15198–15207, 2020c.

A. Extended related work

Low regret reinforcement learning algorithms There has been a long line of works (Brafman & Tennenholtz, 2002; Kearns & Singh, 2002; Jaksch et al., 2010; Osband et al., 2013; Agrawal & Jia, 2017; Jin et al., 2018) focusing on regret minimization for online reinforcement learning. Azar et al. (2017) used model-based algorithm (UCB-Q-values) to achieve the optimal regret bound $\tilde{O}(\sqrt{HSAT})$ for stationary tabular MDP. Dann et al. (2019) used algorithm ORLC to match the lower bound of regret and give policy certificates at the same time. Zhang et al. (2020c) used Q-learning type algorithm (UCB-advantage) to achieve the optimal $\tilde{O}(\sqrt{H^2SAT})$ regret for non-stationary tabular MDP. Zanette & Brunskill (2019) designed the algorithm EULER to get a problem dependent regret bound, which also matches the lower bound.

Reward-free exploration Jin et al. (2020a) first studied the problem of reward-free exploration, they used a regret minimization algorithm EULER (Zanette & Brunskill, 2019) to visit each state as much as possible. The sample complexity for their algorithm is $\tilde{O}(H^5S^2A/\epsilon^2)$ episodes. Kaufmann et al. (2021) designed an algorithm RF-UCRL by building upper confidence bound for any reward function and any policy, their algorithm needs of order $\tilde{O}((S^2AH^4/\epsilon^2))$ episodes to output a near-optimal policy for any reward function with high probability. Ménard et al. (2021) constructed a novel exploration bonus of order $\frac{1}{n}$ and their algorithm achieved sample complexity of $\tilde{O}((S^2AH^3/\epsilon^2))$. Zhang et al. (2020b) considered a more general setting with stationary transition kernel and uniformly bounded reward. They designed a novel condition to achieve the optimal sample complexity $\tilde{O}((S^2A/\epsilon^2))$ under their setting. Also, their result can be used to achieve $\tilde{O}((S^2AH^2/\epsilon^2))$ sample complexity under traditional setting where $r_h \in [0, 1]$, this result matches the lower bound. Wang et al. (2020) and Zanette et al. (2020) analyzed reward-free exploration under the setting of linear MDP. There is a similar setting named task-agnostic exploration. Zhang et al. (2020a) designed an algorithm: UCB-Zero that finds ϵ -optimal policies for N arbitrary tasks after at most $\tilde{O}(H^5SA \log N/\epsilon^2)$ exploration episodes. A concurrent work (Huang et al., 2022) analyzed low adaptive reward-free exploration under linear MDP. In our work, we consider low adaptive reward-free exploration under tabular MDP, our switching cost is of the same order as (Huang et al., 2022) and our sample complexity is much smaller than theirs if directly plugging in $d = SA$ in their bounds.

Bandit algorithms with limited adaptivity There has been a long history of works about multi-armed bandit algorithms with low adaptivity (Cesa-Bianchi et al., 2013; Perchet et al., 2016; Gao et al., 2019; Esfandiari et al., 2021). Cesa-Bianchi et al. (2013) designed an algorithm with $\tilde{O}(\sqrt{KT})$ regret using $O(\log \log T)$ batches. Perchet et al. (2016) proved a regret lower bound of $\Omega(T^{\frac{1}{1-2^{1-M}}})$ for algorithms within M batches under 2-armed bandit setting, which means $\Omega(\log \log T)$ batches are necessary for a regret bound of $\tilde{O}(\sqrt{T})$. The result is generalized to K -armed bandit by Gao et al. (2019). We will show the connection and difference between this setting and the low switching setting. In batched bandit problems, the agent decides a sequence of arms and observes the reward of each arm after all arms in that sequence are pulled. More formally, at the beginning of each batch, the agent decides a list of arms to be pulled. Afterwards, a list of (arm, reward) pairs is given to the agent. Then the agent decides about the next batch. The batch sizes could be chosen non-adaptively or adaptively. In a non-adaptive algorithm, the batch sizes should be decided before the algorithm starts, while in an adaptive algorithm, the batch sizes may depend on the previous observations. (Esfandiari et al., 2021). Under the switching cost setting, the algorithm can monitor the data stream and decide to change policy at any time, which means an algorithm with low switching cost can have $\Omega(T)$ batches. In addition, algorithms with limited batches can have large switching cost because in one batch, the algorithm can use different policies. Under batched bandit problem, algorithms with at most M batches can have a MK upper bound for switching cost. However, if we generalize batched bandit to batched RL, algorithms with at most M batches can have $A^{SH}M$ switching cost in the worst case. We conclude that an upper bound of batches and an upper bound of switching cost can not imply each other in the worst case.

B. Missing algorithm: EstimateTransition (Algorithm 5) and some explanation

Algorithm 5 receives a data set \mathcal{D} , a set \mathcal{F} of infrequent tuples, a transition kernel P and a target layer h which we want to update. The goal is to update the h -th layer of the input transition kernel P while the remaining layers stay unchanged. The construction of P_h is for such tuples in \mathcal{F} , the transition kernel $P_h(s'|s, a)$ is 0. For the states not in \mathcal{F} , $P_h(s'|s, a)$ is the empirical estimate. At last, $P_h(s^\dagger|s, a) = 1 - \sum_{s' \in \mathcal{S}: (h, s, a, s') \notin \mathcal{F}} P_h(s'|s, a)$ holds so that P_h is a valid transition kernel. For a better understanding, the construction is similar to the construction of \tilde{P} . We first let P be the empirical estimate based on \mathcal{D} , then for $(h, s, a, s') \in \mathcal{F}$, we move the probability of $P_h(s'|s, a)$ to $P_h(s^\dagger|s, a)$.

Algorithm 5 Compute Transition Kernel (EstimateTransition)

```

1: Require: Data set  $\mathcal{D}$ , infrequent tuples  $\mathcal{F}$ , absorbing state  $s^\dagger$ , the target layer  $h$ , transition kernel  $P$ .
2: Output: Estimated transition kernel  $P$  from data set  $\mathcal{D}$ .
3:  $\diamond$  Count the visitation number of each state-action pairs from the target layer  $h$ :
4: for  $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$  do
5:    $N_h(s, a, s') = \text{count of } (h, s, a, s') \text{ in } \mathcal{D}$ .
6:    $N_h(s, a) = \text{count of } (h, s, a) \text{ in } \mathcal{D}$ .
7: end for
8:  $\diamond$  Update the  $h$ -th layer of the transition kernel:
9: for  $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$  s.t.  $(h, s, a, s') \in \mathcal{F}$  do
10:   $P_h(s'|s, a) = 0$ .
11: end for
12: for  $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$  s.t.  $(h, s, a, s') \notin \mathcal{F}$  do
13:   $P_h(s'|s, a) = \frac{N_h(s, a, s')}{N_h(s, a)}$ .
14: end for
15: for  $(s, a) \in \mathcal{S} \times \mathcal{A}$  do
16:   $P_h(s^\dagger|s, a) = 1 - \sum_{s' \in \mathcal{S}: (h, s, a, s') \notin \mathcal{F}} P_h(s'|s, a)$ .
17: end for
18: for  $a \in \mathcal{A}$  do
19:   $P_h(s^\dagger|s^\dagger, a) = 1$ .
20: end for
21: Return  $P$ .
    
```

C. Transition between original MDP and absorbing MDP

For any reward function r defined on the original MDP P , we abuse the notation and use it on the absorbing version. We extend the definition as:

$$r(s, a) = \begin{cases} r(s, a), & s \in \mathcal{S}, \\ 0, & s = s^\dagger. \end{cases}$$

For any policy π defined on the original MDP P , we abuse the notation and use it on the absorbing version. We extend the definition as:

$$\pi(\cdot|s) = \begin{cases} \pi(\cdot|s), & s \in \mathcal{S}, \\ \text{arbitrary distribution}, & s = s^\dagger. \end{cases}$$

Under this definition of r and π , the expected reward under the absorbing MDP is fixed because once we enter the absorbing state s^\dagger , we will not get any more reward, so the policy at s^\dagger has no influence on the value function. For any policy π defined under the absorbing MDP, we can directly apply it under the true MDP and analyze its value function because $\pi_h(\cdot|s)$ has definition for any $s \in \mathcal{S}$.

In this paper, P is the real MDP, which is under original MDP. In each stage, \tilde{P} is an absorbing MDP constructed based on infrequent tuples \mathcal{F} and the real MDP P . When we run the algorithm, we don't know the exact \tilde{P} , but we know the intermediate transition kernel P^{int} , which is also an absorbing MDP. In Algorithm 3, the \hat{P} we construct is the empirical estimate of \tilde{P} , which is also an absorbing MDP. In the proof of this paper, a large part of discussion is under the framework of absorbing MDP. When we specify that the discussion is under absorbing MDP with absorbing state s^\dagger , any transition kernel P' satisfies $P'_h(s^\dagger|s^\dagger, a) = 1$ for any $(a, h) \in \mathcal{A} \times [H]$. For the reward functions in this paper, they are all defined under original MDP, when applied under absorbing MDP, the transition rule follows what we just discussed.

D. Technical lemmas

Lemma D.1 (Bernstein's inequality). *Let x_1, \dots, x_n be independent bounded random variables such that $\mathbb{E}[x_i] = 0$ and $|x_i| \leq A$ with probability 1. Let $\sigma^2 = \frac{1}{n} \sum_{i=1}^n \text{Var}[x_i]$, then with probability $1 - \delta$ we have*

$$\left| \frac{1}{n} \sum_{i=1}^n x_i \right| \leq \sqrt{\frac{2\sigma^2 \log(2/\delta)}{n}} + \frac{2A}{3n} \log(2/\delta).$$

Lemma D.2 (Empirical Bernstein's inequality (Maurer & Pontil, 2009)). *Let x_1, \dots, x_n be i.i.d random variables such that $|x_i| \leq A$ with probability 1. Let $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$, and $\widehat{V}_n = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$, then with probability $1 - \delta$ we have*

$$\left| \frac{1}{n} \sum_{i=1}^n x_i - \mathbb{E}[x] \right| \leq \sqrt{\frac{2\widehat{V}_n \log(2/\delta)}{n}} + \frac{7A}{3n} \log(2/\delta).$$

Lemma D.3 (Lemma F.4 in (Dann et al., 2017)). *Let F_i for $i = 1 \dots$ be a filtration and X_1, \dots, X_n be a sequence of Bernoulli random variables with $\mathbb{P}(X_i = 1 | F_{i-1}) = P_i$ with P_i being F_{i-1} -measurable and X_i being F_i measurable. It holds that*

$$\mathbb{P}[\exists n : \sum_{t=1}^n X_t < \sum_{t=1}^n P_t/2 - W] \leq e^{-W}.$$

Lemma D.4. *Let F_i for $i = 1 \dots$ be a filtration and X_1, \dots, X_n be a sequence of Bernoulli random variables with $\mathbb{P}(X_i = 1 | F_{i-1}) = P_i$ with P_i being F_{i-1} -measurable and X_i being F_i measurable. It holds that*

$$\mathbb{P}[\exists n : \sum_{t=1}^n X_t < \sum_{t=1}^n P_t/2 - \iota] \leq \frac{\delta}{HAK},$$

where $\iota = \log(2HAK/\delta)$.

Proof of Lemma D.4. Directly plug in $W = \iota$ in lemma D.3. □

E. Proof of lemmas regarding Crude Exploration (Algorithm 2)

First, we want to highlight that in this paper, under the absorbing MDP, \mathcal{S} only denotes the original states, the absorbing state $s^\dagger \notin \mathcal{S}$.

An upper bound for global switching cost is straightforward.

Lemma E.1. *The global switching cost of Algorithm 2 is bounded by HSA .*

Proof of Lemma E.1. There are at most HSA different $\pi_{h,s,a}$'s, Algorithm 2 will just run each policy for several episodes. □

We can bound the difference between \tilde{P} and P^{int} by empirical Bernstein's inequality (Lemma D.2).

