Provably Efficient Q-Learning with Low Switching Cost

Yu Bai [*] Tengyang Xie [†] Nan Jiang [‡] Yu-Xiang Wang [§]

June 8, 2019

Abstract

We take initial steps in studying PAC-MDP algorithms with limited adaptivity, that is, algorithms that change its exploration policy as infrequently as possible during regret minimization. This is motivated by the difficulty of running fully adaptive algorithms in real-world applications (such as medical domains), and we propose to quantify adaptivity using the notion of *local switching cost*. Our main contribution, Q-Learning with UCB2 exploration, is a model-free algorithm for H-step episodic MDP that achieves sublinear regret whose local switching cost in K episodes is $O(H^3 SA \log K)$, and we provide a lower bound of $\Omega(HSA)$ on the local switching cost for any no-regret algorithm. Our algorithm can be naturally adapted to the concurrent setting [12], which yields nontrivial results that improve upon prior work in certain aspects.

1 Introduction

This paper is concerned with reinforcement learning (RL) under *limited adaptivity* or *low switching cost*, a setting in which the agent is allowed to act in the environment for a long period but is constrained to switch its policy for at most N times. A small switching cost N restricts the agent from frequently adjusting its exploration strategy based on feedback from the environment.

There are strong practical motivations for developing RL algorithms under limited adaptivity. The setting of restricted policy switching captures various real-world settings where deploying new policies comes at a cost. For example, in medical applications where actions correspond to treatments, it is often unrealistic to execute fully adaptive RL algorithms – instead one can only run a fixed policy approved by the domain experts to collect data, and a separate approval process is required every time one would like to switch to a new policy [18, 2, 3]. In personalized recommendation [24], it is computationally impractical to adjust the policy online based on instantaneous data, and a more common practice is to aggregate data in a long period before deploying a new policy. In problems where we run RL for compiler optimization [4] and hardware placements [19], as well as for learning to optimize databases [17], often it is desirable to limit the frequency of changes to the policy since it is costly to recompile the code, to run profiling, to reconfigure an FPGA devices, or to restructure a deployed relational database. The problem is even more prominent in the RL-guided new material discovery as it takes time to fabricate the materials and setup the experiments [23, 20]. In many of these applications, adaptivity turns out to be really the bottleneck.

Understanding limited adaptivity RL is also important from a theoretical perspective. First, algorithms with low adaptivity (a.k.a. "batched" algorithms) that are as effective as their fully

[*]Stanford University. `yub@stanford.edu`

[†]University of Illinois at Urbana-Champaign. `tx10@illinois.edu`

[‡]University of Illinois at Urbana-Champaign. `nanjiang@illinois.edu`

[§]UC Santa Barbara. `yuxiangw@cs.ucsb.edu`

sequential counterparts have been established in bandits [22, 11], online learning [8], and optimization [10], and it would be interesting to extend such undertanding into RL. Second, algorithms with few policy switches are naturally easy to parallelize as there is no need for parallel agents to communicate if they just execute the same policy. Third, limited adaptivity is closed related to off-policy RL[1] and offers a relaxation less challenging than the pure off-policy setting.

In this paper, we take initial steps towards studying theoretical aspects of limited adaptivity RL through designing *low-regret algorithms* with limited adaptivity. We focus on model-free algorithms, in particular Q-Learning, which was recently shown to achieve a $\widetilde{O}(\sqrt{\text{poly}(H) \cdot SAT})$ regret bound with UCB exploration and a careful stepsize choice by Jin et al. [15]. Our goal is to design Q-Learning type algorithms that achieve similar regret bounds with a bounded switching cost.

The main contributions of this paper are summarized as follows:

- We propose a notion of *local switching cost* that captures the adaptivity of an RL algorithm in episodic MDPs (Section 2). Algorithms with lower local switching cost will make fewer switches in its deployed policies.
- Building on insights from the UCB2 algorithm in multi-armed bandits [5] (Section 3), we propose our main algorithms, *Q-Learning with UCB2-*$\{$*Hoeffding, Bernstein*$\}$ *exploration*. We prove that these two algorithms achieve $\widetilde{O}(\sqrt{H^{\{4,3\}}SAT})$ regret (respectively) and $O(H^3SA\log(K/A))$ local switching cost (Section 4). The regret matches their vanilla counterparts of [15] but the switching cost is only logarithmic in the number of episodes.
- We show how our low switching cost algorithms can be applied in the *concurrent RL* setting [12], in which multiple agents can act in parallel (Section 5). The parallelized versions of our algorithms with UCB2 exploration give rise to *Concurrent Q-Learning* algorithms, which achieve a nearly linear speedup in execution time and compares favorably against existing concurrent algorithms in sample complexity for exploration.
- We show a simple $\Omega(HSA)$ lower bound on the switching cost for any sublinear regret algorithm, which has at most a $O(H^2\log(K/A))$ gap from the upper bound (Section 7).

1.1 Prior work

Low-regret RL Sample-efficient RL has been studied extensively since the classical work of Kearns and Singh [16] and Brafman and Tennenholtz [7], with a focus on obtaining a near-optimal policy in polynomial time, i.e. PAC guarantees. A subsequent line of work initiate the study of regret in RL and provide algorithms that achieve regret $\widetilde{O}(\sqrt{\text{poly}(H,S,A)\cdot T})$ [14, 21, 1]. In our episodic MDP setting, the information-theoretic lower bound for the regret is $\Omega(\sqrt{H^2SAT})$, which is matched in recent work by the UCBVI [6] and ORLC [9] algorithms. On the other hand, while all the above low-regret algorithms are essentially model-based, the recent work of [15] shows that model-free algorithms such as Q-learning are able to achieve $\widetilde{O}(\sqrt{H^{\{4,3\}}SAT})$ regret which is only $O(\sqrt{H})$ worse than the lower bound.

Low switching cost / batched algorithms Auer et al. [5] propose UCB2 in bandit problems, which achieves the same regret bound as UCB but has switching cost only $O(\log T)$ instead of the naive $O(T)$. Cesa-Bianchi et al. [8] study the switching cost in online learning in both the adversarial and stochastic setting, and design an algorithm for stochastic bandits that acheive optimal regert and $O(\log\log T)$ switching cost.

[1] In particular, $N = 0$ corresponds to off-policy RL, where the algorithm can only choose one data collection policy [13].

Learning algorithms with switching cost bounded by a fixed $O(1)$ constant is often referred to as *batched algorithms*. Minimax rates for batched algorithms have been established in various problems such as bandits [22, 11] and convex optimization [10]. In all these scenarios, minimax optimal M-batch algorithms are obtained for all M, and their rate matches that of fully adaptive algorithms once $M = O(\log\log T)$.

2 Problem setup

In this paper, we consider undiscounted episodic tabular MDPs of the form $(H, \mathcal{S}, \mathbb{P}, \mathcal{A}, r)$. The MDP has horizon H with trajectories of the form $(x_1, a_1, \ldots, x_H, a_H, x_{H+1})$, where $x_h \in \mathcal{S}$ and $a_h \in \mathcal{A}$. The state space $\mathcal{S}$ and action space $\mathcal{A}$ are discrete with $|\mathcal{S}| = S$ and $|\mathcal{A}| = A$. The initial state x_1 can be either adversarial (chosen by an adversary who has access to our algorithm), or stochastic specified by some distribution $\mathbb{P}_0(x_1)$. For any $(h, x_h, a_h) \in [H] \times \mathcal{S} \times \mathcal{A}$, the transition probability is denoted as $\mathbb{P}_h(x_{h+1}|x_h, a_h)$. The reward is denoted as $r_h(x_h, a_h) \in [0, 1]$, which we assume to be deterministic[2]. We assume in addition that $r_{h+1}(x) = 0$ for all x, so that the last state x_{H+1} is effectively an (uninformative) absorbing state.

A deterministic policy π consists of H sub-policies $\pi^h(\cdot) : \mathcal{S} \to \mathcal{A}$. For any deterministic policy π, let $V_h^\pi(\cdot) : \mathcal{S} \to \mathbb{R}$ and $Q_h^\pi(\cdot, \cdot) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denote its value function and state-action value function at the h-th step respectively. Let $\pi_\star$ denote an optimal policy, and $V_h^\star = V_h^{\pi_\star}$ and $Q_h^\star = Q_h^{\pi_\star}$ denote the optimal V and Q functions for all h. As a convenient short hand, we denote $[\mathbb{P}_h V_{h+1}](x, a) := \mathbb{E}_{x' \sim \mathbb{P}(\cdot|x,a)}[V_{h+1}(x')]$ and also use $[\widehat{\mathbb{P}}_h V_{h+1}](x_h, a_h) := V_{h+1}(x_{h+1})$ in the proofs to denote observed transition. Unless otherwise specified, we will focus on deterministic policies in this paper, which will be without loss of generality as there exists at least one deterministic policy $\pi_\star$ that is optimal.

Regret We focus on the regret for measuring the performance of RL algorithms. Let K be the number of episodes that the agent can play. (so that total number of steps is $T := KH$.) The regret of an algorithm is defined as

$$\mathrm{Regret}(K) := \sum_{k=1}^{K} \left[V_1^\star(x_1^k) - V_1^{\pi_k}(x_1^k) \right],$$

where π_k is the policy it employs before episode k starts, and $V_1^\star$ is the optimal value function for the entire episode.

Miscellanous notation We use standard Big-Oh notations in this paper: $A_n = O(B_n)$ means that there exists an absolute constant $C > 0$ such that $A_n \le CB_n$ (similarly $A_n = \Omega(B_n)$ for $A_n \ge CB_n$). $A_n = \widetilde{O}(B_n)$ means that $A_n \le C_n B_n$ where C_n depends at most poly-logarithmically on all the problem parameters.

2.1 Measuring adaptivity through local switching cost

To quantify the adaptivity of RL algorithms, we consider the following notion of *local switching cost* for RL algorithms.

[2] Our results can be straightforwardly extended to the case with stochastic rewards.

Definition 2.1. *The local switching cost (henceforth also "switching cost") between any pair of policies (π, π') is defined as the number of (h, x) pairs on which π and π' are different:*

$$n_{\text{switch}}(\pi, \pi') := \left| \left\{ (h, x) \in [H] \times \mathcal{S} : \pi^h(x) \neq [\pi']^h(x) \right\} \right|.$$

For an RL algorithm that employs policies $(\pi_1, \dots, \pi_K)$, its local switching cost is defined as

$$N_{\text{switch}} := \sum_{k=1}^{K-1} n_{\text{switch}}(\pi_k, \pi_{k+1}).$$

Note that (1) N_{switch} is a random variable in general, as π_k can depend on the outcome of the MDP; (2) we have the trivial bound $n_{\text{switch}}(\pi, \pi') \le HS$ for any (π, π') and $N_{\text{switch}}(\mathcal{A}) \le HS(K-1)$ for any algorithm $\mathcal{A}$.