Lemma E.2. *Define the event \mathcal{W} as: $\forall (h, s, a, s') \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ such that $(h, s, a, s') \notin \mathcal{F}$,*

$$|P_h^{int}(s'|s, a) - \tilde{P}_h(s'|s, a)| \leq \sqrt{\frac{2P_h^{int}(s'|s, a)\iota}{N_h(s, a)}} + \frac{7\iota}{3N_h(s, a)}.$$

Then with probability $1 - \frac{S^2\delta}{K}$, the event \mathcal{W} holds. In addition, we have that $\forall (h, s, a, s') \in \mathcal{F}$,

$$\tilde{P}_h(s'|s, a) = P_h^{int}(s'|s, a) = 0.$$

Proof of Lemma E.2. The first part is because of Lemma D.2 and a union bound on all $(h, s, a, s') \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{S}$. The second part is because of Definition 3.1 and the definition of P^{int} in Algorithm 2. □

Lemma E.3. *Conditioned on the event \mathcal{W} in Lemma E.2, $\forall (h, s, a, s') \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ such that $(h, s, a, s') \notin \mathcal{F}$, it holds that*

$$\left(1 - \frac{1}{H}\right)P_h^{int}(s'|s, a) \leq \tilde{P}_h(s'|s, a) \leq \left(1 + \frac{1}{H}\right)P_h^{int}(s'|s, a).$$

Proof of Lemma E.3. Because the event \mathcal{W} is true, we have $\forall (h, s, a, s') \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ such that $(h, s, a, s') \notin \mathcal{F}$,

$$|P_h^{int}(s'|s, a) - \tilde{P}_h(s'|s, a)| \leq \sqrt{\frac{2P_h^{int}(s'|s, a)\iota}{N_h(s, a)}} + \frac{7\iota}{3N_h(s, a)}.$$

By the definition of \mathcal{F} that $\mathcal{F} = \{(h, s, a, s') | N_h(s, a, s') \leq C_1 H^2 \iota\}$, $\forall (h, s, a, s') \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ such that $(h, s, a, s') \notin \mathcal{F}$, $N_h(s, a, s') \geq C_1 H^2 \iota$.

Recall that for such $(h, s, a, s') \notin \mathcal{F}$, $P_h^{int}(s'|s, a) = \frac{N_h(s, a, s')}{N_h(s, a)}$, we have

$$\begin{aligned} |P_h^{int}(s'|s, a) - \tilde{P}_h(s'|s, a)| &\leq \sqrt{\frac{2\iota}{N_h(s, a, s')}}P_h^{int}(s'|s, a) + \frac{7P_h^{int}(s'|s, a)\iota}{3N_h(s, a, s')} \\ &\leq \left(\sqrt{\frac{2}{C_1}} + \frac{7}{3C_1 H}\right) \cdot \frac{1}{H}P_h^{int}(s'|s, a) \\ &\leq \frac{1}{H}P_h^{int}(s'|s, a). \end{aligned}$$

The first inequality is because of the definition of P^{int} . The second inequality is because of the definition of \mathcal{F} . The last inequality is because of the choice of $C_1 = 6$.

Then the proof is completed by arranging $|P_h^{int}(s'|s, a) - \tilde{P}_h(s'|s, a)| \leq \frac{1}{H}P_h^{int}(s'|s, a)$. \square

From Lemma E.3, we can see that for those tuples (h, s, a, s') not in \mathcal{F} , the estimate of the transition kernel satisfies $\left(1 - \frac{1}{H}\right)P_h^{int}(s'|s, a) \leq \tilde{P}_h(s'|s, a) \leq \left(1 + \frac{1}{H}\right)P_h^{int}(s'|s, a)$ with high probability. In addition, for those states $(h, s, a, s') \in \mathcal{F}$, $P_h^{int}(s'|s, a) = \tilde{P}_h(s'|s, a) = 0$, which means this inequality holds for all $(h, s, a, s') \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{S}$. For simplicity, we use a new definition θ -multiplicatively accurate to describe the relationship between P^{int} and \tilde{P} .

Definition E.4 (θ -multiplicatively accurate for transition kernels (under absorbing MDP)). Under the absorbing MDP with absorbing state s^\dagger , a transition kernel P' is θ -multiplicatively accurate to another transition kernel P'' if

$$(1 - \theta)P'_h(s'|s, a) \leq P''_h(s'|s, a) \leq (1 + \theta)P'_h(s'|s, a)$$

for all $(h, s, a, s') \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ and there is no requirement for the case when $s' = s^\dagger$.

Because of Lemma E.2 and Lemma E.3, we have that with probability $1 - \frac{S^2\delta}{K}$, P^{int} is $\frac{1}{H}$ -multiplicatively accurate to \tilde{P} . Next, we will compare the visitation probability of each state (h, s, a) under two transition kernels that are close to each other.

Lemma E.5. *Define $1_{h, s, a}$ to be the reward function r' such that $r'_{h'}(s', a') = \mathbb{1}[(h', s', a') = (h, s, a)]$. Similarly, define $1_{h, s}$ to be the reward function r' such that $r'_{h'}(s', a') = \mathbb{1}[(h', s') = (h, s)]$. Then $V^\pi(1_{h, s, a}, P')$ and $V^\pi(1_{h, s}, P')$ denote the visitation probability of (h, s, a) and (h, s) , respectively, under π and P' . Under the absorbing MDP with absorbing state s^\dagger , if P' is $\frac{1}{H}$ -multiplicatively accurate to P'' , for any policy π and any $(h, s, a) \in [H] \times \mathcal{S} \times \mathcal{A}$, it holds that*

$$\frac{1}{4}V^\pi(1_{h, s, a}, P') \leq V^\pi(1_{h, s, a}, P'') \leq 3V^\pi(1_{h, s, a}, P').$$

Proof of Lemma E.5. Under the absorbing MDP, for any trajectory $\tau = \{s_1, a_1, \dots, s_h, a_h\}$ (truncated at time step h) such that $(s_h, a_h) = (s, a)$ and $s \in \mathcal{S}$, we have $s_{h'} \neq s^\dagger$ for any $h' \leq h - 1$. Note that we only need to consider the trajectory truncated at time step h because the visitation to (h, s, a) only depends on this part of trajectory. We have for any truncated trajectory $\tau = \{s_1, a_1, \dots, s_h, a_h\}$ such that $(s_h, a_h) = (s, a)$, it holds that

$$\mathbb{P}_\pi[\tau | P''] = \prod_{i=1}^h \pi_i(a_i | s_i) \times \prod_{i=1}^{h-1} P''_i(s_{i+1} | s_i, a_i)$$

$$\begin{aligned} &\leq \left(1 + \frac{1}{H}\right)^H \prod_{i=1}^h \pi_i(a_i|s_i) \times \prod_{i=1}^{h-1} P'_i(s_{i+1}|s_i, a_i) \\ &\leq 3\mathbb{P}_\pi[\tau|P']. \end{aligned}$$

The first inequality is because when $s_{i+1} \neq s^\dagger$, $(1 - \frac{1}{H})P'_i(s_{i+1}|s_i, a_i) \leq P''_i(s_{i+1}|s_i, a_i) \leq (1 + \frac{1}{H})P'_i(s_{i+1}|s_i, a_i)$. Let $\tau_{h,s,a}$ be the set of truncated trajectories such that $(s_h, a_h) = (s, a)$. Then

$$V^\pi(1_{h,s,a}, P'') = \sum_{\tau \in \tau_{h,s,a}} \mathbb{P}_\pi[\tau|P''] \leq 3 \sum_{\tau \in \tau_{h,s,a}} \mathbb{P}_\pi[\tau|P'] = 3V^\pi(1_{h,s,a}, P').$$

The left side of the inequality can be proven in a similar way, with $(1 - \frac{1}{H})^H \geq \frac{1}{4}$ when $H \geq 2$. \square

Under the absorbing MDP with absorbing state s^\dagger , we have shown (in Lemma E.3) that with high probability, for any $(h, s, a, s') \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, it holds that $(1 - \frac{1}{H})P_h^{int}(s'|s, a) \leq \tilde{P}_h(s'|s, a) \leq (1 + \frac{1}{H})P_h^{int}(s'|s, a)$. Combined with Lemma E.5, we have with high probability, for any policy π and any $(h, s, a) \in [H] \times \mathcal{S} \times \mathcal{A}$,

$$\frac{1}{4}V^\pi(1_{h,s,a}, P^{int}) \leq V^\pi(1_{h,s,a}, \tilde{P}) \leq 3V^\pi(1_{h,s,a}, P^{int}).$$

Careful readers may find that for the visitation probability to the absorbing state s^\dagger , this inequality may not be true. However, this is not a problem for our propose, because we do not need to explore the absorbing state or consider the visitation probability of s^\dagger .

The structure of the absorbing MDP also gives rise to the following lemma about the relationship between \tilde{P} and P .

Lemma E.6. For any policy π and any $(h, s, a) \in [H] \times \mathcal{S} \times \mathcal{A}$,

$$V^\pi(1_{h,s,a}, P) \geq V^\pi(1_{h,s,a}, \tilde{P}).$$

Proof of Lemma E.6. For any truncated trajectory τ that arrives at (h, s, a) under \tilde{P} , $\mathbb{P}_\pi[\tau|P] = \mathbb{P}_\pi[\tau|\tilde{P}]$. \square

Next, we will define the following bad event and explain the decomposition of its probability.

Definition E.7 (Bad event \mathcal{B} and \mathcal{B}_h under original MDP). For a trajectory $\{s_1, a_1, \dots, s_H, a_H, s_{H+1}\}$ under original MDP and some policy, define \mathcal{B} to be the event where there exists h such that $(h, s_h, a_h, s_{h+1}) \in \mathcal{F}$. Define \mathcal{B}_h , for $h = 1, 2, \dots, H$ to be the event that $(h, s_h, a_h, s_{h+1}) \in \mathcal{F}$ and $\forall h' \leq h-1, (h', s_{h'}, a_{h'}, s_{h'+1}) \notin \mathcal{F}$.

We have that under the original MDP, \mathcal{B} is the event that the trajectory finally enters \mathcal{F} and \mathcal{B}_h is the event that the trajectory first enters \mathcal{F} at time step $h+1$.

Definition E.8 (Bad event \mathcal{B} and \mathcal{B}_h under absorbing MDP). For a trajectory $\{s_1, a_1, \dots, s_H, a_H, s_{H+1}\}$ under absorbing MDP and some policy, define \mathcal{B} to be the event where there exists h such that $s_h = s^\dagger$. Define \mathcal{B}_h , for $h = 1, 2, \dots, H$ to be the event that $s_{h+1} = s^\dagger$ and $\forall h' \leq h, s_{h'} \neq s^\dagger$.

We have that under the absorbing MDP, \mathcal{B} is the event that the trajectory finally enters s^\dagger and \mathcal{B}_h is the event that the trajectory first enters s^\dagger at time step $h+1$. Note that under either the original MDP or the absorbing MDP, \mathcal{B} is a disjoint union of \mathcal{B}_h and $\mathbb{P}(\mathcal{B}) = \sum_{h=1}^H \mathbb{P}(\mathcal{B}_h)$.

Now we are ready to prove the key lemma about the difference between \tilde{P} and P . We will prove Lemma 6.2 and state an improved version under the special case where $\phi = \phi^1$.

Lemma E.9 (Restate Lemma 6.2). Conditioned on \mathcal{W} in Lemma E.2, with probability $1 - \frac{\delta}{AK}$, $\sup_{\pi \in \phi} \mathbb{P}_\pi[\mathcal{B}] \leq \frac{168S^3A^2H^4\epsilon}{T}$.

Proof of Lemma E.9. We will prove that $\forall h \in [H]$, with probability $1 - \frac{\delta}{HAK}$, $\sup_{\pi \in \phi} \mathbb{P}_\pi[\mathcal{B}_h] \leq \frac{168S^3A^2H^3\epsilon}{T}$. First, recall that the event \mathcal{B}_h means $(h, s_h, a_h, s_{h+1}) \in \mathcal{F}$ and $\forall h' \leq h-1, (h', s_{h'}, a_{h'}, s_{h'+1}) \notin \mathcal{F}$ under the original

MDP. Also, \mathcal{B}_h means that $s_{h+1} = s^\dagger$ and $\forall h' \leq h, s_{h'} \neq s^\dagger$ under the absorbing MDP. We have

$$\begin{aligned}
 \mathbb{P}_\pi[\mathcal{B}_h|P] &= \sum_{\tau \in \mathcal{B}_h} \mathbb{P}_\pi[\tau|P] \\
 &= \sum_{\tau_{:h+1} \in \mathcal{B}_h} \mathbb{P}_\pi[(s_1, a_1, \dots, s_h, a_h)|P] P_h(s_{h+1}|s_h, a_h) \\
 &= \sum_{\tau_{:h+1} \in \mathcal{B}_h} \mathbb{P}_\pi[(s_1, a_1, \dots, s_h, a_h)|\tilde{P}] P_h(s_{h+1}|s_h, a_h) \\
 &= \sum_{s \in \mathcal{S}, a} V^\pi(1_{h,s,a}, \tilde{P}) \sum_{s' \in \mathcal{S}: (h,s,a,s') \in \mathcal{F}} P_h(s'|s, a) \\
 &= \sum_{s \in \mathcal{S}, a} V^\pi(1_{h,s,a}, \tilde{P}) \tilde{P}_h(s^\dagger|s, a) \\
 &= \mathbb{P}_\pi[\mathcal{B}_h|\tilde{P}],
 \end{aligned} \tag{4}$$

where $\tau_{:h+1}$ in line 2 and line 3 means the trajectory τ truncated at s_{h+1} . Note that we only need to consider the trajectory truncated at s_{h+1} because the event \mathcal{B}_h only depends on this part of trajectory. The \mathcal{B}_h in the first three lines are defined under original MDP, while the \mathcal{B}_h in the last line is defined under absorbing MDP. The third equation is because for $(h, s, a, s') \notin \mathcal{F}$, $P = \tilde{P}$. The fourth equation is because there is a bijection between trajectories that arrive at $(h, s, a) \in [H] \times \mathcal{S} \times \mathcal{A}$ under absorbing MDP and trajectories in \mathcal{B}_h that arrive at the same (h, s, a) under the original MDP. The fifth equation is because of the definition of \tilde{P} . The last equation is because of the definition of \mathcal{B}_h under the absorbing MDP \tilde{P} .

Recall that in Algorithm 2, $\pi_{h,s,a} = \operatorname{argmax}_{\pi \in \phi} V^\pi(1_{h,s,a}, P^{int})$, then because of Lemma E.5 and the fact that when constructing $\pi_{h,s,a}$, the first $h-1$ layers of P^{int} is already same to the final output P^{int} , we have

$$\begin{aligned}
 V^{\pi_{h,s,a}}(1_{h,s,a}, \tilde{P}) &\geq \frac{1}{4} V^{\pi_{h,s,a}}(1_{h,s,a}, P^{int}) \\
 &= \frac{1}{4} \sup_{\pi \in \phi} V^\pi(1_{h,s,a}, P^{int}) \\
 &\geq \frac{1}{12} \sup_{\pi \in \phi} V^\pi(1_{h,s,a}, \tilde{P}),
 \end{aligned} \tag{5}$$

where the two inequalities are because of Lemma E.5.

Define π_h to be a policy that chooses each $\pi_{h,s,a}$ with probability $\frac{1}{SA}$ for any $(s, a) \in \mathcal{S} \times \mathcal{A}$. Then we have

$$\begin{aligned}
 V^{\pi_h}(1_{h,s,a}, \tilde{P}) &\geq \frac{1}{SA} V^{\pi_{h,s,a}}(1_{h,s,a}, \tilde{P}) \\
 &\geq \frac{1}{12SA} \sup_{\pi \in \phi} V^\pi(1_{h,s,a}, \tilde{P}).
 \end{aligned} \tag{6}$$

Note that in our Algorithm 2, each policy $\pi_{h,s,a}$ will be run for $\frac{T}{HSA}$ episodes. Then for any event \mathcal{E} , we have that

$$\sum_{s,a} \frac{T}{HSA} \times \mathbb{P}_{\pi_{h,s,a}}[\mathcal{E}|\tilde{P}] = \frac{T}{H} \times \mathbb{P}_{\pi_h}[\mathcal{E}|\tilde{P}].$$

We will assume that running each $\pi_{h,s,a}$ for $\frac{T}{HSA}$ episodes is equivalent to running π_h for $\frac{T}{H}$ episodes because with Lemma D.4, we can derive the same lower bound for the total number of event \mathcal{E} . In the remaining part of the proof, we will analyze assuming we run π_h for $\frac{T}{H}$ episodes.

With the definition of π_h , we have

$$\begin{aligned}
 \mathbb{P}_{\pi_h}[\mathcal{B}_h|P] &= \mathbb{P}_{\pi_h}[\mathcal{B}_h|\tilde{P}] \\
 &= \sum_{s \in \mathcal{S}, a} V^{\pi_h}(1_{h,s,a}, \tilde{P}) \tilde{P}_h(s^\dagger|s, a) \\
 &\geq \frac{1}{12SA} \sum_{s \in \mathcal{S}, a} \sup_{\pi \in \phi} V^\pi(1_{h,s,a}, \tilde{P}) \tilde{P}_h(s^\dagger|s, a)
 \end{aligned}$$

$$\begin{aligned}
 &\geq \frac{1}{12SA} \sup_{\pi \in \phi} \sum_{s \in \mathcal{S}, a} V^\pi(1_{h,s,a}, \tilde{P}) \tilde{P}_h(s^\dagger | s, a) \\
 &= \frac{1}{12SA} \sup_{\pi \in \phi} \mathbb{P}_\pi[\mathcal{B}_h | \tilde{P}] \\
 &= \frac{1}{12SA} \sup_{\pi \in \phi} \mathbb{P}_\pi[\mathcal{B}_h | P],
 \end{aligned}$$

where $\mathcal{B}_h | P$ is defined under original MDP while $\mathcal{B}_h | \tilde{P}$ is defined under absorbing MDP. The first and the last equation is because of (4). The first inequality is because of (6). The second inequality is because the summation of maximum is larger than the maximum of summation.

Suppose $\sup_{\pi \in \phi} \mathbb{P}_\pi[\mathcal{B}_h | P] \geq \frac{168S^3 A^2 H^3 \iota}{T}$, then we have

$$\mathbb{P}_{\pi_h}[\mathcal{B}_h | P] \geq \frac{1}{12SA} \sup_{\pi \in \phi} \mathbb{P}_\pi[\mathcal{B}_h | P] \geq \frac{14S^2 AH^3 \iota}{T}.$$

Therefore, by Lemma D.4, with probability $1 - \frac{\delta}{HAK}$, \mathcal{B}_h occurs for at least $\frac{T}{H} \times \frac{14S^2 AH^3 \iota}{2T} - \iota > 6S^2 AH^2 \iota$ times during the exploration of the h -th layer. However, by the definition of \mathcal{F} , at each time step h , for each (s, a, s') , the event $\mathcal{B}_h \cap \{(s_h, a_h, s_{h+1}) = (s, a, s')\}$ occurs for at most $6H^2 \iota$ times, so the event $\mathcal{B}_h = \bigcup_{(s,a,s')} (\mathcal{B}_h \cap \{(s_h, a_h, s_{h+1}) = (s, a, s')\})$ occurs for at most $6S^2 AH^2 \iota$ times in total, which leads to contradiction.

As a result, we have $\forall h \in [H]$, with probability $1 - \frac{\delta}{HAK}$, $\sup_{\pi \in \phi} \mathbb{P}_\pi[\mathcal{B}_h] \leq \frac{168S^3 A^2 H^3 \iota}{T}$. Combining these H results, because $\mathbb{P}_\pi[\mathcal{B}] = \sum_{h=1}^H \mathbb{P}_\pi[\mathcal{B}_h]$, we have with probability $1 - \frac{\delta}{AK}$,

$$\sup_{\pi \in \phi} \mathbb{P}_\pi[\mathcal{B}] \leq \frac{168S^3 A^2 H^4 \iota}{T}.$$

□

The following Lemma E.10 is an improved version of the previous Lemma E.9 under the special case where $\phi = \phi^1$. When the policy set contains all the deterministic policies, we can have a bound with tighter dependence on S .

Lemma E.10. *Conditioned on \mathcal{W} in Lemma E.2, if $\phi = \phi^1 = \{\text{the set of all deterministic policies}\}$, with probability $1 - \frac{2S\delta}{K}$, $\sup_{\pi \in \phi^1} \mathbb{P}_\pi[\mathcal{B}] \leq \frac{672S^3 AH^4 \iota}{T}$.*

Proof of Lemma E.10. We will prove that $\forall h \in [H]$, with probability $1 - \frac{2S\delta}{HK}$, $\sup_{\pi \in \phi^1} \mathbb{P}_\pi[\mathcal{B}_h] \leq \frac{672S^3 AH^3 \iota}{T}$. First, same to (4) and (5) we have

$$\begin{aligned}
 \mathbb{P}_\pi[\mathcal{B}_h | P] &= \mathbb{P}_\pi[\mathcal{B}_h | \tilde{P}], \\
 V^{\pi_{h,s,a}}(1_{h,s,a}, \tilde{P}) &\geq \frac{1}{12} \sup_{\pi \in \phi^1} V^\pi(1_{h,s,a}, \tilde{P}).
 \end{aligned}$$

Same to the proof of Lemma E.9, we define π_h to be a policy that chooses each $\pi_{h,s,a}$ with probability $\frac{1}{SA}$ for any $(s, a) \in \mathcal{S} \times \mathcal{A}$. Then we have

$$V^{\pi_h}(1_{h,s,a}, \tilde{P}) \geq \frac{1}{12SA} \sup_{\pi \in \phi^1} V^\pi(1_{h,s,a}, \tilde{P}) = \frac{1}{12SA} \sup_{\pi \in \phi^1} V^\pi(1_{h,s}, \tilde{P}).$$

The last equation is because ϕ^1 consists of all the deterministic policies, for the optimal policy π to visit (h, s) , we can just let $\pi_h(s) = a$ to construct a policy that can visit (h, s, a) with the same probability. Similar to the proof of Lemma E.9, we can assume that running each $\pi_{h,s,a}$ for $\frac{T}{HSA}$ episodes is equivalent to running π_h for $\frac{T}{H}$ episodes.

Because of Lemma E.6, we have $V^{\pi_h}(1_{h,s,a}, P) \geq V^{\pi_h}(1_{h,s,a}, \tilde{P})$. Also, there are $\frac{T}{H}$ episodes used to explore the h -th layer of the MDP, by Lemma D.4 and a union bound, we have with probability $1 - \frac{S\delta}{HK}$, for any $(s, a) \in \mathcal{S} \times \mathcal{A}$,

$$N_h(s, a) \geq \frac{T}{H} \times \frac{V^{\pi_h}(1_{h,s,a}, P)}{2} - \iota \geq \frac{T}{H} \times \frac{V^{\pi_h}(1_{h,s,a}, \tilde{P})}{2} - \iota \geq \frac{T \cdot \sup_{\pi \in \phi^1} V^\pi(1_{h,s}, \tilde{P})}{24HSA} - \iota.$$

For fixed $(s, a) \in \mathcal{S} \times \mathcal{A}$, if $\frac{T \cdot \sup_{\pi \in \phi^1} V^\pi(1_{h,s}, \tilde{P})}{24HSA} \leq 2\iota$, we have that

$$\sup_{\pi \in \phi^1} V^\pi(1_{h,s}, \tilde{P}) \leq \frac{48HSA\iota}{T}.$$

Otherwise, $N_h(s, a) \geq \frac{T \cdot \sup_{\pi \in \phi^1} V^\pi(1_{h,s}, \tilde{P})}{48HSA}$.

By Lemma D.4 and a union bound, we have with probability $1 - \frac{S\delta}{HK}$, for any $(s, a) \in \mathcal{S} \times \mathcal{A}$,

$$6SH^2\iota \geq \frac{N_h(s, a)\mathbb{P}[\mathcal{F}|(h, s, a)]}{2} - \iota,$$

where $\mathbb{P}[\mathcal{F}|(h, s, a)]$ is the conditional probability of entering \mathcal{F} at time step h , given the state-action pair at time step h is (s, a) . This is because similar to the proof of Lemma E.9, the event of entering \mathcal{F} from (h, s, a) happens for at most $6SH^2\iota$ times.