Remark The local switching cost extends naturally the notion of switching cost in online learning [8] and is suitable in scenarios where the cost of deploying a new policy scales with the portion of (h, x) on which the action $\pi^h(x)$ is changed.

A closely related notion of adaptivity is the *global switching cost*, which simply measures how many times the algorithm switches its entire policy:

$$N_{\text{switch}}^{\text{gl}} = \sum_{k=1}^{K-1} \mathbf{1}\left\{\pi_k \neq \pi_{k+1}\right\}.$$

As $\pi_k \neq \pi_{k+1}$ implies $n_{\text{switch}}(\pi_k, \pi_{k+1}) \ge 1$, we have the trivial bound that $N_{\text{switch}}^{\text{gl}} \le N_{\text{switch}}$. However, the global switching cost can be substantially smaller for algorithms that tend to change the policy "entirely" rather than "locally". In this paper, we focus on bounding N_{switch}, and leave the task of tighter bounds on $N_{\text{switch}}^{\text{gl}}$ as future work.

3 UCB2 for multi-armed bandits

To gain intuition about the switching cost, we briefly review the UCB2 algorithm [5] on multi-armed bandit problems, which achieves the same regret bound as the original UCB but has a substantially lower switching cost.

The multi-armed bandit problem can be viewed as an RL problem with $H = 1$, $S = 1$, so that the agent needs only play one action $a \in \mathcal{A}$ and observe the (random) reward $r(a) \in [0, 1]$. The distribution of $r(a)$'s are unknown to the agent, and the goal is to achieve low regret.

The UCB2 algorithm is a variant of the celebrated UCB (Upper Confidence Bound) algorithm for bandits. UCB2 also maintains upper confidence bounds on the true means $\mu_1, \dots, \mu_A$, but instead plays each arm multiple times rather than just once when it's found to maximize the upper confidence bound. Specifically, when an arm is found to maximize the UCB for the r-th time, UCB2 will play it $\tau(r) - \tau(r-1)$ times, where

$$\tau(r) = (1 + \eta)^r$$

for $r = 0, 1, 2, \dots$ and some parameter $\eta \in (0, 1)$ to be determined. [3] The full UCB2 algorithm is presented in Algorithm 1.

[3] For convenience, here we treat $(1+\eta)^r$ as an integer. In Q-learning we could not make this approximation (as we choose η super small), and will massage the sequence $\tau(r)$ to deal with it.

Algorithm 1 UCB2 for multi-armed bandits

input Parameter $\eta \in (0,1)$.

Initialize: $r_j = 0$ for $j = 1, \ldots, A$. Play each arm once. Set $t \leftarrow 0$ and $T \leftarrow T - A$.

while $t \le T$ **do**

Select arm j that maximizes $\overline{r}_j + a_{r_j}$, where $\overline{r}_j$ is the average reward obtained from arm j and $a_r = O(\sqrt{\log T/\tau(r)})$ (with some specific choice.)

Play arm j exactly $\tau(r_j+1) - \tau(r_j)$ times.

Set $t \leftarrow t + \tau(r_j+1) - \tau(r_j)$ and $r_j \leftarrow r_j + 1$.

end while

Theorem 1 (Auer et al. [5]). *For $T \ge \max_{i:\mu_i<\mu^\star} \frac{1}{2\Delta_i^2}$, the UCB2 algorithm acheives expected regret bound*

$$\mathbb{E}\left[\sum_{t=1}^{T}(\mu^\star - \mu_t)\right] \le O_\eta\left(\log T \cdot \sum_{i:\mu_i<\mu^\star} \frac{1}{\Delta_i}\right),$$

where $\Delta_i := \mu^\star - \mu_i$ is the gap between arm i and the optimal arm. Further, the switching cost is at most $O(\frac{A\log(T/A)}{\eta})$.

The switching cost bound in Theorem 1 comes directly from the fact that $\sum_{i=1}^{A}(1+\eta)^{r_i} \le T$ implies $\sum_{i=1}^{A} r_i \le O(A\log(T/A)/\eta)$, by the convexity of $r \mapsto (1+\eta)^r$ and Jensen's inequality. Such an approach can be fairly general, and we will follow it in sequel to develop RL algorithm with low switching cost.

4 Q-learning with UCB2 exploration

In this section, we propose our main algorithm, Q-learning with UCB2 exploration, and show that it achieves sublinear regret as well as logarithmic local switching cost.

4.1 Algorithm description

High-level idea Our algorithm maintains wo sets of optimistic Q estimates: a *running estimate* $\widetilde{Q}$ which is updated after every episode, and a *delayed estimate* Q which is only updated occasionally but used to select the action. In between two updates to Q, the policy stays fixed, so the number of policy switches is bounded by the number of updates to Q.

To describe our algorithm, let $\tau(r)$ be defined as

$$\tau(r) = \lceil (1+\eta)^r \rceil, \quad r = 1, 2, \ldots$$

and define the *triggering sequence* as

$$\{t_n\}_{n\ge 1} = \{1, 2, \ldots, \tau(r_\star)\} \cup \{\tau(r_\star+1), \tau(r_\star+2), \ldots\}, \tag{1}$$

where the parameters $(\eta, r_\star)$ will be inputs to the algorithm. Define for all $t \in \{1, 2, \ldots\}$ the quantities

$$\tau_{\mathrm{last}}(t) := \max\{t_n : t_n \le t\} \quad \text{and} \quad \alpha_t = \frac{H+1}{H+t}.$$

Two-stage switching strategy The triggering sequence (1) defines a *two-stage strategy* for switching policies. Suppose for a given (h, x_h), the algorithm decides to take some particular a_h for the t-th time, and has observed (r_h, x_{h+1}) and updated the running estimate $\widetilde{Q}_h(x_h, a_h)$ accordingly. Then, whether to also update the policy network Q is decided as

- Stage I: if $t \le \tau(r_\star)$, then always perform the update $Q_h(x_h, a_h) \leftarrow \widetilde{Q}_h(x_h, a_h)$.
- Stage II: if $t > \tau(r_\star)$, then perform the above update only if t is in the triggering sequence, that is, $t = \tau(r) = \lceil (1+\eta)^r \rceil$ for some $r > r_\star$.

In other words, for any state-action pair, the algorithm performs eager policy update in the beginning $\tau(r_\star)$ visitations, and switches to delayed policy update after that according to UCB2 scheduling.

Optimistic exploration bonus We employ either a Hoeffding-type or a Bernstein-type exploration bonus to make sure that our running Q estimates are optimistic. The full algorithm with Hoeffding-style bonus is presented in Algorithm 2.

Algorithm 2 Q-learning with UCB2-Hoeffding (UCB2H) Exploration

input Parameter $\eta \in (0, 1)$, $r_\star \in \mathbb{Z}_{>0}$, and $c > 0$.
  Initialize: $\widetilde{Q}_h(x, a) \leftarrow H$, $Q_h \leftarrow \widetilde{Q}_h$, $N_h(x, a) \leftarrow 0$ for all $(x, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$.
  for episode $k = 1, \dots, K$ **do**
    Receive x_1.
    for step $h = 1, \dots, H$ **do**
      Take action $a_h \leftarrow \arg\max_{a'} Q_h(x_h, a')$, and observe x_{h+1}.
      $t = N_h(x_h, a_h) \leftarrow N_h(x_h, a_h) + 1$;
      $b_t = c\sqrt{H^3 \ell / t}$ (Hoeffding-type bonus);
      $\widetilde{Q}_h(x_h, a_h) \leftarrow (1 - \alpha_t)\widetilde{Q}_h(x_h, a_h) + \alpha_t[r_h(x_h, a_h) + \widetilde{V}_{h+1}(x_{h+1}) + b_t]$.
      $\widetilde{V}_h(x_h) \leftarrow \min \left\{ H, \max_{a' \in \mathcal{A}} \widetilde{Q}_h(x_h, a') \right\}$.
      if $t \in \{t_n\}_{n \ge 1}$ (where t_n is defined in (1)) **then**
        (Update policy) $Q_h(x_h, \cdot) \leftarrow \widetilde{Q}_h(x_h, \cdot)$.
      end if
    end for
  end for

4.2 Regret and switching cost guarantee

We now present our main results.

Theorem 2 (Q-learning with UCB2H exploration achieves sublinear regret and low switching cost)**.** *Choosing* $\eta = \frac{1}{2H(H+1)}$ *and* $r_\star = \left\lceil \frac{\log(10H^2)}{\log(1+\eta)} \right\rceil$, *with probability at least* $1 - p$, *the regret of Algorithm 2 is bounded by* $\widetilde{O}(\sqrt{H^4 SAT})$. *Further, the local switching cost is bounded as* $N_{\rm switch} \le O(H^3 SA \log(K/A))$.

Theorem 2 shows that the total regret of Q-learning with UCB2 exploration is $\widetilde{O}(\sqrt{H^4 SAT})$, the same as UCB version of [15]. In addition, the local switching cost of our algorithm is only $O(H^3 SA \log(K/A))$, which is logarithmic in K, whereas the UCB version can have in the worst case the trivial bound $HS(K - 1)$. We give a high-level overview of the proof Theorem 2 in Section 6, and defer the full proof to Appendix A.

Bernstein version Replacing the Hoeffding bonus with a Bernstein-type bonus, we can achieve $\widetilde{O}(\sqrt{H^3SAT})$ regret ($\sqrt{H}$ better than UCB2H) and the same switching cost bound.

Theorem 3 (Q-learning with UCB2B exploration achieves sublinear regret and low switching cost)**.** *Choosing $\eta = \frac{1}{2H(H+1)}$ and $r_\star = \left\lceil \frac{\log(10H^2)}{\log(1+\eta)} \right\rceil$, with probability at least $1-p$, the regret of Algorithm 3 is bounded by $\widetilde{O}(\sqrt{H^3SAT})$ as long as $T = \widetilde{\Omega}(H^6S^2A^2)$. Further, the local switching cost is bounded as $N_{\rm switch} \le O(H^3SA\log(K/A))$.*

The full algorithm description, as well as the proof of Theorem 3, are deferred to Appendix B.

Compared with Q-learning with UCB [15], Theorem 2 and 3 demonstrate that "vanilla" low-regret RL algorithms such as Q-Learning can be turned into low switching cost versions without any sacrifice on the regret bound.

4.3 PAC guarantee

Our low switching cost algorithms can also achieve the PAC learnability guarantee. Specifically, we have the following

Corollary 4 (PAC bound for Q-Learning with UCB2 exploration)**.** *Suppose (WLOG) that x_1 is deterministic. For any $\varepsilon > 0$, Q-Learning with {UCB2H, UCB2B} exploration can output a (stochastic) policy $\widehat{\pi}$ such that with high probability*

$$V_1^\star(x_1) - V_1^{\widehat{\pi}}(x_1) \le \varepsilon$$

after $K = \widetilde{O}(H^{\{5,4\}}SA/\varepsilon^2)$ episodes.

The proof of Corollary 4 involves turning the regret bounds in Theorem 2 and 3 to PAC bounds using the online-to-batch conversion, similar as in [15]. The full proof is deferred to Appendix C.

5 Application: Concurrent Q-Learning

Our low switching cost Q-Learning can be applied to developing algorithms for *Concurrent RL* [12] – a setting in which multiple RL agents can act in parallel and hopefully accelerate the exploration in wall time.

Setting We assume there are M agents / machines, where each machine can interact with a independent copy of the episodic MDP (so that the transitions and rewards on the M MDPs are mutually independent). Within each episode, the M machines must play synchronously and cannot communiate, and can only exchange information after the entire episode has finished. Note that our setting is in a way more stringent than [12], which allows communication after each timestep.

We define a "round" as the duration in which the M machines simultanesouly finish one episode and (optionally) communicate and update their policies. We measure the performance of a concurrent algorithm in its required number of rounds to find an ε near-optimal policy. With larger M, we expect such number of rounds to be smaller, and the best we can hope for is a *linear speedup* in which the number of rounds scales as M^{-1}.

Concurrent Q-Learning Intuitively, any low switching cost algorithm can be made into a concurrent algorithm, as its execution can be parallelized in between two consecutive policy switches. Indeed, we can design concurrent versions of our low switching Q-Learning algorithm and achieve a nearly linear speedup.

Theorem 5 (Concurrent Q-Learning achieves nearly linear speedup)**.** *There exists concurrent versions of Q-Learning with {UCB2H, UCB2B} exploration such that, given a budget of M parallel machines, returns an ε near-optimal policy in*

$$\widetilde{O}\left(H^3SA + \frac{H^{\{5,4\}}SA}{\varepsilon^2 M}\right)$$

rounds of execution.

Theorem 5 shows that concurrent Q-Learning has a linear speedup so long as $M = \widetilde{O}(H^{\{2,1\}}/\varepsilon^2)$. In particular, in high-accuracy (small ε) cases, the constant overhead term H^3SA can be negligible and we essentially have a linear speedup over a wide range of M. The proof of Theorem 5 is deferred to Appendix D.

Comparison with existing concurrent algorithms Theorem 5 implies a PAC mistake bound as well: there exists concurrent algorithms on M machines, Concurrent Q-Learning with {UCB2H, UCB2B}, that performs a ε near-optimal action on all but

$$\widetilde{O}\left(H^4SAM + \frac{H^{\{6,5\}}SA}{\varepsilon^2}\right) := N_\varepsilon^{\mathsf{CQL}}$$

actions with high probability (detailed argument in Appendix D.2).

We compare ourself with the Concurrent MBIE (CMBIE) algorithm in [12], which considers the discounted and infinite-horizon MDPs, and has a mistake bound[4]

$$\widetilde{O}\left(\frac{S'A'M}{\varepsilon(1-\gamma')^2} + \frac{S'^2A'}{\varepsilon^3(1-\gamma')^6}\right) := N_\varepsilon^{\mathsf{CMBIE}}$$

Our concurrent Q-Learning compares favorably against CMBIE in terms of the mistake bound:

- Dependence on ε. CMBIE achieves $N_\varepsilon^{\mathsf{CMBIE}} = \widetilde{O}(\varepsilon^{-3} + \varepsilon^{-1}M)$, whereas our algorithm achieves $N_\varepsilon^{\mathsf{CQL}} = \widetilde{O}(\varepsilon^{-2} + M)$, better by a factor of ε^{-1}.

- Dependence on (H, S, A). These are not comparable in general, but under the "typical" correspondence[5] $S' \leftarrow HS$, $A' \leftarrow A$, $(1-\gamma')^{-1} \leftarrow H$, we get $N_\varepsilon^{\mathsf{CMBIE}} = \widetilde{O}(H^3SAM\varepsilon^{-1} + H^8S^2A\varepsilon^{-3})$. Compared to $N_\varepsilon^{\mathsf{CQL}}$, CMBIE has a higher dependence on H as well as a S^2 term due to its model-based nature.

[4] (S', A', γ') are the {# states, # actions, discount factor} of the discounted infinite-horizon MDP.

[5] One can transform an episodic MDP with S states to an infinite-horizon MDP with HS states. Also note that the "effective" horizon for discounted MDP is $(1-\gamma)^{-1}$.

6 Proof overview of Theorem 2

The proof of Theorem 2 involves two parts: the switching cost bound and the regret bound. The switching cost bound results directly from the UCB2 switching schedule, similar as in the bandit case (cf. Section 3). However, such a switching schedule results in delayed policy updates, which makes establishing the regret bound technically challenging.

The key to the $\widetilde{O}(\text{poly}(H)\cdot\sqrt{SAT})$ regret bound for "vanilla" Q-Learning in [15] is a *propagation of error* argument, which shows that the regret[6] from the h-th step and forward (henceforth the h-regret), defined as

$$\sum_{k=1}^{K}\widetilde{\delta}_h^k := \sum_{k=1}^{K}\left[\widetilde{V}_h^k - V_h^{\pi_k}\right](x_h^k),$$

is bounded by $1+1/H$ times the $(h+1)$-regret, plus some bounded error term. As $(1+1/H)^H = O(1)$, this fact can be applied recursively for $h = H, \dots, 1$ which will result in a total regret bound that is not exponential in H. The control of the (excess) error propagation factor by $1/H$ and the ability to converge are then achieved simultaneously via the stepsize choice $\alpha_t = \frac{H+1}{H+t}$.

In constrast, our low-switching version of Q-Learning updates the exploration policy in a delayed fashion according to the UCB2 schedule. Specifically, the policy at episode k does not correspond to the argmax of the running estimate $\widetilde{Q}^k$, but rather a previous version $Q^k = \widetilde{Q}^{k'}$ for some $k' \le k$. This introduces a mismatch between the Q used for exploration and the Q being updated, and it is a priori possible whether such a mismatch will blow up the propagation of error.

We resolve this issue via a novel error analysis, which at a high level consists of the following steps:

(i) We show that the quantity $\widetilde{\delta}_h^k$ is upper bounded by a *max error*

$$\widetilde{\delta}_h^k \le \left(\max\left\{\widetilde{Q}_h^{k'}, \widetilde{Q}_h^k\right\} - Q_h^{\pi_k}\right)(x_h^k, a_h^k) = \left(\widetilde{Q}_h^{k'} - Q_h^{\pi_k} + \left[\widetilde{Q}_h^k - \widetilde{Q}_h^{k'}\right]_+\right)(x_h^k, a_h^k)$$

(Lemma A.3). On the right hand side, the first term $\widetilde{Q}_h^{k'} - Q_h^{\pi_k}$ does not have a mismatch (as π_k depends on $\widetilde{Q}^{k'}$) and can be bounded similarly as in [15]. The second term $[\widetilde{Q}_h^k - \widetilde{Q}_h^{k'}]_+$ is a perturbation term, which we bound in a precise way that relates to stepsizes in between episodes k' to k and the $(h+1)$-regret (Lemma A.4).

(ii) We show that, under the UCB2 scheduling, the combined error above results a mild blowup in the relation between h-regret and $(h+1)$-regret – the multiplicative factor can be now bounded by $(1+1/H)(1+O(\eta H))$ (Lemma A.5). Choosing $\eta = O(1/H^2)$ will make the multiplicative factor $1+O(1/H)$ and the propagation of error argument go through.

We hope that the above analysis can be applied more broadly in analyzing exploration problems with delayed updates or asynchronous parallelization.

7 Lower bound on switching cost

Theorem 6. *Let $A \ge 4$ and $\mathcal{M}$ be the set of episodic MDPs satisfying the conditions in Section 2. For any RL algorithm $\mathcal{A}$ satisfying $N_{\text{switch}} \le HSA/2$, we have*

$$\sup_{M\in\mathcal{M}} \mathbb{E}_{x_1,M}\left[\sum_{k=1}^{K} V_1^\star(x_1) - V_1^{\pi_k}(x_1)\right] \ge KH/4.$$

[6] Technically it is an upper bound on the regret.

i.e. the worst case regret is linear in K.

Theorem 6 implies that the switching cost of any no-regret algorithm is lower bounded by $\Omega(HSA)$, which is quite intuitive as one would like to play each action at least once on all (h, x). Compared with the lower bound, the switching cost $O(H^3SA\log K)$ we achieve through UCB2 scheduling is at most off by a factor of $O(H^2\log K)$. We believe that the $\log K$ factor is not necessary as there exist algorithms achieving double-log [8] in bandits, and would also like to leave the tightening of the H^2 factor as future work. The proof of Theorem 6 is deferred to Appendix E.

8 Conclusion

In this paper, we take steps toward studying limited adaptivity RL. We propose a notion of local switching cost to account for the adaptivity of RL algorithms. We design a Q-Learning algorithm with infrequent policy switching that achieves $\widetilde{O}(\sqrt{H^{\{4,3\}}SAT})$ regret while switching its policy for at most $O(\log T)$ times. Our algorithm works in the concurrent setting through parallelization and achieves nearly linear speedup and favorable sample complexity. Our proof involves a novel perturbation analysis for exploration algorithms with delayed updates, which could be of broader interest.

There are many interesting future directions, including (1) low switching cost algorithms with tighter regret bounds, most likely via model-based approaches; (2) algorithms with even lower switching cost; (3) investigate the connection to other settings such as off-policy RL.

References

[1] S. Agrawal and R. Jia. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. In *Advances in Neural Information Processing Systems*, pages 1184–1194, 2017.

[2] D. Almirall, S. N. Compton, M. Gunlicks-Stoessel, N. Duan, and S. A. Murphy. Designing a pilot sequential multiple assignment randomized trial for developing an adaptive treatment strategy. *Statistics in medicine*, 31(17):1887–1902, 2012.

[3] D. Almirall, I. Nahum-Shani, N. E. Sherwood, and S. A. Murphy. Introduction to smart designs for the development of adaptive interventions: with application to weight loss research. *Translational behavioral medicine*, 4(3):260–274, 2014.

[4] A. H. Ashouri, W. Killian, J. Cavazos, G. Palermo, and C. Silvano. A survey on compiler autotuning using machine learning. *ACM Computing Surveys (CSUR)*, 51(5):96, 2018.

[5] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[6] M. G. Azar, I. Osband, and R. Munos. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 263–272. JMLR. org, 2017.

[7] R. I. Brafman and M. Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct):213–231, 2002.

[8] N. Cesa-Bianchi, O. Dekel, and O. Shamir. Online learning with switching costs and other adaptive adversaries. In *Advances in Neural Information Processing Systems*, pages 1160–1168, 2013.

[9] C. Dann, L. Li, W. Wei, and E. Brunskill. Policy certificates: Towards accountable reinforcement learning. *arXiv preprint arXiv:1811.03056*, 2018.

[10] J. Duchi, F. Ruan, and C. Yun. Minimax bounds on stochastic batched convex optimization. In *Conference On Learning Theory*, pages 3065–3162, 2018.