Then we have with probability $1 - \frac{2S\delta}{HK}$, $\sup_{\pi \in \phi^1} V^\pi(1_{h,s}, \tilde{P}) \leq \frac{48HSA\iota}{T}$ or $\mathbb{P}[\mathcal{F}|(h, s, a)] \leq \frac{672H^3S^2A\iota}{T \cdot \sup_{\pi \in \phi^1} V^\pi(1_{h,s}, \tilde{P})}$ holds for any $(s, a) \in \mathcal{S} \times \mathcal{A}$. Therefore, it holds that

$$\begin{aligned} \sup_{\pi \in \phi^1} \mathbb{P}_\pi[\mathcal{B}_h|P] &= \sup_{\pi \in \phi^1} \mathbb{P}_\pi[\mathcal{B}_h|\tilde{P}] \\ &= \sup_{\pi \in \phi^1} \sum_{s \in \mathcal{S}} V^\pi(1_{h,s}, \tilde{P}) \max_{a \in \mathcal{A}} \tilde{P}_h(s^\dagger|s, a) \\ &= \sup_{\pi \in \phi^1} \sum_{s \in \mathcal{S}} V^\pi(1_{h,s}, \tilde{P}) \max_{a \in \mathcal{A}} \mathbb{P}[\mathcal{F}|(h, s, a)] \\ &\leq \sup_{\pi \in \phi^1} \sum_{s \in \mathcal{S}} \max\left\{\frac{48HSA\iota}{T}, \frac{672H^3S^2A\iota}{T}\right\} \\ &= \frac{672H^3S^3A\iota}{T}. \end{aligned}$$

The inequality is because if $\frac{T \cdot \sup_{\pi \in \phi^1} V^\pi(1_{h,s}, \tilde{P})}{24HSA} \leq 2\iota$,

$$V^\pi(1_{h,s}, \tilde{P}) \max_{a \in \mathcal{A}} \mathbb{P}[\mathcal{F}|(h, s, a)] \leq V^\pi(1_{h,s}, \tilde{P}) \leq \sup_{\pi \in \phi^1} V^\pi(1_{h,s}, \tilde{P}) \leq \frac{48HSA\iota}{T}.$$

Otherwise we have that

$$V^\pi(1_{h,s}, \tilde{P}) \max_{a \in \mathcal{A}} \mathbb{P}[\mathcal{F}|(h, s, a)] \leq V^\pi(1_{h,s}, \tilde{P}) \frac{672H^3S^2A\iota}{T \cdot \sup_{\pi \in \phi^1} V^\pi(1_{h,s}, \tilde{P})} \leq \frac{672H^3S^2A\iota}{T}.$$

As a result, we have $\forall h \in [H]$, with probability $1 - \frac{2S\delta}{HK}$, $\sup_{\pi \in \phi^1} \mathbb{P}_\pi[\mathcal{B}_h] \leq \frac{672S^3AH^3\iota}{T}$. Combining these H results, because $\mathbb{P}_\pi[\mathcal{B}] = \sum_{h=1}^H \mathbb{P}_\pi[\mathcal{B}_h]$, we have with probability $1 - \frac{2S\delta}{K}$,

$$\sup_{\pi \in \phi^1} \mathbb{P}_\pi[\mathcal{B}] \leq \frac{672S^3AH^4\iota}{T}.$$

□

Remark E.11. We can see that with the same algorithm, the analysis is different for the general case and the special case that ϕ contains all the deterministic policies. The main technical reason is when $\phi = \phi^1$, $\sup_{\pi \in \phi^1} V^\pi(1_{h,s,a}, \tilde{P}) = \sup_{\pi \in \phi^1} V^\pi(1_{h,s}, \tilde{P})$ while this does not hold for general policy set ϕ . We will show that the part of the regret due to the construction of \tilde{P} is a lower order term. We prove a better bound for the case $\phi = \phi^1$ mainly for a better sample complexity in reward-free setting.

Lemma E.12 (Restate Lemma 6.3). *Conditioned on \mathcal{W} , under the high-probability event in Lemma E.9 and Lemma E.10, we have for any policy $\pi \in \phi$ and reward function r' ,*

$$0 \leq V^\pi(r', P) - V^\pi(r', \tilde{P}) \leq \frac{168S^3 A^2 H^5 \iota}{T}.$$

In addition, if $\phi = \phi^1 = \{\text{the set of all deterministic policies}\}$, we have that for any policy $\pi \in \phi^1$ and reward function r' ,

$$0 \leq V^\pi(r', P) - V^\pi(r', \tilde{P}) \leq \frac{672S^3 A H^5 \iota}{T}.$$

Proof of Lemma E.12. We will only prove the first part, the second part is almost the same. Because \tilde{P} is the absorbing version of P , the left hand side is obvious. For the right hand side, we have that if $\pi \in \phi$,

$$\begin{aligned} V^\pi(r', P) &= \sum_{\tau \in \mathcal{B}^c} r'(\tau) P_\pi(\tau) + \sum_{\tau \in \mathcal{B}} r'(\tau) P_\pi(\tau) \\ &= \sum_{\tau \in \mathcal{B}^c} r'(\tau) \tilde{P}_\pi(\tau) + \sum_{\tau \in \mathcal{B}} r'(\tau) P_\pi(\tau) \\ &\leq V^\pi(r', \tilde{P}) + \sum_{\tau \in \mathcal{B}} r'(\tau) P_\pi(\tau) \\ &\leq V^\pi(r', \tilde{P}) + \sum_{\tau \in \mathcal{B}} H P_\pi(\tau) \\ &\leq V^\pi(r', \tilde{P}) + H P_\pi(\mathcal{B}) \\ &\leq V^\pi(r', \tilde{P}) + \frac{168S^3 A^2 H^5 \iota}{T}. \end{aligned}$$

Note that all the \mathcal{B} here are defined under original MDP. The second equation is because $\tilde{P} = P$ when $\tau \in \mathcal{B}^c$. The first inequality is due to non-negative reward in \mathcal{B} under \tilde{P} . The last inequality follows from Lemma E.9. \square

F. Proof of lemmas regarding Fine Exploration (Algorithm 3)

We first state a conclusion about the global switching cost of Algorithm 3.

Lemma F.1. *The global switching cost of Algorithm 3 is bounded by HSA.*

Proof of Lemma F.1. There are at most HSA different $\pi_{h,s,a}$'s, Algorithm 3 will just run each policy for several times. \square

Lemma F.2 (Simulation lemma (Dann et al., 2017)). *For any two MDPs M' and M'' with rewards r' and r'' and transition probabilities \mathcal{P}' and \mathcal{P}'' , the difference in values V' , V'' with respect to the same policy π can be written as*

$$V'_h(s) - V''_h(s) = \mathbb{E}_{M'', \pi} \left[\sum_{i=h}^H [r'_i(s_i, a_i) - r''_i(s_i, a_i) + (\mathbb{P}'_i - \mathbb{P}''_i) V'_{i+1}(s_i, a_i)] \mid s_h = s \right].$$

Now we can prove that value functions under \tilde{P} and \hat{P} are close to each other.

Lemma F.3 (Restate Lemma 6.4). *Conditioned on the fact that P^{int} is $\frac{1}{H}$ -multiplicatively accurate to \tilde{P} (the case in Lemma E.3), with probability $1 - \frac{T\delta}{2K}$, for any policy $\pi \in \phi$ and reward function r' ,*

$$|V^\pi(r', \hat{P}) - V^\pi(r', \tilde{P})| = O\left(\sqrt{\frac{H^5 S^2 A \iota}{T}}\right).$$

Proof of Lemma F.3. In this part of proof, note that the reward function r' is defined under the original MDP. When we transfer r' to be a reward function under the absorbing MDP, $r'_h(s^\dagger, a) = 0$ for any $(h, a) \in [H] \times \mathcal{A}$. Therefore, if \hat{V}

denotes the value function of some policy π under the MDP with reward function r' and transition kernel \hat{P} , we have $\hat{V}_h(s^\dagger) = 0$, for any $h \in [H]$. Because of simulation lemma (Lemma F.2), we have that $\forall \pi \in \phi$,

$$|V^\pi(r', \hat{P}) - V^\pi(r', \tilde{P})| \leq \mathbb{E}_{\tilde{P}}^\pi \sum_{h=1}^H |(\hat{P}_h - \tilde{P}_h) \cdot \hat{V}_{h+1}^\pi|, \quad (7)$$

where \hat{V}_h^π is the value function of policy π under \hat{P} and r' at time h . The proof holds simultaneously for all reward function r' , so we will omit r' for simplicity.

Then we have

$$\begin{aligned} \mathbb{E}_{\tilde{P}}^\pi |(\hat{P}_h - \tilde{P}_h) \cdot \hat{V}_{h+1}^\pi| &= \sum_{a, s \in \mathcal{S}} |(\hat{\mathbb{P}}_h - \tilde{\mathbb{P}}_h) \hat{V}_{h+1}^\pi(s, a)| V^\pi(\mathbf{1}_{h, s, a}, \tilde{P}) \\ &\leq \sqrt{\sum_{a, s \in \mathcal{S}} |(\hat{\mathbb{P}}_h - \tilde{\mathbb{P}}_h) \hat{V}_{h+1}^\pi(s, a)|^2 V^\pi(\mathbf{1}_{h, s, a}, \tilde{P})} \\ &= \sqrt{\sum_{a, s \in \mathcal{S}} |(\hat{\mathbb{P}}_h - \tilde{\mathbb{P}}_h) \hat{V}_{h+1}^\pi(s, a)|^2 V^\pi(\mathbf{1}_{h, s, a}, \tilde{P}) \mathbb{1}\{a = \pi_h(s)\}}. \end{aligned}$$

The first equation is because if the trajectory arrives at the absorbing state s^\dagger at time step h , then $\hat{P}_h(s'|s^\dagger, a) = \tilde{P}_h(s'|s^\dagger, a)$ for any a, s' . The first inequality is because of Cauchy-Schwarz inequality. The last equation is because for any $\pi \in \phi$, π is deterministic.

Define π_{random} to be a policy that chooses each $\pi_{h, s, a}$ with probability $\frac{1}{HSA}$ for any $(h, s, a) \in [H] \times \mathcal{S} \times \mathcal{A}$. Define $\mu_h(s, a)$ to be $\mu_h(s, a) = V^{\pi_{random}}(\mathbf{1}_{h, s, a}, \tilde{P})$. Then similar to (5) and (6), we have for any $(h, s, a) \in [H] \times \mathcal{S} \times \mathcal{A}$,

$$\sup_{\pi \in \phi} \frac{V^\pi(\mathbf{1}_{h, s, a}, \tilde{P})}{\mu_h(s, a)} \leq 12HSA. \quad (8)$$

Plugging in this result into the previous inequality, we have

$$\begin{aligned} \mathbb{E}_{\tilde{P}}^\pi |(\hat{P}_h - \tilde{P}_h) \cdot \hat{V}_{h+1}^\pi| &\leq \sqrt{\sum_{a, s \in \mathcal{S}} |(\hat{\mathbb{P}}_h - \tilde{\mathbb{P}}_h) \hat{V}_{h+1}^\pi(s, a)|^2 V^\pi(\mathbf{1}_{h, s, a}, \tilde{P}) \mathbb{1}\{a = \pi_h(s)\}} \\ &\leq \sqrt{\sum_{a, s \in \mathcal{S}} |(\hat{\mathbb{P}}_h - \tilde{\mathbb{P}}_h) \hat{V}_{h+1}^\pi(s, a)|^2 \cdot 12HSA \mu_h(s, a) \cdot \mathbb{1}\{a = \pi_h(s)\}} \\ &\leq \sqrt{12HSA \cdot \sup_{\nu: \mathcal{S} \rightarrow \mathcal{A}} \sum_{a, s \in \mathcal{S}} |(\hat{\mathbb{P}}_h - \tilde{\mathbb{P}}_h) \hat{V}_{h+1}^\pi(s, a)|^2 \mu_h(s, a) \cdot \mathbb{1}\{a = \nu(s)\}} \\ &\leq \sqrt{12HSA \cdot \sup_{\nu: \mathcal{S} \rightarrow \mathcal{A}} \sum_{a, s \in \mathcal{S}} |(\hat{\mathbb{P}}_h - \tilde{\mathbb{P}}_h) \hat{V}_{h+1}^\pi(s, a)|^2 \mu'_h(s, a) \cdot \mathbb{1}\{a = \nu(s)\}} \\ &= \sqrt{12HSA \cdot \sup_{\nu: \mathcal{S} \rightarrow \mathcal{A}} \mathbb{E}_{\mu'_h} |(\hat{\mathbb{P}}_h - \tilde{\mathbb{P}}_h) \hat{V}_{h+1}^\pi(s, a)|^2 \cdot \mathbb{1}\{a = \nu(s)\}} \\ &\leq \sqrt{12HSA \cdot \sup_{G: \mathcal{S} \cup s^\dagger \rightarrow [0, H]} \sup_{\nu: \mathcal{S} \cup s^\dagger \rightarrow \mathcal{A}} \mathbb{E}_{\mu'_h} |(\hat{\mathbb{P}}_h - \tilde{\mathbb{P}}_h) G(s, a)|^2 \cdot \mathbb{1}\{a = \nu(s)\}}, \end{aligned}$$

where $\mu'_h(s, a) = V^{\pi_{random}}(\mathbf{1}_{h, s, a}, P)$ is the distribution of the data. The fourth inequality is because $\mu'_h(s, a) \geq \mu_h(s, a)$ for any $(h, s, a) \in [H] \times \mathcal{S} \times \mathcal{A}$ (because of Lemma E.6). In the equation, we extend the definition of μ' by letting $\mu'(s^\dagger, a) = 0$ so that μ' is a distribution on $\mathcal{S} \cup s^\dagger \times \mathcal{A}$. The last inequality is because \hat{V}_{h+1}^π is a function from $\mathcal{S} \cup s^\dagger$ to $[0, H]$.