[11] Z. Gao, Y. Han, Z. Ren, and Z. Zhou. Batched multi-armed bandits problem. *arXiv preprint arXiv:1904.01763*, 2019.

[12] Z. Guo and E. Brunskill. Concurrent pac rl. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[13] J. P. Hanna, P. S. Thomas, P. Stone, and S. Niekum. Data-efficient policy evaluation through behavior policy search. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1394–1403. JMLR. org, 2017.

[14] T. Jaksch, R. Ortner, and P. Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

[15] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pages 4868–4878, 2018.

[16] M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2-3):209–232, 2002.

[17] S. Krishnan, Z. Yang, K. Goldberg, J. Hellerstein, and I. Stoica. Learning to optimize join queries with deep reinforcement learning. *arXiv preprint arXiv:1808.03196*, 2018.

[18] H. Lei, I. Nahum-Shani, K. Lynch, D. Oslin, and S. A. Murphy. A ”smart” design for building individualized treatment sequences. *Annual review of clinical psychology*, 8:21–48, 2012.

[19] A. Mirhoseini, H. Pham, Q. V. Le, B. Steiner, R. Larsen, Y. Zhou, N. Kumar, M. Norouzi, S. Bengio, and J. Dean. Device placement optimization with reinforcement learning. In *nternational Conference on Machine Learning (ICML-17)*, pages 2430–2439. JMLR. org, 2017.

[20] P. Nguyen, T. Tran, S. Gupta, S. Rana, M. Barnett, and S. Venkatesh. Incomplete conditional density estimation for fast materials discovery. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 549–557. SIAM, 2019.

[21] I. Osband, D. Russo, and B. Van Roy. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pages 3003–3011, 2013.

[22] V. Perchet, P. Rigollet, S. Chassang, E. Snowberg, et al. Batched bandit problems. *The Annals of Statistics*, 44(2):660–681, 2016.

[23] P. Raccuglia, K. C. Elbert, P. D. Adler, C. Falk, M. B. Wenny, A. Mollo, M. Zeller, S. A. Friedler, J. Schrier, and A. J. Norquist. Machine-learning-assisted materials discovery using failed experiments. *Nature*, 533(7601):73, 2016.

[24] G. Theocharous, P. S. Thomas, and M. Ghavamzadeh. Personalized ad recommendation systems for life-time value optimization with guarantees. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

A Proof of Theorem 2

This section is structured as follows. We collect notation in Section A.1 and list some basic properties of the running estimate $\widetilde{Q}$ in Section A.2, establish useful perturbation bounds on $[\widetilde{Q}_h^k - \widetilde{Q}_h^{k'}]_+$ in Section A.3, and present the proof of the main theorem in Section A.4.

A.1 Notation

Let $\widetilde{Q}_h^k(x,a)$ and $Q_h^k(x,a)$ denote the estimates $\widetilde{Q}$ and Q in Algorithm 2 before the k-th episode has started. Note that $\widetilde{Q}_h^1(x,a) = Q_h^1(x,a) \equiv H$.

Define the sequences

$$\alpha_t^0 := \prod_{i=1}^t (1-\alpha_i), \quad \alpha_t^i := \alpha_i \cdot \prod_{\tau=i+1}^t (1-\alpha_\tau).$$

For $t \ge 1$, we have $\alpha_t^0 = 0$ and $\sum_{i=1}^t \alpha_t^i = 1$. For $t = 0$, we have $\alpha_t^0 = 1$.

With the definition of α_t^i in hand, we have the following explicit formula for $\widetilde{Q}_h^k$:

$$\widetilde{Q}_h^k(x,a) = \alpha_t^0 H + \sum_{i=1}^t \alpha_t^i \left(r_h(x,a) + \widetilde{V}_{h+1}^{k_i}(x_{h+1}^{k_i}) + b_i \right),$$

where t is the number of updates on $\widetilde{Q}_h(x,a)$ **prior to** the k-th epoch, and $k_1, \dots, k_t$ are the indices for the epochs. Note that $k = k_{t+1}$ if the algorithm indeed observes x and takes the action a on the h-th step of episode k.

Throughout the proof we let $\ell := \log(SAT/p)$ denote a log factor, where we recall p is the pre-specified tail probability.

A.2 Basics

Lemma A.1 (Properties of α_t^i; Lemma 4.1, [15]). *The following properties hold for the sequence α_t^i:*

(a) $\frac{1}{\sqrt{t}} \le \sum_{i=1}^t \frac{\alpha_t^i}{\sqrt{i}} \le \frac{2}{\sqrt{t}}$ *for every* $t \ge 1$.

(b) $\max_{i \in [t]} \alpha_t^i \le \frac{2H}{t}$ *and* $\sum_{i=1}^t (\alpha_t^i)^2 \le \frac{2H}{t}$ *for every* $t \ge 1$.

(c) $\sum_{t=i}^\infty \alpha_t^i = 1 + \frac{1}{H}$ *for every* $i \ge 1$.

Lemma A.2 ($\widetilde{Q}$ is optimistic and accurate; Lemma 4.2 & 4.3, [15]). *We have for all* $(h,x,a,k) \in [H] \times \mathcal{S} \times \mathcal{A} \times [K]$ *that*

$$\begin{aligned}
&\widetilde{Q}_h^k(x,a) - Q_h^\star(x,a) \\
&= \alpha_t^0 (H - Q_h^\star(x,a)) + \sum_{i=1}^t \alpha_t^i \left(r_h(x,a) + \widetilde{V}_{h+1}^{k_i}(x_{h+1}^{k_i}) - V_{h+1}^\star(x_{h+1}^{k_i}) + \left[\left(\widehat{\mathbb{P}}_h^{k_i} - \mathbb{P}_h \right) V_{h+1}^\star \right](x,a) + b_i \right),
\end{aligned}$$

where $[\hat{\mathbb{P}}_h^{k_i} V_{h+1}](x,a) := V_{h+1}(x_{h+1}^{k_i})$.

Further, with probability at least $1-p$*, choosing* $b_t = c\sqrt{H^3\ell/t}$ *for some absolute constant* $c > 0$*, we have for all* (h,x,a,k) *that*

$$0 \le \widetilde{Q}_h^k(x,a) - Q_h^\star(x,a) \le \alpha_t^0 H + \sum_{i=1}^t \alpha_t^i (\widetilde{V}_{h+1}^{k_i} - V_{h+1}^\star)(x_{h+1}^{k_i}) + \beta_t$$

where $\beta_t := 2\sum_{i=1}^t \alpha_t^i b_i \le 4c\sqrt{H^3\ell/t}$.

Remark. This first part of the Lemma, i.e. the expression of $\widetilde{Q}_h^k - Q_h^\star$ in terms of rewards and value functions, is an aggregated form for the Q functions under the Q-Learning updates, and is independent to the actual exploration policy as well as the bonus.

A.3 Perturbation bound under delayed Q updates

For any $(h,k) \in [H] \times [K]$, let

$$\widetilde{\delta}_h^k := \left(\widetilde{V}_h^k - V_h^{\pi_k}\right)(x_h^k), \quad \widetilde{\phi}_h^k := \left(\widetilde{V}_h^k - V_h^\star\right)(x_h^k)$$

denote the errors of the estimated $\widetilde{V}_h^k$ relative to V^{π_k} and $V^\star$. As $\widetilde{Q}$ is optimistic, the regret can be bounded as

$$\mathrm{Regret}(K) = \sum_{k=1}^K \left[V_1^\star(x_1^k) - V_1^{\pi_k}(x_1^k)\right] \le \sum_{k=1}^K \left[\widetilde{V}_1^k(x_1^k) - V_1^{\pi_k}(x_1^k)\right] = \sum_{k=1}^K \widetilde{\delta}_1^k.$$

The goal of the propagation of error is to related $\sum_{k=1}^K \widetilde{\delta}_h^k$ by $\sum_{k=1}^K \widetilde{\delta}_{h+1}^k$.

We begin by showing that $\widetilde{\delta}_h^k$ is controlled by the max of $\widetilde{Q}_h^k$ and $\widetilde{Q}_h^{k'}$, where $k' = k_{\tau_{\mathrm{last}}(t)+1}$.

Lemma A.3 (Max error under delayed policy update)**.** *We have*

$$\widetilde{\delta}_h^k \le \left(\max\left\{\widetilde{Q}_h^{k'}, \widetilde{Q}_h^k\right\} - Q_h^{\pi_k}\right)(x_h^k, a_h^k) = \left(\widetilde{Q}_h^{k'} - Q_h^{\pi_k} + \left[\widetilde{Q}_h^k - \widetilde{Q}_h^{k'}\right]_+\right)(x_h^k, a_h^k). \tag{2}$$

where $k' = k_{\tau_{\mathrm{last}}(t)+1}$ (which depends on k.) In particular, if $t = \tau_{\mathrm{last}}(t)$, then $k = k'$ and the upper bound reduces to $(\widetilde{Q}_h^{k'} - Q_h^{\pi_k})(x_h^k, a_h^k)$.

Proof. We first show (2). By definition of π_k we have $V_h^{\pi_k}(x_h^k) = Q_h^{\pi_k}(x_h^k, a_h^k)$,

so it suffices to show that

$$\widetilde{V}_h^k(x_h^k) \le \max\left\{\widetilde{Q}_h^k(x_h^k, a_h^k), \widetilde{Q}_h^{k'}(x_h^k, a_h^k)\right\}.$$

Indeed, we have

$$\widetilde{V}_h^k(x_h^k) = \min\left\{H, \max_{a'} \widetilde{Q}_h^k(x_h^k, a')\right\} \le \max_{a'} \widetilde{Q}_h^k(x_h^k, a').$$

On the other hand, a_h^k maximizes $Q_h(x_h^k, \cdot)$. Due to the scheduling of the delayed update, $Q_h(x_h^k, \cdot)$ was set to $\widetilde{Q}_h^{k_{\tau_{\mathrm{last}}(t)}+1}(x_h^k, \cdot)$, and $\widetilde{Q}_h^{\widetilde{k}}(x_h^k, a_h^k)$ was not updated since then before $\widetilde{k} = k' = k_{\tau_{\mathrm{last}}(t)+1}$, so $Q_h(x_h^k, \cdot) = \widetilde{Q}_h^{k'}(x_h^k, \cdot)$.

Now, defining

$$q_{\mathrm{old}}(\cdot) := \widetilde{Q}_h^{k'}(x_h^k, \cdot), \quad q_{\mathrm{new}}(\cdot) := \widetilde{Q}_h^k(x_h^k, \cdot),$$

the vectors q_{old} and q_{new} only differ in the a_h^k-th component (which is the only action taken therefore also the only component that is updated). If q_{new} is also maximized at a_h^k, then we have $\widetilde{V}_h^k(x_h^k) \le q_{\mathrm{new}}(a_h^k)$; otherwise it is maximized at some $a' \ne a_h^k$ and we have

$$\widetilde{V}_h^k(x_h^k) \le q_{\mathrm{new}}(a') = q_{\mathrm{old}}(a') \le \max_a q_{\mathrm{old}}(a) = \widetilde{Q}_h^{k'}(x_h^k, a_h^k).$$

Putting together we get

$$\widetilde{V}_h^k(x_h^k) \le \max\left\{\widetilde{Q}_h^k(x_h^k, a_h^k), \widetilde{Q}_h^{k'}(x_h^k, a_h^k)\right\},$$

which implies (2).

$\square$

Lemma A.3 suggests bounding $\widetilde{\delta}_h^k$ via bounding the "main term" $\widetilde{Q}^{k'} - Q_h^{\pi_k}$ and "perturbation term" $[\widetilde{Q}_h^k - \widetilde{Q}_h^{k'}]_+$ separately. We now establish the bound on the perturbation term.

Lemma A.4 (Perturbation bound on $(\widetilde{Q}_h^k - \widetilde{Q}_h^{k'})_+$)**.** *For any k such that $k > k'$ (so that the perturbation term is non-zero), we have*

$$\left[\widetilde{Q}_h^k - \widetilde{Q}_h^{k'}\right]_+ (x_h^k, a_h^k) \le \beta_t + \sum_{i=\tau_{\rm last}(t)+1}^{t} \alpha_t^i \widetilde{\phi}_{h+1}^{k_i} + \overline{\zeta}_h^k,$$

where

$$\overline{\zeta}_h^k := \left| \sum_{i=\tau_{\rm last}(t)+1}^{t} \alpha_t^i [(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h) V_{h+1}^\star](x_h^k, a_h^k) \right|$$

and w.h.p. we have uniformly over all (h,k) that $\overline{\zeta}_h^k \le C\sqrt{H^3\ell/t}$ for some absolute constant $C > 0$.

Proof. Throughout this proof we will omit the arguments (x_h^k, a_h^k) in $\widetilde{Q}_h$ and r_h as they are clear from the context. By the update formula for $\widetilde{Q}$ in Algorithm 2, we get

$$\widetilde{Q}_h^k = \left(\prod_{i=\tau_{\rm last}(t)+1}^{t} (1-\alpha_i)\right) \widetilde{Q}_h^{k'} + \sum_{i=\tau_{\rm last}(t)+1}^{t} \alpha_t^i \left[r_h(x_h^k, a_h^k) + \widetilde{V}_{h+1}^{k_i}(x_{h+1}^{k_i}) + b_i\right].$$

Subtracting $\widetilde{Q}_h^{k'}$ on both sides (and noting that $\left(\prod_{i=\tau_{\rm last}(t)+1}^{t}(1-\alpha_i)\right) + \sum_{i=\tau_{\rm last}(t)+1}^{t} \alpha_t^i = 1$), we get

$$\widetilde{Q}_h^k - \widetilde{Q}_h^{k'} = \sum_{i=\tau_{\rm last}(t)+1}^{t} \alpha_t^i \underbrace{\left[r_h + \widetilde{V}_{h+1}^{k_i}(x_{h+1}^{k_i}) + b_i - \widetilde{Q}_h^{k'}\right]}_{d_i}. \tag{3}$$

We now upper bound d_i for each i. Adding and subtracting $Q_h^\star$, we obtain

$$\begin{aligned} d_i &= \left(r_h + \widetilde{V}_{h+1}^{k_i}(x_{h+1}^{k_i}) + b_i - Q_h^\star\right) - (\widetilde{Q}_h^{k'} - Q_h^\star) \\ &\overset{\text{(i)}}{=} \widetilde{V}_{h+1}^{k_i}(x_{h+1}^{k_i}) - V^\star(x_{h+1}^{k_i}) + (\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V_{h+1}^\star + b_i - \left(\widetilde{Q}_h^{k'} - Q_h^\star\right) \\ &\overset{\text{(ii)}}{\le} b_i + \widetilde{\phi}_{h+1}^{k_i} + \underbrace{(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V_{h+1}^\star}_{:=\zeta_i}. \end{aligned}$$

where (i) follows from the Bellman optimality equation on $Q_h^\star$, and that $[\widehat{\mathbb{P}}_h^k V_{h+1}^\star](x_h^k, a_h^k) = V_{h+1}^\star(x_{h+1}^k)$ and (ii) follows from the optimistic property of $\widetilde{Q}_h^{k'}$ (from Lemma A.2) and the definition of $\widetilde{\phi}_{h+1}^{k_i}$. Substituting this into (3) gives

$$\left[\widetilde{Q}_h^k - \widetilde{Q}_h^{k'}\right]_+ \le \left[\sum_{i=\tau_{\rm last}(t)+1}^{t} \alpha_t^i \left(b_i + \widetilde{\phi}_{h+1}^{k_i} + \zeta_i\right)\right]_+ \le \beta_t + \sum_{i=\tau_{\rm last}(t)+1}^{t} \alpha_t^i \widetilde{\phi}_{h+1}^{k_i} + \underbrace{\left|\sum_{i=\tau_{\rm last}(t)+1}^{t} \alpha_t^i \zeta_i\right|}_{\overline{\zeta}_h^k}.$$

Finally, note that ζ_i is a martingale difference sequence, so we can apply the Azuma-Hoeffding inequality to get that

$$\bar{\zeta}_h^k \le c\sqrt{\sum_{i=\tau_{\text{last}}(t)+1}^{t} (\alpha_t^i)^2 H^2 \ell} \overset{\text{(i)}}{\le} c\sqrt{\frac{2H}{t}\cdot H^2\ell} = C\sqrt{\frac{H^3\ell}{t}}$$

uniformly over (h, k), where (i) follows from Lemma A.1(b). $\square$

A.4 Proof of Theorem 2

Proof of the main theorem is done through combining the perturbation bound and the "main term", and showing that the propagation of error argument still goes through.

Lemma A.5 (Error accumulation under delayed update)**.** *Suppose we choose* $\eta = \frac{1}{2H(H+1)}$ *and* $r_\star = \left\lceil \frac{\log(10H^2)}{\log(1+\eta)} \right\rceil$ *for the triggering sequence* (1) *then we have for all* i *that*

$$\sum_{t:t\ge i, \tau_{\text{last}}(t)\le i-1} \alpha_t^i + \sum_{t:\tau_{\text{last}}(t)\ge i} \alpha^i_{\tau_{\text{last}}(t)} \le 1 + 3/H.$$

Proof. Let $\widetilde{S}_i$ denote the above sum. We compare $\widetilde{S}_i$ with

$$S_i := \sum_{t=i}^{\infty} \alpha_t^i = 1 + \frac{1}{H},$$

where the last equality follows from Lemma A.1(c).

Let us consider $\widetilde{S}_i - S_i$ by looking at the difference of the individual terms for each $t \ge i$. When taking the difference, the term $\sum_{t:t\ge i,\tau_{\text{last}}(t)\le i-1} \alpha_t^i$ will vanish, and all terms in $\sum_{t:\tau_{\text{last}}(t)\ge i} \alpha^i_{\tau_{\text{last}}(t)}$ will vanish if $\tau_{\text{last}}(t) = t$. By the design of the triggering sequence $\{t_n\}$, we know that this happens for all $t \le \tau(r_\star)$, so we have

$$\widetilde{S}_i - S_i = \sum_{t:\tau_{\text{last}}(t)\ge i; t>\tau(r_\star)} \alpha^i_{\tau_{\text{last}}(t)} - \alpha_t^i.$$

Let $r(i) = \min\{r : \tau(r) \ge i\}$, then the above can be rewritten as

$$\widetilde{S}_i - S_i = \sum_{r\ge \max\{r_\star, r(i)\}} \sum_{t=\tau(r)}^{\tau(r+1)-1} \alpha^i_{\tau(r)} - \alpha_t^i.$$

For each t (and associated $r \ge r_\star$), we have the bound

$$\begin{aligned}
&\alpha^i_{\tau(r)} - \alpha_t^i = \alpha_t^i\left[\prod_{j=\tau(r)+1}^{t}(1-\alpha_j)^{-1} - 1\right] = \alpha_t^i\left[\prod_{j=\tau(r)+1}^{t}\left(1 - \frac{H+1}{H+j}\right)^{-1} - 1\right] \\
&= \alpha_t^i\left[\prod_{j=\tau(r)+1}^{t}\left(1 + \frac{H+1}{j-1}\right) - 1\right] \le \alpha_t^i\left[\left(1 + \frac{H+1}{\tau(r)}\right)^{t-\tau(r)} - 1\right] \\
&\le \alpha_t^i\left[\left(1 + \frac{H+1}{\tau(r)}\right)^{\tau(r+1)-\tau(r)-1} - 1\right] \overset{\text{(i)}}{\le} \alpha_t^i\left[\left(1 + \frac{H+1}{\tau(r)}\right)^{\eta\tau(r)} - 1\right] \\
&\overset{\text{(ii)}}{\le} \alpha_t^i\left[e^{\eta(H+1)} - 1\right] \le \alpha_t^i \cdot 2\eta(H+1).
\end{aligned}$$

In the above, (i) holds as we have

$$\tau(r+1)-1-\tau(r)=\left\lceil(1+\eta)^{r+1}\right\rceil-1-\lceil(1+\eta)^r\rceil\le(1+\eta)^{r+1}-(1+\eta)^r\le\eta\tau(r),$$

and (ii) holds whenever $\eta(H+1)\le 1/2$. Choosing

$$\eta=\frac{1}{2H(H+1)}\quad\text{and}\quad r_\star=\left\lceil\frac{\log(10H^2)}{\log(1+\eta)}\right\rceil\le 8H^2\log(10H^2),$$

the above requirement will be satisfied. Therefore we have

$$\widetilde{S}_i-S_i\le 2\eta(H+1)\sum_{r\ge\max\{r_\star,r(i)\}}\sum_{t=\tau(r)}^{\tau(r+1)-1}\alpha_t^i\le 2\eta(H+1)\sum_{t=i}^{\infty}\alpha_t^i=\frac{1}{H}S_i,$$

and thus

$$\widetilde{S}_i\le\left(1+\frac{1}{H}\right)S_i\le 1+\frac{3}{H}.$$

□

We are now in position to prove the main theorem.

Theorem 2 (Q-learning with UCB2H, restated)**.** *Choosing* $\eta=\frac{1}{2H(H+1)}$ *and* $r_\star=\left\lceil\frac{\log(10H^2)}{\log(1+\eta)}\right\rceil$, *with probability at least* $1-p$*, the regret of Algorithm 2 is bounded by* $O(\sqrt{H^4SAT\ell})$*, where* $\ell:=\log(SAT/p)$ *is a log factor. Further, the local switching cost is bounded as* $N_{\text{switch}}\le O(H^3SA\log(K/A))$.

Proof of Theorem 2 The proof consists of two parts: upper bounding the regret, and upper bounding the local switching cost.