Note that our data follows the distribution μ' . In addition, from the definition of \tilde{P} and \hat{P} , we have that \hat{P} is the empirical estimate of \tilde{P} . By Lemma F.4 (which we state right after) we have with probability $1 - \frac{T\delta}{2K}$, for any $h \in [H]$, policy $\pi \in \phi$ and reward function r' ,

$$\mathbb{E}_{\tilde{P}}^\pi |(\hat{P}_h - \tilde{P}_h) \cdot \hat{V}_{h+1}^\pi| = O\left(\sqrt{\frac{H^3 S^2 A t}{T}}\right).$$

By (7), we have with probability $1 - \frac{T\delta}{2K}$, for any policy $\pi \in \phi$ and reward function r' ,

$$|V^\pi(r', \hat{P}) - V^\pi(r', \tilde{P})| = O\left(\sqrt{\frac{H^5 S^2 A \epsilon}{T}}\right).$$

□

Lemma F.4 (Lemma C.2 in (Jin et al., 2020a)). *Suppose $\hat{\mathbb{P}}$ is the empirical transition matrix formed by sampling according to μ distribution for N samples, G can be any function from \mathcal{S} to $[0, H]$, ν can be any function from \mathcal{S} to \mathcal{A} , then with probability at least $1 - \delta$, we have for any $h \in [H]$:*

$$\max_{G: \mathcal{S} \rightarrow [0, H]} \max_{\nu: \mathcal{S} \rightarrow \mathcal{A}} \mathbb{E}_{\mu_h} |(\hat{\mathbb{P}}_h - \mathbb{P}_h)G(s, a)|^2 \mathbb{1}\{a = \nu(s)\} \leq O\left(\frac{H^2 S \log(HAN/\delta)}{N}\right).$$

This is a critical lemma that requires delicate arguments to prove. For self-containedness, we include a full proof with more technical details.

Proof of Lemma F.4. Define random variables

$$\begin{aligned} X &= (\hat{\mathbb{P}}_h G(s, a) - G(s'))^2 - (\mathbb{P}_h G(s, a) - G(s'))^2, \\ \bar{X} &= (f(s, a) - G'(s'))^2 - (g(s, a) - G'(s'))^2, \\ Y &= X \mathbb{1}\{a = \nu(s)\}, \bar{Y} = \bar{X} \mathbb{1}\{a = \nu'(s)\}, \end{aligned}$$

where $(s, a, s') \sim \mu_h \times \mathbb{P}_h(\cdot | s, a)$. Here, ν' can be any function from \mathcal{S} to \mathcal{A} (in equation (9)) and will be the same as ν in equation (10), (11), (12). Also, f, g and G' can be any function (in equation (9)) from the ϵ -cover defined later and will be the closest function from the ϵ -cover to $\hat{\mathbb{P}}G(s, a) \mathbb{1}\{a = \nu(s)\}$, $\mathbb{P}G(s, a) \mathbb{1}\{a = \nu(s)\}$ and G in equation (10), (11), (12). Also, we define

$$\begin{aligned} X_i &= (\hat{\mathbb{P}}_h G(s_i, a_i) - G(s'_i))^2 - (\mathbb{P}_h G(s_i, a_i) - G(s'_i))^2, \\ \bar{X}_i &= (f(s_i, a_i) - G'(s'_i))^2 - (g(s_i, a_i) - G'(s'_i))^2, \\ Y_i &= X_i \mathbb{1}\{a_i = \nu(s_i)\}, \bar{Y}_i = \bar{X}_i \mathbb{1}\{a_i = \nu'(s_i)\}, \end{aligned}$$

where (s_i, a_i, s'_i) is the i -th sample in time step h we collect. Notice that for every tuple (v', f, g, G') and \bar{Y}, \bar{Y}_i related to this tuple, we have that \bar{Y}_i 's are i.i.d samples from the distribution of \bar{Y} .

Same to the proof in (Jin et al., 2020a), we have these three properties of Y and Y_i .

$$(1) \mathbb{E}Y = \mathbb{E}_{\mu_h} |(\hat{\mathbb{P}}_h - \mathbb{P}_h)G(s, a)|^2 \mathbb{1}\{a = \nu(s)\}.$$

$$(2) \sum_{i=1}^N Y_i \leq 0.$$

$$(3) \text{Var}\{Y\} \leq 4H^2 \mathbb{E}(Y).$$

Since we are taking maximum over ν and $G(s)$, and $\hat{\mathbb{P}}$ is random, we need to cover all the possible ν and all the possible values of $\hat{\mathbb{P}}G(s, a) \mathbb{1}\{a = \nu(s)\}$, $\mathbb{P}G(s, a) \mathbb{1}\{a = \nu(s)\}$ and G to ϵ accuracy to use Bernstein's inequality for the functions in the cover. For ν , there are A^S deterministic policies in total. Given a fixed ν , $\hat{\mathbb{P}}G(s, a) \mathbb{1}\{a = \nu(s)\}$, $\mathbb{P}G(s, a) \mathbb{1}\{a = \nu(s)\}$ and G can be covered by $(H/\epsilon)^{3S}$ values because the first two functions can be covered by $(H/\epsilon)^{2S}$ values (for $a \neq \nu(s)$ the first two will be 0) and G itself can be covered by $(H/\epsilon)^S$ values.

By Bernstein's inequality (Lemma D.1) and a union bound, we have with probability $1 - \delta$, for any (ν', f, g, G') in the ϵ -cover, it holds that

$$\mathbb{E}(\bar{Y}) - \frac{1}{N} \sum_{i=1}^N \bar{Y}_i \leq \sqrt{\frac{2\text{Var}\{\bar{Y}\} \log\left(\left(\frac{H}{\epsilon}\right)^{3S} \cdot A^S \cdot \frac{H}{\delta}\right)}{N}} + \frac{H^2 \log\left(\left(\frac{H}{\epsilon}\right)^{3S} \cdot A^S \cdot \frac{H}{\delta}\right)}{3N}. \quad (9)$$

Under this high probability case, for any G, ν and $\widehat{\mathbb{P}}$, choose $\nu' = \nu$ and (f, g, G') to be the closest function from the ϵ -cover to $\widehat{\mathbb{P}}G(s, a)\mathbb{1}\{a = \nu(s)\}$, $\mathbb{P}G(s, a)\mathbb{1}\{a = \nu(s)\}$ and G . Let \bar{Y} be defined according to this (ν, f, g, G') . Then we have

$$|Y - \bar{Y}| \leq 4H\epsilon, \quad (10)$$

for any possible (s, a, s') (hence this inequality is also true when adding expectation to both terms) and

$$\begin{aligned} |\text{Var}\{Y\} - \text{Var}\{\bar{Y}\}| &\leq |\mathbb{E}\{\bar{Y}^2\} - \mathbb{E}\{Y^2\}| + |(\mathbb{E}\bar{Y})^2 - (\mathbb{E}Y)^2| \\ &\leq 2H^2 \cdot 4H\epsilon + 2H^2 \cdot 4H\epsilon \\ &= 16H^3\epsilon. \end{aligned} \quad (11)$$

As a result, we have

$$\begin{aligned} &\mathbb{E}_{\mu_h} |(\widehat{\mathbb{P}}_h - \mathbb{P}_h)G(s, a)|^2 \mathbb{1}\{a = \nu(s)\} = \mathbb{E}Y \\ &\leq \mathbb{E}Y - \frac{1}{N} \sum_{i=1}^N Y_i \\ &\leq \mathbb{E}\bar{Y} - \frac{1}{N} \sum_{i=1}^N \bar{Y}_i + 8H\epsilon \\ &\leq \sqrt{\frac{2\text{Var}\{\bar{Y}\} \log\left(\left(\frac{H}{\epsilon}\right)^{3S} \cdot A^S \cdot \frac{H}{\delta}\right)}{N}} + \frac{H^2 \log\left(\left(\frac{H}{\epsilon}\right)^{3S} \cdot A^S \cdot \frac{H}{\delta}\right)}{3N} + 8H\epsilon \\ &\leq \sqrt{\frac{2\text{Var}\{Y\} \log\left(\left(\frac{H}{\epsilon}\right)^{3S} \cdot A^S \cdot \frac{H}{\delta}\right)}{N}} + \frac{H^2 \log\left(\left(\frac{H}{\epsilon}\right)^{3S} \cdot A^S \cdot \frac{H}{\delta}\right)}{3N} \\ &\quad + 8H\epsilon + \sqrt{\frac{2 \cdot 16H^3\epsilon \log\left(\left(\frac{H}{\epsilon}\right)^{3S} \cdot A^S \cdot \frac{H}{\delta}\right)}{N}}. \end{aligned} \quad (12)$$

The first inequality is because of property 2. The second inequality is because of (10). The third inequality is because of (9). The last inequality is because of (11) and $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$.

We can simply choose $\epsilon = \frac{HS}{32N}$ and plug in property (3) of Y , then by solving the following quadratic inequality we can finish the proof.

$$\begin{aligned} &\mathbb{E}_{\mu_h} |(\widehat{\mathbb{P}}_h - \mathbb{P}_h)G(s, a)|^2 \mathbb{1}\{a = \nu(s)\} \\ &\leq \sqrt{\frac{8H^2 \mathbb{E}_{\mu_h} |(\widehat{\mathbb{P}}_h - \mathbb{P}_h)G(s, a)|^2 \mathbb{1}\{a = \nu(s)\} \cdot 3S \log\left(\frac{32HAN}{\delta}\right)}{N}} + \frac{2H^2 S \log\left(\frac{32HAN}{\delta}\right)}{N}. \end{aligned} \quad (13)$$

□

G. Proof of Theorem 4.1

We first give a proof for the upper bound on the number of stages.

Lemma G.1. *If $T^{(k)} = K^{1 - \frac{1}{2^k}}$ for $k = 1, 2, \dots$, we have*

$$K_0 = \min\left\{j : 2 \sum_{k=1}^j T^{(k)} \geq K\right\} = O(\log \log K).$$

Proof of Lemma G.1. Take $j = \log_2 \log_2 K$, we have $2T^{(j)} = 2 \frac{K}{K^{(\log_2 K)^{-1}}} = 2 \frac{K}{2} = K$, which means that

$$K_0 \leq \log_2 \log_2 K + 1 = O(\log \log K).$$

□

Then we are able to bound the total global switching cost of Algorithm 1.