Part I: Regret bound By Lemma A.3, we have

$$\widetilde{\delta}_h^k\le\left(\widetilde{Q}_h^{k'}-Q_h^{\pi_k}+\left[\widetilde{Q}_h^k-\widetilde{Q}_h^{k'}\right]_+\right)(x_h^k,a_h^k).$$

Applying Lemma A.2 with the $k'=k_{\tau_{\text{last}}(t)+1}$-th episode (so that there are $\tau_{\text{last}}(t)$ visitations to (x_h^k,a_h^k) prior to the k'-th episode), we have the bound

$$\begin{aligned}&\left(\widetilde{Q}_h^{k'}-Q_h^{\pi_k}\right)(x_h^k,a_h^k)\le\left(\widetilde{Q}_h^{k'}-Q_h^\star\right)(x_h^k,a_h^k)+\left(Q_h^\star-Q_h^{\pi_k}\right)(x_h^k,a_h^k)\\&\le\alpha^0_{\tau_{\text{last}}(t)}H+\sum_{i=1}^{\tau_{\text{last}}(t)}\alpha^i_{\tau_{\text{last}}(t)}\widetilde{\phi}_{h+1}^{k_i}+\beta_{\tau_{\text{last}}(t)}-\widetilde{\phi}_{h+1}^k+\widetilde{\delta}_{h+1}^k+\xi_{h+1}^k,\end{aligned}\tag{4}$$

where we recall that $\beta_t=2\sum_i\alpha_t^ib_i=\Theta(\sqrt{H^3\ell/t})$ and $\xi_{h+1}^k:=[(\widehat{\mathbb{P}}_h^{k_i}-\mathbb{P}_h)(V_{h+1}^\star-V_{h+1}^{\pi_k}](x_h^k,a_h^k)$. By Lemma A.4, the perturbation term $[\widetilde{Q}_h^k-\widetilde{Q}_h^{k'}]_+$ can be bounded as

$$[\widetilde{Q}_h^k-\widetilde{Q}_h^{k'}]_+(x_h^k,a_h^k)\le\beta_t+\sum_{i=\tau_{\text{last}}(t)+1}^{t}\alpha_t^i\widetilde{\phi}_{h+1}^{k_i}+C\sqrt{\frac{H^3\ell}{t}}.\tag{5}$$

We now study the effect of adding (5) onto (4). The term $C\sqrt{H^3\ell/t}$ in (5) and $\beta_{\tau_{\rm last}(t)}$ in (4) can be both absorbed into β_t (as $\beta_t \geq 2\sqrt{H^3\ell/t}$ and $\beta_{\tau_{\rm last}(t)} \leq \sqrt{1+\eta}\beta_t$), so these together is bounded by $C'\beta_t$ where C' is an absolute constant.

Adding (5) onto (4), we obtain

$$\widetilde{\delta}_h^k \leq \underbrace{\alpha^0_{\tau_{\rm last}(t)} H}_{\rm I} + \underbrace{\sum_{i=1}^{\tau_{\rm last}(t)} \alpha^i_{\tau_{\rm last}(t)} \widetilde{\phi}^{k_i}_{h+1} + \sum_{i=\tau_{\rm last}(t)+1}^{t} \alpha^i_t \widetilde{\phi}^{k_i}_{h+1}}_{\rm II} + C'\beta_t - \widetilde{\phi}^k_{h+1} + \widetilde{\delta}^k_{h+1} + \xi^k_{h+1}.$$

We now sum the above bound over $k \in [K]$. For term I, it equals H only when $\tau_{\rm last}(t) = 0$, which happens only if $t = 0$, so the sum over k is upper bounded by SAH.

For term II, we consider the coefficient in front of $\widetilde{\phi}^{k'}_{h+1}$ for each $k' \in [K]$ when summing over k. Let n_h^k denote the number of visitations to (x_h^k, a_h^k) prior to the k-th episode. For each k', $\widetilde{\phi}^{k'}_{h+1}$ is counted if $i = n_h^{k'}$ and $(x_h^k, a_h^k) = (x_h^{k'}, a_h^{k'})$. We use t to denote n_h^k, then an $\alpha^{n_h^{k'}}_{\tau_{\rm last}(t)}$ appears if $\tau_{\rm last}(t) \geq n_h^{k'}$, and an $\alpha_t^{n_h^{k'}}$ appears if $\tau_{\rm last}(t) + 1 \leq n_h^{k'} \leq t$. So the total coefficient in front of $\widetilde{\phi}^{k'}_{h+1}$ is at most

$$\sum_{t: t \geq n_h^{k'}, \tau_{\rm last}(t) \leq n_h^{k'} - 1} \alpha_t^{n_h^{k'}} + \sum_{t: \tau_{\rm last}(t) \geq n_h^{k'}} \alpha^{n_h^{k'}}_{\tau_{\rm last}(t)},$$

for each $k' \in [K]$. Choosing $\eta = \frac{1}{2H(H+1)}$ and $r_\star = \left\lceil \frac{\log(10H^2)}{\log(1+\eta)} \right\rceil$, applying Lemma A.5, the above is upper bounded by $1 + 3/H$.

For the remaining terms, we can adapt the proof of Theorem 1 in [15] and obtain a propagation of error inequality, and deduce (as $(1+3/H)^H = O(1)$) that the regret is bounded by $O(\sqrt{H^4SAT\ell})$. This concludes the proof.

Part II: Bound on local switching cost For each $(h, x) \in [H] \times \mathcal{S}$ and each action $a \in \mathcal{A} = [A]$, either it is in stage I, which induces a switching cost of at most $\tau(r_\star)$, or it is in stage II, which according to the triggering sequence induces a switching cost of

$$\tau(r_\star) + r_a - r_\star \leq \tau(r_\star) + r_a,$$

where r_a is the final index for action a satisfying

$$\sum_{a=1}^{A} \lceil (1+\eta)^{r_a} \rceil \leq K + H,$$

(define $r_a = 0$ if action a has not reached the second stage.) Applying Jensen's inequality gives that

$$\sum_{a=1}^{A} r_a \leq \frac{A \log((K+H)/A)}{\log(1+\eta)} = O\left(H^2 A \log(K/A)\right)$$

So the switching cost for (h, x) can be bounded as

$$\begin{aligned} & A\tau(r_\star) + \sum_{a=1}^{A} r_a \\ \leq\ & A \left\lceil (1+\eta)^{r_\star} \right\rceil + O\left(H^2 A \log(K/A)\right) \leq A \left\lceil (1+\eta) \cdot 10H^2 \right\rceil + O\left(H^2 A \log(K/A)\right) \\ \leq\ & 20H^2 A + O\left(H^2 A \log(K/A)\right) = O\left(H^2 A \log(K/A)\right). \end{aligned}$$

Multiplying the above by HS (the number of (h,x) pairs) gives the desired bound. $\square$

B Q-learning with UCB2-Bernstein exploration

B.1 Algorithm description

We present the algorithm, Q-Learning with UCB2-Bernstein (UCB2B) exploration, in Algorithm 3 below.

Algorithm 3 Q-learning with UCB2-Bernstein (UCB2B) Exploration

---

input Parameter $\eta \in (0,1)$, $r_\star \in \mathbb{Z}_{>0}$, and $c > 0$.

 Initialize: $\widetilde{Q}_h(x,a) \leftarrow H$, $Q_h \leftarrow \widetilde{Q}_h$, $N_h(x,a) \leftarrow 0$ for all $(x,a,h) \in \mathcal{S} \times \mathcal{A} \times [H]$.

 for episode $k = 1, \dots, K$ **do**

  Receive x_1.

  for step $h = 1, \dots, H$ **do**

   Take action $a_h \leftarrow \arg\max_{a'} Q_h(x_h, a')$, and observe x_{h+1}.

   $t = N_h(x_h,a_h) \leftarrow N_h(x_h,a_h) + 1$.

   $\mu_h(x_h,a_h) \leftarrow \mu_h(s_h,a_h) + V_{h+1}(x_{h+1})$.

   $\sigma_h(x_h,a_h) \leftarrow \sigma_h(x_h,a_h) + (V_{h+1}(x_{h+1}))^2$.

   $W_t(x_h,a_h,h) = \frac{1}{t}\left(\sigma_h(x_h,a_h) - (\mu_h(x_h,a_h))^2\right)$.

   $\beta_t(x_h,a_h,h) \leftarrow \min\left\{c_1\left(\sqrt{\frac{H}{t}(W_t(x_h,a_h,h)+H)\ell} + \frac{\sqrt{H^7SA}\cdot\ell}{t}\right), c_2\sqrt{\frac{H^3\ell}{t}}\right\}$.

   $b_t \leftarrow \frac{\beta_t(x_h,a_h,h) - (1-\alpha_t)\beta_{t-1}(x_h,a_h,h)}{2\alpha_t}$ (Bernstein-type bonus).

   $\widetilde{Q}_h(x_h,a_h) \leftarrow (1-\alpha_t)\widetilde{Q}_h(x_h,a_h) + \alpha_t[r_h(x_h,a_h) + \widetilde{V}_{h+1}(x_{h+1}) + b_t]$.

   $\widetilde{V}_h(x_h) \leftarrow \min\left\{H, \max_{a' \in \mathcal{A}} \widetilde{Q}_h(x_h,a')\right\}$.

   if $t \in \{t_n\}_{n\geq 1}$ (where t_n is defined in (1)) **then**

    (Update policy) $Q_h(x_h,\cdot) \leftarrow \widetilde{Q}_h(x_h,\cdot)$.

   end if

  end for

 end for

B.2 Proof of Theorem 3

We first present the analogs of Lemmas that we used in the proof of Theorem 2.

Lemma B.1 ($\widetilde{Q}$ is optimistic and accurate for the Bernstein case; Lemma C.1 & C.4, [15])**.** *We have for all $(h,x,a,k) \in [H] \times \mathcal{S} \times \mathcal{A} \times [K]$ that*

$$\begin{aligned}
&\widetilde{Q}_h^k(x,a) - Q_h^\star(x,a) \\
=&\alpha_t^0(H - Q_h^\star(x,a)) + \\
&\sum_{i=1}^t \alpha_t^i\left(r_h(x,a) + \widetilde{V}_{h+1}^{k_i}(x_{h+1}^{k_i}) - V_{h+1}^\star(x_{h+1}^{k_i}) + \left[\left(\widehat{\mathbb{P}}_h^{k_i} - \mathbb{P}_h\right)V_{h+1}^\star\right](x,a) + b_i\right),
\end{aligned}$$

where $[\widehat{\mathbb{P}}_h^{k_i} V_{h+1}](x,a) := V_{h+1}(x_{h+1}^{k_i})$.

Further, with probability at least $1-p$, under the choice of b_t and β_t in Algorithm 3, we have for all (h,x,a,k) that

$$0 \le \widetilde{Q}_h^k(x,a) - Q_h^\star(x,a) \le \alpha_t^0 H + \sum_{i=1}^t \alpha_t^i (\widetilde{V}_{h+1}^{k_i} - V_{h+1}^\star)(x_{h+1}^{k_i}) + \beta_t.$$

The following Lemma is the analog of Lemma A.3 in the Bernstein case.