Lemma G.2. *The total global switching cost of Algorithm 1 is bounded by $O(HSA \log \log T)$.*

Proof of Lemma G.2. For each stage, the global switching cost is bounded by $2HSA$ because of Lemma E.1 and Lemma F.1. There are $O(\log \log T)$ stages, so the total global switching cost is at most $O(HSA \log \log T)$. \square

Recall that in Algorithm 1, in each stage k ($1 \leq k \leq K_0$), we run Algorithm 2 to construct the infrequent tuples \mathcal{F}^k and the intermediate transition kernel $P^{int,k}$. The absorbing MDP \tilde{P}^k is constructed as in Definition 3.1 based on \mathcal{F}^k and the real MDP P . Then we run Algorithm 3 to construct an empirical estimate of \tilde{P}^k , which is \hat{P}^k .

Lemma G.3 (Restate Lemma 6.1). *There exists a constant C , such that with probability $1 - \delta$, it holds that for any k and $\pi \in \phi^k$,*

$$|V^\pi(r, \hat{P}^k) - V^\pi(r, P)| \leq C \left(\sqrt{\frac{H^5 S^2 A \iota}{T^{(k)}}} + \frac{S^3 A^2 H^5 \iota}{T^{(k)}} \right),$$

where $\iota = \log(2HAK/\delta)$.

Proof of Lemma G.3. For the choice of the universal constant C , we can first let C be the constant hidden by the big O in Lemma F.3. Then because of Lemma E.12, we can choose $C = \max\{C, 168\}$. Note that this C is also used as the universal constant in the elimination step in Algorithm 1.

Because of triangular inequality, we have for any k and any $\pi \in \phi^k$,

$$|V^\pi(r, \hat{P}^k) - V^\pi(r, P)| \leq |V^\pi(r, \hat{P}^k) - V^\pi(r, \tilde{P}^k)| + |V^\pi(r, \tilde{P}^k) - V^\pi(r, P)|. \quad (14)$$

For any $k \leq K_0$, because of Lemma E.3 and Lemma E.12, we have with probability $1 - (\frac{S^2 \delta}{K} + \frac{\delta}{AK})$, it holds that for any $\pi \in \phi^k$, $|V^\pi(r, P) - V^\pi(r, \tilde{P}^k)| \leq C \frac{S^3 A^2 H^5}{T^{(k)}}$ while $P^{int,k}$ is $\frac{1}{H}$ -multiplicatively accurate to \tilde{P}^k . Conditioned on this case, because of Lemma F.3, we have with probability $1 - \frac{T^{(k)} \delta}{2K}$, for any $\pi \in \phi^k$, $|V^\pi(r, \hat{P}^k) - V^\pi(r, \tilde{P}^k)| \leq C \sqrt{\frac{H^5 S^2 A \iota}{T^{(k)}}}$.

Combining these two results, we have for any $1 \leq k \leq K_0$, with probability $1 - (\frac{S^2 \delta}{K} + \frac{\delta}{AK} + \frac{T^{(k)} \delta}{2K})$, for any $\pi \in \phi^k$,

$$|V^\pi(r, \hat{P}^k) - V^\pi(r, P)| \leq C \left(\sqrt{\frac{H^5 S^2 A \iota}{T^{(k)}}} + \frac{S^3 A^2 H^5 \iota}{T^{(k)}} \right).$$

Finally, the proof is completed through a union bound on k and the fact that the failure probability is bounded by

$$\left(\frac{S^2 \delta}{K} + \frac{\delta}{AK} \right) \times O(\log \log K) + \sum_{k=1}^{K_0} \frac{T^{(k)} \delta}{2K} \leq \delta,$$

if $K \geq \tilde{\Omega}(S^2)$. \square

Lemma G.4. *Conditioned on the same high probability event of Lemma G.3, the optimal policy π^* will never be eliminated, i.e., $\pi^* \in \phi^k$ for $k = 1, 2, 3, \dots$.*

Proof of Lemma G.4. We will prove this lemma by induction. First, because ϕ^1 contains all the deterministic policies, $\pi^* \in \phi^1$. Assume $\pi^* \in \phi^k$, then we have

$$\begin{aligned} \sup_{\hat{\pi} \in \phi^k} V^{\hat{\pi}}(r, \hat{P}^k) - V^{\pi^*}(r, \hat{P}^k) &= V^{\hat{\pi}^k}(r, \hat{P}^k) - V^{\pi^*}(r, \hat{P}^k) \\ &\leq |V^{\hat{\pi}^k}(r, \hat{P}^k) - V^{\hat{\pi}^k}(r, P)| + V^{\hat{\pi}^k}(r, P) - V^{\pi^*}(r, P) + |V^{\pi^*}(r, P) - V^{\pi^*}(r, \hat{P}^k)| \\ &\leq 2C \left(\sqrt{\frac{H^5 S^2 A \iota}{T^{(k)}}} + \frac{S^3 A^2 H^5 \iota}{T^{(k)}} \right). \end{aligned}$$

The last inequality is because of Lemma G.3 and π^* is the optimal policy.

Then according to the elimination rule in Algorithm 1, we have that $\pi^* \in \phi^{k+1}$, which means the optimal policy π^* will never be eliminated. \square

Lemma G.5. *Conditioned on the same high probability event of Lemma G.3, for any remaining policies, i.e., $\pi \in \phi^{k+1}$, we have that*

$$V^{\pi^*}(r, P) - V^\pi(r, P) \leq 4C\left(\sqrt{\frac{H^5 S^2 A \ell}{T^{(k)}}} + \frac{S^3 A^2 H^5 \ell}{T^{(k)}}\right).$$

Proof of Lemma G.5. For $\pi \in \phi^{k+1}$, because the optimal policy π^* will never be eliminated (Lemma G.4), we have that

$$V^{\pi^*}(r, \hat{P}^k) - V^\pi(r, \hat{P}^k) \leq \sup_{\hat{\pi} \in \phi^k} V^{\hat{\pi}}(r, \hat{P}^k) - V^\pi(r, \hat{P}^k) \leq 2C\left(\sqrt{\frac{H^5 S^2 A \ell}{T^{(k)}}} + \frac{S^3 A^2 H^5 \ell}{T^{(k)}}\right). \quad (15)$$

Then we have

$$\begin{aligned} V^{\pi^*}(r, P) - V^\pi(r, P) &\leq |V^{\pi^*}(r, P) - V^{\pi^*}(r, \hat{P}^k)| + V^{\pi^*}(r, \hat{P}^k) - V^\pi(r, \hat{P}^k) + |V^\pi(r, \hat{P}^k) - V^\pi(r, P)| \\ &\leq 4C\left(\sqrt{\frac{H^5 S^2 A \ell}{T^{(k)}}} + \frac{S^3 A^2 H^5 \ell}{T^{(k)}}\right). \end{aligned}$$

The last inequality is because of Lemma G.3 and (15). \square

Lemma G.6. *Conditioned on the same high probability event of Lemma G.3, if $K \geq \tilde{\Omega}(S^8 A^6 H^{10})$, the total regret is at most $\tilde{O}(\sqrt{H^4 S^2 A T})$.*

Proof of Lemma G.6. The regret for the first stage (stage 1) is at most $2HT^{(1)} = O(HK^{\frac{1}{2}})$.

For stage $k \geq 2$, because of Lemma G.5, the policies we use (any $\pi \in \phi^k$) are at most $4C\left(\sqrt{\frac{H^5 S^2 A \ell}{T^{(k-1)}}} + \frac{S^3 A^2 H^5 \ell}{T^{(k-1)}}\right)$

sub-optimal, so the regret for the k -th stage ($2T^{(k)}$ episodes) is at most $8CT^{(k)}\left(\sqrt{\frac{H^5 S^2 A \ell}{T^{(k-1)}}} + \frac{S^3 A^2 H^5 \ell}{T^{(k-1)}}\right)$.

Adding up the regret for each stage, we have that the total regret is bounded by

$$\begin{aligned} \text{Regret}(T) &\leq 2HK^{\frac{1}{2}} + \sum_{k=2}^{K_0} 8CT^{(k)}\left(\sqrt{\frac{H^5 S^2 A \ell}{T^{(k-1)}}} + \frac{S^3 A^2 H^5 \ell}{T^{(k-1)}}\right) \\ &= O(HK^{\frac{1}{2}}) + O(\sqrt{H^5 S^2 A K \ell} \cdot \log \log K) + O(S^3 A^2 H^5 K^{\frac{1}{4}} \ell) \\ &= O(\sqrt{H^5 S^2 A K \ell} \cdot \log \log K) + O(S^3 A^2 H^5 K^{\frac{1}{4}} \ell) \\ &= \tilde{O}(\sqrt{H^4 S^2 A T}), \end{aligned}$$

where the last equality is because $K \geq \tilde{\Omega}(S^8 A^6 H^{10})$. \square

Then Theorem 4.1 holds because of Lemma G.2, Lemma G.3 and Lemma G.6.

Corollary G.7 (Transition to a PAC bound). *Under the same assumption as Theorem 4.1, for any $\epsilon > 0$, Algorithm 1 can output a stochastic policy $\hat{\pi}$ such that with high probability,*

$$V_1^*(s_1) - V_1^{\hat{\pi}}(s_1) \leq \epsilon$$

after $K = \tilde{O}(\frac{H^5 S^2 A}{\epsilon^2})$ episodes.

Proof. By Theorem 4.1, we have that the regret is bounded by $\tilde{O}(\sqrt{H^4 S^2 A T})$ with high probability, which means we have

$$\sum_{k=1}^K V_1^*(s_1) - V_1^{\pi_k}(s_1) \leq \tilde{O}(\sqrt{H^4 S^2 A T}),$$

where π_k is the policy used in episode k . Now define a stochastic policy $\hat{\pi}$ as

$$\hat{\pi} = \frac{1}{K} \sum_{k=1}^K \pi_k.$$

Then we have

$$\mathbb{E}[V_1^*(s_1) - V_1^{\hat{\pi}}(s_1)] = \frac{1}{K} \sum_{k=1}^K V_1^*(s_1) - V_1^{\pi_k}(s_1) \leq \tilde{O}\left(\sqrt{\frac{H^5 S^2 A}{K}}\right).$$

By Markov inequality, we have with high probability that

$$V_1^*(s_1) - V_1^{\hat{\pi}}(s_1) \leq \tilde{O}\left(\sqrt{\frac{H^5 S^2 A}{K}}\right).$$

Taking $K = \tilde{O}\left(\frac{H^5 S^2 A}{\epsilon^2}\right)$ bounds the above by ϵ . \square

Remark G.8. In addition to constructing a random policy, we can output any policy in the remaining policy set ϕ^{K_0+1} . Because there are $K_0 = O(\log \log K)$ stages in total, the maximal $T^{(k)}$ is larger than $\Omega\left(\frac{K}{\log \log K}\right)$. According to Lemma G.5, for any $\pi \in \phi^{K_0+1}$,

$$V^{\pi^*}(r, P) - V^\pi(r, P) \leq 4C\left(\sqrt{\frac{H^5 S^2 A t}{T^{(k)}}} + \frac{S^3 A^2 H^5 t}{T^{(k)}}\right), \forall k = 1, 2, \dots, K_0.$$

Combining these two results, we have for any $\pi \in \phi^{K_0+1}$,

$$V^{\pi^*}(r, P) - V^\pi(r, P) \leq \tilde{O}\left(\sqrt{\frac{H^5 S^2 A}{K}}\right).$$

Then $K = \tilde{O}\left(\frac{H^5 S^2 A}{\epsilon^2}\right)$ bounds the above by ϵ .