Lemma B.2 (Max error under delayed policy update)**.** *We have*

$$\widetilde{\delta}_h^k \le \left(\max\left\{\widetilde{Q}_h^{k'}, \widetilde{Q}_h^k\right\} - Q_h^{\pi_k}\right)(x_h^k, a_h^k) = \left(\widetilde{Q}_h^{k'} - Q_h^{\pi_k} + \left[\widetilde{Q}_h^k - \widetilde{Q}_h^{k'}\right]_+\right)(x_h^k, a_h^k).$$

where $k' = k_{\tau_{\mathrm{last}}(t)+1}$ (which depends on k.) In particular, if $t = \tau_{\mathrm{last}}(t)$, then $k = k'$ and the upper bound reduces to $(\widetilde{Q}_h^{k'} - Q_h^{\pi_k})(x_h^k, a_h^k)$.

The proof of Lemma B.2 can be adapted from the proof of Lemma A.3. The following Lemma is the analog of Lemma A.4 in the Bernstein case.

Lemma B.3 (Perturbation bound on $(\widetilde{Q}_h^k - \widetilde{Q}_h^{k'})_+$)**.** *For any k such that $k > k'$ (so that the perturbation term is non-zero), we have*

$$\left[\widetilde{Q}_h^k - \widetilde{Q}_h^{k'}\right]_+ (x_h^k, a_h^k) \le \beta_t + \sum_{i=\tau_{\mathrm{last}}(t)+1}^t \alpha_t^i \widetilde{\phi}_{h+1}^{k_i} + \overline{\zeta}_h^k,$$

where

$$\overline{\zeta}_h^k := \left| \sum_{i=\tau_{\mathrm{last}}(t)+1}^t \alpha_t^i [(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h) V_{h+1}^\star](x_h^k, a_h^k) \right|.$$

The proof of Lemma B.3 can be adapted from the proof of Lemma A.4, but we used a finer bound on the summation $\overline{\zeta}_h^k$ over $k \in [K]$ in the proof of Theorem 3.

Lemma B.4 (Variance is bounded and W_t is accurate; Lemma C.5 & C.6, [15])**.** *There exists an absolute constant c, such that*

$$\sum_{k=1}^K \sum_{h=1}^H \mathbb{V}_h V_{h+1}^{\pi_k}(x_h^k, a_h^k) \le c(HT + H^3\ell),$$

w.p. at least $(1-p)$.

Further, w.p. at least $(1-4p)$, there exists an absolute constant $c > 0$ such that, letting $(x,a) = (x_h^k, a_h^k)$ and $t = n_h^k = N_h^k(x,a)$, we have

$$W_t(x,a,h) \le \mathbb{V}_h V_{h+1}^{\pi_k}(x,a) + 2H(\delta_{h+1}^k + \xi_{h+1}^k) + c\left(\frac{SA\sqrt{H^7\ell}}{t} + \sqrt{\frac{SAH^7\ell}{t}}\right)$$

for all $(k,h) \in [K] \times [H]$, where the variance operator $\mathbb{V}_h$ is defined by

$$[\mathbb{V}_h V_{h+1}](x,a) := \mathit{Var}_{x' \sim \mathbb{P}_h(\cdot|x,a)}(V_{h+1}(x')) = \mathbb{E}_{x' \sim \mathbb{P}_h(\cdot|x,a)}\left[V_{h+1}(x') - [\mathbb{P}_h V_{h+1}](x,a)\right]^2.$$

Now, it is ready to present the proof of Theorem 3.

Theorem 3 (Q-learning with UCB2B, restated)**.** *Choosing* $\eta = \frac{1}{2H(H+1)}$ *and* $r_\star = \left\lceil \frac{\log(10H^2)}{\log(1+\eta)} \right\rceil$, *with probability at least* $1-p$, *the regret of Algorithm 3 is bounded by* $O(\sqrt{H^3SAT\ell^2} + \sqrt{S^3A^3H^9\ell^4})$, *where* $\ell := \log(SAT/p)$ *is a log factor. Further, the local switching cost is bounded as* $N_{\rm switch} \le O(H^3SA\log(K/A))$.

Proof of Theorem 3

By Lemma B.2, we have

$$\widetilde{\delta}_h^k \le \left(\widetilde{Q}_h^{k'} - Q_h^{\pi_k} + \left[\widetilde{Q}_h^k - \widetilde{Q}_h^{k'}\right]_+\right)(x_h^k, a_h^k).$$

Applying Lemma B.1 with the $k' = k_{\tau_{\rm last}(t)+1}$-th episode (so that there are $\tau_{\rm last}(t)$ visitations to (x_h^k, a_h^k) prior to the k'-th episode), we have the bound

$$\begin{aligned}
&\left(\widetilde{Q}_h^{k'} - Q_h^{\pi_k}\right)(x_h^k, a_h^k) \le \left(\widetilde{Q}_h^{k'} - Q_h^\star\right)(x_h^k, a_h^k) + \left(Q_h^\star - Q_h^{\pi_k}\right)(x_h^k, a_h^k) \\
&\le \alpha^0_{\tau_{\rm last}(t)} H + \sum_{i=1}^{\tau_{\rm last}(t)} \alpha^i_{\tau_{\rm last}(t)} \widetilde{\phi}_{h+1}^{k_i} + \beta_{\tau_{\rm last}(t)} - \widetilde{\phi}_{h+1}^k + \widetilde{\delta}_{h+1}^k + \xi_{h+1}^k.
\end{aligned} \tag{6}$$

By Lemma B.3, the perturbation term $[\widetilde{Q}_h^k - \widetilde{Q}_h^{k'}]_+$ can be bounded as

$$[\widetilde{Q}_h^k - \widetilde{Q}_h^{k'}]_+(x_h^k, a_h^k) \le \beta_t + \sum_{i=\tau_{\rm last}(t)+1}^{t} \alpha_t^i \widetilde{\phi}_{h+1}^{k_i} + \overline{\zeta}_h^k. \tag{7}$$

Thus, adding (7) onto (6), we obtain

$$\begin{aligned}
\widetilde{\delta}_h^k \le& \underbrace{\alpha^0_{\tau_{\rm last}(t)} H}_{\rm I} + \underbrace{\sum_{i=1}^{\tau_{\rm last}(t)} \alpha^i_{\tau_{\rm last}(t)} \widetilde{\phi}_{h+1}^{k_i} + \sum_{i=\tau_{\rm last}(t)+1}^{t} \alpha_t^i \widetilde{\phi}_{h+1}^{k_i}}_{\rm II} + \underbrace{\overline{\zeta}_h^k}_{\rm III} \\
&+ \underbrace{\beta_{\tau_{\rm last}(t)}}_{\rm IV} + \underbrace{\xi_{h+1}^k}_{V} - \widetilde{\phi}_{h+1}^k + \widetilde{\delta}_{h+1}^k + \beta_t.
\end{aligned}$$

We now sum the above bound over $k \in [K]$ and $h \in [H]$. For term I, it equals H only when $\tau_{\rm last}(t) = 0$, which happens only if $t = 0$, so the sum over k is upper bounded by SAH.

For term II, we follow the same argument in the proof of Theorem 2 and obtain:

$$\sum_{k=1}^K \left(\sum_{i=1}^{\tau_{\rm last}(t)} \alpha^i_{\tau_{\rm last}(t)} \widetilde{\phi}_{h+1}^{k_i} + \sum_{i=\tau_{\rm last}(t)+1}^{t} \alpha_t^i \widetilde{\phi}_{h+1}^{k_i}\right) \le \left(1 + \frac{3}{H}\right) \sum_{k=1}^K \widetilde{\phi}_{h+1}^k$$

For term III, we first apply the Azuma-Hoeffding inequality to get that

$$\overline{\zeta}_h^k \le c\sqrt{\sum_{i=\tau_{\rm last}(t)+1}^{t} (\alpha_t^i)^2 H^2 \ell}$$

uniformly over (h,k), then we sum it the above over $k\in[K]$, and then we obtain

$$\begin{aligned}
\sum_{k=1}^{K}\bar{\zeta}_h^k \leq & cH\sqrt{\ell}\sum_{k=1}^{K}\sqrt{\sum_{i=\tau_{\text{last}}(t)+1}^{t}(\alpha_t^i)^2}\leq cH\sqrt{\ell}\sum_{k=1}^{K}\sqrt{\sum_{i=\left\lceil\frac{n_h^k}{1+\eta}\right\rceil}^{n_h^k}\left(\alpha_{n_h^k}^i\right)^2}\\
\leq & cH\sqrt{\ell}\sum_{k=1}^{K}\sqrt{\left(n_h^k-\left\lceil\frac{n_h^k}{1+\eta}\right\rceil\right)\left(\max_{i\in[n_h^k]}\alpha_{n_h^k}^i\right)^2}\\
\leq & cH\sqrt{\ell}\sum_{k=1}^{K}\sqrt{\eta n_h^k\frac{4H^2}{(n_h^k)^2}}\\
\leq & cH\sqrt{\ell}\sum_{k=1}^{K}\sqrt{\frac{1}{n_h^k}}\overset{\text{(i)}}{=}cH\sqrt{\ell}\sum_{x,a}\sum_{n=1}^{N_h^k(s,a)}\sqrt{\frac{1}{n}}\overset{\text{(ii)}}{\leq}cH\sqrt{SAK\ell},
\end{aligned}\tag{8}$$

where (i) follows the fact $\sum_{s,a}N_h^K(x,a)=K$, and (ii) follows the property that the LHS of (ii) is maximized when $N_h^K(x,a)=K/SA$ for all x,a.