H. Proof of lower bounds (Theorem 4.2 and Theorem 4.3)

Theorem H.1 (Restate Theorem 4.2). *If $S \leq A^{\frac{H}{2}}$, for any algorithm with near-optimal $\tilde{O}^*(\sqrt{T})$ regret bound, the global switching cost is at least $\Omega(HSA \log \log T)$.*

Theorem H.2 (Restate Theorem 4.3). *If $S \leq A^{\frac{H}{2}}$, for any algorithm with sub-linear regret bound, the global switching cost is at least $\Omega(HSA)$.*

First, we will state the high level idea of the proof and some related discussions. For the $\Omega(HSA \log \log T)$ and $\Omega(HSA)$ lower bounds, we construct a MDP to show that it is at least as difficult as multi-armed bandits with $\Omega(HSA)$ arms. Previously, Bai et al. (2019) proved an $\Omega(HSA)$ lower bound for local switching cost, which can only imply an $\Omega(A)$ lower bound for global switching cost in the worst case. In (Huang et al., 2022), the authors proved an $\Omega(dH)$ lower bound for global switching cost under linear setting for all algorithms with PAC bound. The same lower bound is derived in (Gao et al., 2021) for all algorithms with sub-linear regret. However, for the MDPs constructed by both papers, the number of actions available at each state is not the same, which means we can not get an $\Omega(HSA)$ lower bound for global switching cost by directly plugging in $d = SH$. Finally, we state that both lower bounds we present are optimal. An $\Omega(HSA \log \log T)$ lower bound on global switching cost is optimal since this bound is matched by the upper bound of switching cost in Theorem 4.1. Also, an $\Omega(HSA)$ lower bound on global switching cost is the optimal result for any no-regret algorithms. This is because we can run our low adaptive reward-free exploration (Algorithm 4) for $K^{\frac{2}{3}}$ episodes and run the policy $\hat{\pi}^r$ where r is the real reward function for the remaining episodes. It can be shown that the regret is of order $O(T^{\frac{2}{3}})$ and global switching cost is bounded by $2HSA$.

Proof of Theorem H.1 and Theorem H.2. In this part of proof, we will add a mild assumption to parameters H, S, A by assuming that $S \leq A^{\frac{H}{2}}$. First we will show that under this assumption, a MDP with S states, A actions and horizon H can be at least as difficult as a multi-armed bandit with $\Omega(HSA)$ arms. We will consider a MDP with deterministic transition kernel and a fixed initial state s_1 , a special state s^\dagger will be used as absorbing state. The state space \mathcal{S} can be divided into $\mathcal{S} = \{s_1, s_2, \dots, s_{S-1}, s^\dagger\}$. The action space \mathcal{A} can be divided into $\mathcal{A} = \{a_1, a_2, \dots, a_A\}$. Then the construction of MDP can be divided into three parts.

Absorbing state For the absorbing state s^\dagger , for any action a and $h \in [H]$, $P_h(s^\dagger | s^\dagger, a) = 1$ while $P_h(s | s^\dagger, a) = 0$ for any $s \neq s^\dagger$. The reward is defined as $r_h(s^\dagger, a) = 0$ for any $a, h \in [H]$.

The first several layers Let H_0 be the minimal positive integer such that $S \leq A^{H_0}$. By the assumption that $S \leq A^{\frac{H}{2}}$, we have $H_0 \leq \frac{H}{2}$. Then we can use an A -armed tree structure to ensure that for any state $s \neq s^\dagger$, there exists a unique path starting from s_1 to arrive at $(H_0 + 1, s)$. Formally, for each $h \leq H_0$, $i \in [1, S - 1]$, $j \in [1, A]$, $P_h(s_{A(i-1)+j}|s_i, a_j) = 1$ (here if $A(i-1) + j \geq S$, $s_{A(i-1)+j} = s^\dagger$). By induction, we can see that for $s \neq s^\dagger$, there is a unique trajectory that arrives at s at time step $H_0 + 1$. The reward r_h is always 0 when $h \leq H_0$.

The remaining layers For each state $s \neq s^\dagger$ and $H > h \geq H_0 + 1$, there exists a single action $a_{s,h}$ such that $P_h(s|s, a_{s,h}) = 1$ while $P_h(s'|s, a_{s,h}) = 0$ for $s' \neq s$. For any $a \neq a_{s,h}$, we have that $P_h(s^\dagger|s, a) = 1$. The reward function $r_h(s, a_{s,h}) = 0$ when the action is the $a_{s,h}$ that keeps the agent at s , and the reward for other actions $r_h(s, a)$ is unknown (can be non-zero) when $H > h \geq H_0 + 1$, $a \neq a_{s,h}$. For the last layer, for any state $s \neq s^\dagger$ and any action $a \in \mathcal{A}$, $P_H(s^\dagger|s, a) = 1$. The reward $r_H(s, a)$ is unknown and can be non-zero.

We can see that under this MDP, there are two cases. The first one is for some policies, the agent arrives at $(H_0 + 1, s^\dagger)$ with no reward, then such a trajectory will have total reward 0. The second one is the agent arrives at some $(H_0 + 1, s)$ ($s \neq s^\dagger$) and finally arrives at s^\dagger from the tuple (h, s, a) ($H > h \geq H_0 + 1$ and $a \neq a_{s,h}$ or $h = H$) with total reward $r_h(s, a)$. Also, for any deterministic policy, the trajectory is fixed, like pulling an ‘‘arm’’ in bandit setting. Note that the total number of such ‘‘arms’’ with non-zero unknown reward is at least $(S - 1)(A - 1)\frac{H}{2} \geq \Omega(HSA)$. Even if the transition kernel is known to the agent, this MDP is still as difficult as a multi-armed bandit problem with $\Omega(HSA)$ arms. Then Theorem H.1 results from Lemma H.3 while Theorem H.2 holds because of the following Lemma H.4. \square

Lemma H.3 (Theorem 2 in (Simchi-Levi & Xu, 2019)). *Under the K -armed bandits problem, there exists an absolute constant $C > 0$ such that for all $K > 1, S \geq 0, T \geq 2K$ and for all policy π with switching budget S , the regret satisfies*

$$R^\pi(K, T) \geq \frac{C}{\log T} \cdot K^{1 - \frac{1}{2 - 2^{-q(S, K)} - 1}} T^{\frac{1}{2 - 2^{-q(S, K)} - 1}},$$

where $q(S, K) = \lfloor \frac{S-1}{K-1} \rfloor$. This further implies that $\Omega(K \log \log T)$ switches are necessary for achieving $\tilde{O}(\sqrt{T})$ regret bound.

Lemma H.4 (Switching cost lower bound under MAB setting). *For any algorithm with sub-linear regret bound under K -armed bandit problem, the switching cost is at least $\Omega(K)$.*

Proof of Lemma H.4. The K -armed bandit problem can be described by a vector $\mu = [\mu_1, \dots, \mu_K]$, where μ_k is the mean reward of the k -th arm. We consider the following base problem and K possible problems.

$$\mu_{\text{base}} = [0, 0, \dots, 0], \mu_{\text{problem } k} = [\mu_1, \dots, \mu_K, \text{ where } \mu_i = \mathbb{1}(i = k)], \forall k \in [K].$$

Note that under the base problem, all arms have reward 0 while under problem k , only the k -th arm has reward 1 and all other arms have reward 0. We will prove that for any algorithm with switching cost bounded by $\frac{K}{2} - 1$, even if the reward is deterministic, the regret bound can not be sub-linear.

For any algorithm Alg with switching cost smaller than $\frac{K}{2} - 1$, because the maximum is larger than the average, we only need to provide a lower bound for $R = \frac{1}{K} \sum_{k=1}^K \mathbb{E}_{Alg}[\text{Regret}(T)|\text{problem } k]$.

Consider using algorithm Alg on the base problem, let $\mathcal{T} = \{\tau\}$ be all the possible strings of $\{a_1, r_1, \dots, a_T, r_T\}$, where a_i is the index of the arm pulled at time i and r_i is its reward. Then we have $\sum_{\tau \in \mathcal{T}} \mathbb{P}_{Alg}[\tau|\text{base problem}] = 1$. In addition, because of the restriction on Alg that the switching cost is bounded by $\frac{K}{2} - 1$, each τ can only pull at most $\frac{K}{2}$ arms. Then it

holds that

$$\begin{aligned}
 R &= \frac{1}{K} \sum_{k=1}^K \mathbb{E}_{Alg}[\text{Regret}(T)|\text{problem } k] \\
 &= \frac{1}{K} \sum_{k=1}^K \sum_{\tau \in \mathcal{T}} [\text{Regret}(\tau)|\text{problem } k] \times \mathbb{P}_{Alg}[\tau|\text{problem } k] \\
 &= \sum_{\tau \in \mathcal{T}} \frac{1}{K} \sum_{k=1}^K [\text{Regret}(\tau)|\text{problem } k] \times \mathbb{P}_{Alg}[\tau|\text{problem } k] \\
 &\geq \sum_{\tau \in \mathcal{T}} \frac{T \times \mathbb{P}_{Alg}[\tau|\text{base problem}]}{2} \\
 &= \frac{T}{2},
 \end{aligned} \tag{16}$$

where $[\text{Regret}(\tau)|\text{problem } k]$ is the regret of string τ under problem k . The inequality is because for any string τ , if problem k satisfies that the k -th arm does not appear in τ , then $[\text{Regret}(\tau)|\text{problem } k] = T$ and $\mathbb{P}_{Alg}[\tau|\text{problem } k] = \mathbb{P}_{Alg}[\tau|\text{base problem}]$. \square

I. Proof of Theorem 5.1

Lemma I.1. *The total global switching cost of Algorithm 4 is bounded by $2HSA$.*

Remark I.2. The switching cost $O(HSA)$ is optimal for reward-free setting. With the same MDP in Appendix H, we can show that it takes $\Omega(HSA)$ switching cost to find out the mapping between the policy and which ‘‘arm’’ it is pulling.

In this part of proof, we also construct the absorbing MDP \tilde{P} in the same way as Definition 3.1 based on the infrequent tuples \mathcal{F} and the true MDP P . Similar to the proof of Lemma G.3, we construct the universal constant C , we can first let C be the constant hidden by the big O in Lemma F.3. Then because of Lemma E.12, we can choose $C = \max\{C, 672\}$. With the C , we are able to choose the universal constant in this section to be $c' = 16C^2$. Note that the ι used in Algorithm 4 is $\iota = \log(\frac{2HA(N_0+N)}{\delta})$, while the ι' used in the upper bound of sample complexity is $\iota' = \log(\frac{HSA}{\epsilon\delta})$. According to the conclusion about Algorithm 2, we have the following lemma regarding the choice of N_0 .

Lemma I.3. *There exists $c' > 0$, for any $\epsilon > 0$, when $N_0 > c' \cdot \frac{S^3AH^5\iota}{\epsilon}$, it holds that with probability $1 - \frac{\delta}{2}$, for any $\pi \in \phi^1$ and any reward function r , $|V^\pi(r, P) - V^\pi(r, \tilde{P})| \leq \frac{\epsilon}{4}$, while P^{int} is $\frac{1}{H}$ -multiplicatively accurate to \tilde{P} .*

Proof of Lemma I.3. This is a direct corollary of lemma E.3 and lemma E.12. \square

Then because of the conclusions about Algorithm 3, we have the following lemma regarding the choice of N .

Lemma I.4. *There exists $c' > 0$, for any $\epsilon > 0$, when $N > c' \cdot \frac{H^5S^2A\iota}{\epsilon^2}$, conditioned on the case in Lemma I.3 that P^{int} is $\frac{1}{H}$ -multiplicatively accurate to \tilde{P} , with probability $1 - \frac{\delta}{2}$, for any $\pi \in \phi^1$ and reward function r , $|V^\pi(r, \hat{P}) - V^\pi(r, \tilde{P})| \leq \frac{\epsilon}{4}$.*

Proof of Lemma I.4. This is a direct corollary of Lemma F.3. \square

Lemma I.5. *There exists $c' > 0$, for any $\epsilon > 0$, when the number of total episodes $K > c' \cdot (\frac{H^5S^2A\iota}{\epsilon^2} + \frac{S^3AH^5\iota}{\epsilon})$, there exists a choice of N_0 and N such that $N_0 + N = K$ and with probability $1 - \delta$, for any $\pi \in \phi^1$ and reward function r , $|V^\pi(r, \hat{P}) - V^\pi(r, P)| \leq \frac{\epsilon}{2}$.*

Proof of Lemma I.5. Because of triangular inequality,

$$|V^\pi(r, \hat{P}) - V^\pi(r, P)| \leq |V^\pi(r, \hat{P}) - V^\pi(r, \tilde{P})| + |V^\pi(r, \tilde{P}) - V^\pi(r, P)|.$$

Because of Lemma I.3 and Lemma I.4, if we choose $N_0 > c' \cdot \frac{S^3AH^5\iota}{\epsilon}$ and $N > c' \cdot \frac{H^5S^2A\iota}{\epsilon^2}$, it holds that with probability $1 - \delta$, for any $\pi \in \phi^1$ and reward function r ,

$$|V^\pi(r, P) - V^\pi(r, \tilde{P})| \leq \frac{\epsilon}{4}, \quad |V^\pi(r, \hat{P}) - V^\pi(r, \tilde{P})| \leq \frac{\epsilon}{4}.$$

Then the proof is finished by plugging in these two inequalities to the triangular inequality. \square

Note that $K > c' \cdot (\frac{H^5 S^2 A \iota}{\epsilon^2} + \frac{S^3 A H^5 \iota'}{\epsilon})$ is not a good representation of constraints on K because K appears on both sides of the inequality. The following Lemma I.6 gives a valid solution to this inequality.

Lemma I.6. *For a fixed c' , there exists a constant c such that if $K > c(\frac{H^5 S^2 A \iota'}{\epsilon^2} + \frac{S^3 A H^5 \iota'}{\epsilon})$, then $K > c' \cdot (\frac{H^5 S^2 A \iota}{\epsilon^2} + \frac{S^3 A H^5 \iota'}{\epsilon})$, where $\iota = \log(\frac{2HAK}{\delta})$ and $\iota' = \log(\frac{HSA}{\epsilon\delta})$.*

Lemma I.7. *Conditioned on the case in Lemma I.5, we have that $0 \leq V^{\pi^*}(r, P) - V^{\hat{\pi}^r}(r, P) \leq \epsilon$ holds for any reward function r .*

Proof of Lemma I.7. $V^{\pi^*}(r, P) - V^{\hat{\pi}^r}(r, P) \geq 0$ directly results from the definition of optimal policy. Also,

$$\begin{aligned} V^{\pi^*}(r, P) - V^{\hat{\pi}^r}(r, P) &\leq |V^{\pi^*}(r, P) - V^{\pi^*}(r, \hat{P})| + V^{\pi^*}(r, \hat{P}) - V^{\hat{\pi}^r}(r, \hat{P}) + |V^{\hat{\pi}^r}(r, \hat{P}) - V^{\hat{\pi}^r}(r, P)| \\ &\leq \epsilon. \end{aligned}$$

The second inequality is because of Lemma I.5 and $\hat{\pi}^r = \operatorname{argmax}_{\pi \in \phi^1} V^\pi(r, \hat{P})$. \square

Then Theorem 5.1 holds because of Lemma I.1, Lemma I.5, Lemma I.6 and Lemma I.7.

J. Improved algorithm with near optimal number of batches

First of all, we will analyze the batch complexity of APEVE (Algorithm 1). Note that one application of Crude Exploration (Algorithm 2) can be finished in H batches, because the exploration of each layer can be finished in one batch. Besides, one application of Fine Exploration (Algorithm 3) can be finished in one batch. According to the schedule of APEVE (Algorithm 1), there are $O(\log \log T)$ stages, each stage contains one Crude Exploration and one Fine Exploration. Therefore, the batch complexity of APEVE is $O(H \log \log T)$.

We further improve the batch complexity by revising APEVE (Algorithm 1) slightly to get this APEVE+ (Algorithm 6). The main difference is that in Algorithm 6, only the first two stages contain the use of Crude Exploration (Algorithm 2), the remaining stages only run Fine Exploration (Algorithm 3) and eliminate policies at the end of each stage. First, we consider the number of batches. In Algorithm 6, there are two applications of Crude Exploration and $O(\log \log T)$ applications of Fine Exploration, so we have the following theorem.

Theorem J.1. *APEVE+ (Algorithm 6) can be applied in $O(2H + \log \log T) = O(H + \log \log T)$ batches.*

Remark J.2. Gao et al. (2019) proved that for any algorithm with $\tilde{O}(\sqrt{T})$ regret bound under multi-armed bandit setting, the number of batches is at least $\Omega(\log \log T)$. In the construction of our lower bound in Appendix H, we show that tabular RL can be at least as difficult as a multi-armed bandit problem, which means the $\Omega(\log \log T)$ lower bound on batches also applies to tabular RL. Theorem B.3 in (Huang et al., 2022) states an $\Omega(\frac{H}{\log T})$ lower bound for number of batches for any algorithm with PAC guarantee. Because regret guarantee is stronger than PAC guarantee, this lower bound also applies to any algorithm with $\tilde{O}(\sqrt{T})$ regret bound. Combining these two results, we have an $\Omega(\frac{H}{\log T} + \log \log T)$ lower bound on number of batches for any algorithm with $\tilde{O}(\sqrt{T})$ regret. According to Theorem J.1, we conclude that our Algorithm 6 nearly matches the lower bound of batches.

Now we will consider the regret bound of APEVE+. We have the following key lemma whose proof and choice of the constant C is identical to Lemma G.3.

Lemma J.3. *There exists a constant C , such that with probability $1 - \delta$, for any $k = 1, 2$ and $\pi \in \phi^k$,*

$$|V^\pi(r, \hat{P}^k) - V^\pi(r, P)| \leq C \left(\sqrt{\frac{H^5 S^2 A \iota}{T^{(k)}}} + \frac{S^3 A^2 H^5 \iota}{T^{(k)}} \right),$$

while for any $k \geq 3$ and $\pi \in \phi^k$,

$$|V^\pi(r, \hat{P}^k) - V^\pi(r, P)| \leq C \left(\sqrt{\frac{H^5 S^2 A \iota}{T^{(k)}}} + \frac{S^3 A^2 H^5 \iota}{T^{(2)}} \right).$$

Then using the identical proof as Lemma G.4 and Lemma G.5, we have an upper bound on the sub-optimality of policies used in each stage.

Lemma J.4. *With probability $1 - \delta$, for the policies that have not been eliminated at stage k , i.e. $\pi \in \phi^{k+1}$, we have that*

$$V^{\pi^*}(r, P) - V^\pi(r, P) \leq 4C\left(\sqrt{\frac{H^5 S^2 A \iota}{T^{(k)}}} + \frac{S^3 A^2 H^5 \iota}{T^{(k)}}\right), \quad k = 1, 2.$$

$$V^{\pi^*}(r, P) - V^\pi(r, P) \leq 4C\left(\sqrt{\frac{H^5 S^2 A \iota}{T^{(k)}}} + \frac{S^3 A^2 H^5 \iota}{T^{(2)}}\right), \quad k \geq 3.$$

Now we are ready to bound the total regret of Algorithm 6.

Theorem J.5. *If $K \geq \tilde{\Omega}(S^8 A^6 H^{10})$, with probability $1 - \delta$, the total regret of APEVE+ (Algorithm 6) is at most $\tilde{O}(\sqrt{H^4 S^2 A T})$.*

Proof of Theorem J.5. Because the case in Lemma J.4 holds with probability $1 - \delta$, we will prove under the case in Lemma J.4 and show that the regret is $\tilde{O}(\sqrt{H^4 S^2 A T})$.

The regret for the first stage (stage 1) is at most $2HT^{(1)} = O(HK^{\frac{1}{2}})$.

Because of Lemma J.4, the regret for the second stage (stage 2) is at most $2T^{(2)} \times 4C\left(\sqrt{\frac{H^5 S^2 A \iota}{T^{(1)}}} + \frac{S^3 A^2 H^5 \iota}{T^{(1)}}\right)$.

For stage $k \geq 3$, the policies we use (any $\pi \in \phi^k$) are at most $4C\left(\sqrt{\frac{H^5 S^2 A \iota}{T^{(k-1)}}} + \frac{S^3 A^2 H^5 \iota}{T^{(2)}}\right)$ sub-optimal, so the regret for the k -th stage is at most $8CT^{(k)}\left(\sqrt{\frac{H^5 S^2 A \iota}{T^{(k-1)}}} + \frac{S^3 A^2 H^5 \iota}{T^{(2)}}\right)$.

Adding up the regret for each stage, we have that the total regret is bounded by

$$\begin{aligned} \text{Regret}(T) &\leq 2HK^{\frac{1}{2}} + \sum_{k=2}^{K_0} 8CT^{(k)}\sqrt{\frac{H^5 S^2 A \iota}{T^{(k-1)}}} + 8CT^{(2)}\frac{S^3 A^2 H^5 \iota}{T^{(1)}} + \sum_{k=3}^{K_0} 8CT^{(k)}\frac{S^3 A^2 H^5 \iota}{T^{(2)}} \\ &= O(HK^{\frac{1}{2}}) + O(\sqrt{H^5 S^2 A K \iota} \cdot \log \log K) + O(S^3 A^2 H^5 K^{\frac{1}{4}} \iota) \\ &= O(\sqrt{H^5 S^2 A K \iota} \cdot \log \log K) + O(S^3 A^2 H^5 K^{\frac{1}{4}} \iota) \\ &= \tilde{O}(\sqrt{H^4 S^2 A T}), \end{aligned}$$

where the last equality is because $K \geq \tilde{\Omega}(S^8 A^6 H^{10})$. □

Algorithm 6 Adaptive Policy Elimination by Value Estimation+ (APEVE+)

- 1: **Require:** Number of episodes for exploration K , r is the known deterministic reward. Universal constant C . Failure probability δ .
- 2: **Initialize:** $T^{(k)} = K^{1-\frac{1}{2^k}}$, $k \leq K_0 = O(\log \log K)$, $\phi^1 := \{\text{the set of all the deterministic policies}\}$, $\iota = \log(2HAK/\delta)$.
- 3: **for** $k = 1, 2, \dots, K_0$ **do**
- 4: \diamond **Number of episodes in k -th stage:**
- 5: **if** $T^{(1)} + T^{(2)} + \sum_{i=1}^k T^{(i)} \geq K$ **then**
- 6: $T^{(k)} = K - T^{(1)} - T^{(2)} - \sum_{i=1}^{k-1} T^{(i)}$. (o.w. $T^{(k)} = K^{1-\frac{1}{2^k}}$)
- 7: **end if**
- 8: **end for**
- 9: **Adaptive policy elimination for the first two stages:**
- 10: **for** $k = 1, 2$ **do**
- 11: **Update the infrequent set \mathcal{F}^k and construct empirical estimate of the absorbing MDP:**
- 12: $\mathcal{F}^k, P^{int,k} = \text{Crude Exploration}(\phi^k, T^{(k)})$.
- 13: $\hat{P}^k = \text{Fine Exploration}(\mathcal{F}^k, P^{int,k}, T^{(k)}, \phi^k)$.
- 14: $U^k = \emptyset$
- 15: **for** $\pi \in \phi^k$ **do**
- 16: **if** $V^\pi(r, \hat{P}^k) \leq \sup_{\hat{\pi} \in \phi^k} V^{\hat{\pi}}(r, \hat{P}^k) - 2C(\sqrt{\frac{H^5 S^2 A \iota}{T^{(k)}}} + \frac{S^3 A^2 H^5 \iota}{T^{(k)}})$ **then**
- 17: Update $U^k \leftarrow U^k \cup \{\pi\}$.
- 18: **end if**
- 19: **end for**
- 20: $\phi^{k+1} \leftarrow \phi^k \setminus U^k$.
- 21: **end for**
- 22: **Adaptive policy elimination for the remaining stages:**
- 23: **for** $k = 3, 4, \dots, K_0$ **do**
- 24: **Keep the infrequent set \mathcal{F}^2 and construct empirical estimate of the absorbing MDP with new data set:**
- 25: $\hat{P}^k = \text{Fine Exploration}(\mathcal{F}^2, P^{int,2}, T^{(k)}, \phi^k)$.
- 26: $U^k = \emptyset$
- 27: **for** $\pi \in \phi^k$ **do**
- 28: **if** $V^\pi(r, \hat{P}^k) \leq \sup_{\hat{\pi} \in \phi^k} V^{\hat{\pi}}(r, \hat{P}^k) - 2C(\sqrt{\frac{H^5 S^2 A \iota}{T^{(k)}}} + \frac{S^3 A^2 H^5 \iota}{T^{(2)}})$ **then**
- 29: Update $U^k \leftarrow U^k \cup \{\pi\}$.
- 30: **end if**
- 31: **end for**
- 32: $\phi^{k+1} \leftarrow \phi^k \setminus U^k$.
- 33: **end for**