For term IV, we have

$$\sum_{k=1}^{K}\sum_{h=1}^{H}\beta_{\tau_{\text{last}}(n_h^k)}\leq c_1\sum_{k=1}^{K}\sum_{h=1}^{H}\left(\sqrt{\frac{H}{\tau_{\text{last}}(n_h^k)}(W_{\tau_{\text{last}}(n_h^k)}(x,a,h)+H)\ell}+\frac{\sqrt{H^7SA}\cdot\ell}{\tau_{\text{last}}(n_h^k)}\right)\tag{9}$$

by our choice of β_t in Algorithm 3. We first upper bound summation the $W_{\tau_{\text{last}}(n_h^k)}(x,a,h)$ term as follows

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{h=1}^{H}W_{\tau_{\text{last}}(n_h^k)}(x,a,h)\\
\overset{\text{(i)}}{\leq}&\sum_{k=1}^{K}\sum_{h=1}^{H}\left[\mathbb{V}_hV_{h+1}^{\pi_k}(x,a)+2H(\delta_{h+1}^k+\xi_{h+1}^k)+c\left(\frac{SA\sqrt{H^7\ell}}{\tau_{\text{last}}(n_h^k)}+\sqrt{\frac{SAH^7\ell}{\tau_{\text{last}}(n_h^k)}}\right)\right]\\
\overset{\text{(ii)}}{\leq}&\sum_{k=1}^{K}\sum_{h=1}^{H}\left[\mathbb{V}_hV_{h+1}^{\pi_k}(x,a)+2H(\delta_{h+1}^k+\xi_{h+1}^k)+c(1+\eta)\left(\frac{SA\sqrt{H^7\ell}}{n_h^k}+\sqrt{\frac{SAH^7\ell}{n_h^k}}\right)\right]\\
\overset{\text{(iii)}}{\leq}&\sum_{k=1}^{K}\sum_{h=1}^{H}\left[\mathbb{V}_hV_{h+1}^{\pi_k}(x,a)+2H(\delta_{h+1}^k+\xi_{h+1}^k)\right]+c(1+\eta)\left(S^2A^2\sqrt{H^9\ell^3}+SA\sqrt{H^8T\ell}\right)\\
\overset{\text{(iv)}}{\leq}&2H\sum_{k=1}^{K}\sum_{h=1}^{H}(\delta_{h+1}^k+\xi_{h+1}^k)+c'\left(HT+H^3\ell+S^2A^2\sqrt{H^9\ell^3}+SA\sqrt{H^8T\ell}\right),
\end{aligned}\tag{10}$$

where inequalities (i) and (iv) follow from Lemma B.4, inequality (ii) follows from $\tau_{\text{last}}(n_h^k)\geq n_h^k/(1+\eta)$, and inequality (iii) uses the properties that $\sum_{k=1}^{K}(n_h^k)^{-1}$ and $\sum_{k=1}^{K}(n_h^k)^{-1/2}$ are maximized when $N_h^K(x,a)=K/SA$ for all x,a (similar to (8)).

We now consider the first term in (10). By the Azuma-Hoeffding inequality, we have

$$\left|\sum_{h'=h}^{H}\sum_{k=1}^{K}\xi_{h'+1}^k\right|\leq\left|\sum_{h'=h}^{H}\sum_{k=1}^{K}[(\widehat{\mathbb{P}}_{h'}^{k_i}-\mathbb{P}_h)(V_{h'+1}^{\star}-V_{h'+1}^{\pi_k})](x_{h'}^k,a_{h'}^k)\right|\leq O(H\sqrt{T\ell}),\tag{11}$$

w.p. $1-p$ for all $h \in [H]$. Recall $\beta_t(x,a,h) \le c\sqrt{H^3\ell/t}$, we can simply obtain

$$\sum_{k=1}^{K} \delta_h^k \le O(\sqrt{H^4 SAT\ell}), \tag{12}$$

for all $h \in [H]$ by adapting the proof of Theorem 2. Then, using (11) and (12), we obtain the upper bound of the summation of $W_{\tau_{\text{last}}(n_h^k)}(x,a,h)$ term for $h \in [H]$ and $k \in [K]$

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{h=1}^{H} W_{\tau_{\text{last}}(n_h^k)}(x,a,h) \\
&2H\sum_{k=1}^{K}\sum_{h=1}^{H}(\delta_{h+1}^k + \xi_{h+1}^k) + c'\left(HT + H^3\ell + S^2A^2\sqrt{H^9\ell^3} + SA\sqrt{H^8T\ell}\right) \\
\le& O\left(HT + S^2A^2H^7\ell + S^2A^2\sqrt{H^9\ell^3}\right).
\end{aligned} \tag{13}$$

Now it is ready to upper bounded the summation of the first term in (9),

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{h=1}^{H}\sqrt{\frac{H}{\tau_{\text{last}}(n_h^k)}(W_{\tau_{\text{last}}(n_h^k)}(x,a,h)+H)\ell} \\
\overset{\text{(i)}}{\le}& \sqrt{\left(\sum_{k=1}^{K}\sum_{h=1}^{H}(W_{\tau_{\text{last}}(n_h^k)}(x,a,h)+H)\right)\left(\sum_{k=1}^{K}\sum_{h=1}^{H}\frac{H}{\tau_{\text{last}}(n_h^k)}\right)\ell} \\
\overset{\text{(ii)}}{\le}& (1+\eta)\sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H}W_{\tau_{\text{last}}(n_h^k)}(x,a,h)}\cdot\sqrt{H^2SA\ell^2} + (1+\eta)\sqrt{H^3SAT\ell^2} \\
\overset{\text{(iii)}}{\le}& O(\sqrt{H^3SAT\ell^2})
\end{aligned} \tag{14}$$

where inequality (i) follows from the Cauchy–Schwarz inequality, inequality (ii) follows from the facts that $\tau_{\text{last}}(n_h^k) \ge n_h^k/(1+\eta)$ and $\sum_{k=1}^{K}(n_h^k)^{-1}$ is maximized when $N_h^K(x,a) = K/SA$ for all x, a, and inequality (iii) follows from (13).

The summation of the second term in (9) can be upper bounded by

$$\sum_{k=1}^{K}\sum_{h=1}^{H}\frac{\sqrt{H^7SA}\cdot\ell}{\tau_{\text{last}}(n_h^k)} \le \sum_{k=1}^{K}\sum_{h=1}^{H}\frac{(1+\eta)\sqrt{H^7SA}\cdot\ell}{n_h^k} \le (1+\eta)\sqrt{H^9S^3A^3\ell^4}, \tag{15}$$

by following $\tau_{\text{last}}(n_h^k) \ge n_h^k/(1+\eta)$ and $1 + 1/2 + 1/3 + \cdots \le \ell$.

Putting (9), (14), and (15) together, we have

$$\sum_{k=1}^{K}\sum_{h=1}^{H}\beta_{\tau_{\text{last}}(n_h^k)} \le O\left(\sqrt{H^3SAT\ell^2} + \sqrt{S^3A^3H^9\ell^4}\right).$$

For the remaining terms, we can adapt the proof of Theorem 2 in [15] and obtain a propagation of error inequality. Thus, we deduce that the regret is bounded by $O(\sqrt{H^3SAT\ell^2} + \sqrt{S^3A^3H^9\ell^4})$. The bound on local switching cost can be adapted from the proof of Theorem 2. This concludes the proof. $\square$

C Proof of Corollary 4

Consider first Q-Learning with UCB2H exploration. By Theorem 2, we know that the regret is bounded by $\widetilde{O}(\sqrt{H^4SAT})$ with high probability, that is, we have

$$\sum_{k=1}^{K} V_1^\star(x_1) - V_1^{\pi_k}(x_1) \le \widetilde{O}(\sqrt{H^4SAT}).$$

Now, define a stochastic policy $\widehat{\pi}$ as

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

By definition we have

$$\mathbb{E}\left[V_1^\star(x_1) - V_1^{\widehat{\pi}}(x_1)\right] = \frac{1}{K}\sum_{k=1}^{K}\left[V_1^\star(x_1) - V_1^{\pi_k}(x_1)\right] \le \widetilde{O}\left(\frac{\sqrt{H^4SAT}}{K}\right) = \widetilde{O}\left(\sqrt{\frac{H^5SA}{K}}\right).$$

So by the Markov inequality, we have with high probability that

$$V_1^\star(x_1) - V_1^{\widehat{\pi}}(x_1) \le \widetilde{O}\left(\sqrt{\frac{H^5SA}{K}}\right).$$

Taking $K = \widetilde{O}(H^5SA/\varepsilon^2)$ bounds the above by ε.

For Q-Learning with UCB2B exploration, the regret bound is $\widetilde{O}(\sqrt{H^3SAT})$. A similar argument as above gives that $K = \widetilde{O}(H^4SA/\varepsilon^2)$ episodes guarantees an ε near-optimal policy with high probability. $\square$

D Proof of Theorem 5

We first present the concurrent version of low-switching cost Q-learning with {UCB2H, UCB2B} exploration.

Algorithm description At a high level, our algorithm is a very intuitive parallelization of the vanilla version – we "parallelize as much as you can" until the next scheduled switch.

More concretely, suppose we the policy Q_h has been switched $(t-1)$ times and we have a new policy yet to be executed. For each (h,x), let $a_t(x,h) = \arg\max_{a'\in\mathcal{A}} Q_h(x,a')$ denote the current policy, and $\ell_t(h,x)$ denote the number of times $a_t(x,h)$ is scheduled to be taken (according to the triggering sequence) *before the next switch*. Define

$$L_t := \min_{(h,x)\in[H]\times\mathcal{S}} \ell_t(h,x). \tag{16}$$

In other words, the policy is guaranteed to not switch under the triggering sequence until L_t episodes have been played.

Now, if $1 \le L_t \le M$, then we parallelize the execution in the first L_t machines and finish in one round; if $L_t > M$, we use all M machines for $r_t = \lceil L_t/M \rceil$ rounds. After the execution we update the policy accordingly and set $t \leftarrow t+1$. The full algorithm is presented in Algorithm 4.

Algorithm 4 Concurrent Q-learning with UCB2 scheduling

input One of the UCB2-{Hoeffding, Bernstein} bonuses for updating $\widetilde{Q}$.
 Initialize: $\widetilde{Q}_h(x,a) \leftarrow H$, $Q_h \leftarrow \widetilde{Q}_h$, $t \leftarrow 1$.
 while stopping criterion not satisfied **do**
  Receive x_1.
  Compute L_t according to (16)
  for rounds $r_t = 1, \dots, \lceil L_t/M \rceil$ **do**
   Take action according to Q_h concurrently on M machines and store rewards.
  end for
  Update the policy $\widetilde{Q}_h$ from the stored rewards, according to the given bonus.
  Set $Q_h(\cdot,\cdot) \leftarrow \widetilde{Q}_h(\cdot,\cdot)$ and $t \leftarrow t+1$.
 end while

D.1 Proof of Theorem 5

Suppose we wish to play a total of K episodes concurrently with M machines, and the corresponding non-parallel version of Q-learning is guaranteed to have at most N_{switch} local switches. Let R denote the total number of rounds, then we have

$$R = \sum_{t=1}^{N_{\text{switch}}} r_t = \sum_{t=1}^{N_{\text{switch}}} \left\lceil \frac{L_t}{M} \right\rceil \le \sum_{t=1}^{N_{\text{switch}}} \left(1 + \frac{L_t}{M}\right) \le N_{\text{switch}} + \frac{K}{M}.$$

Now, to find ε near-optimal policy, we know by Corollary 4 that Q-learning with {UCB2H, UCB2B} exploration requires at most

$$K = O\left(\frac{H^{\{5,4\}} SA \log(HSA)}{\varepsilon^2}\right)$$

episodes. Further, choosing K as above, by Theorem 2 and 3, the switching cost is bounded as

$$N_{\text{switch}} \le O\left(H^3 SA \log(K/A)\right) = O\left(H^3 SA \log(HSA/\varepsilon)\right).$$

Plugging these into the preceding bound on R yields

$$R \le O\left(H^3 SA \log(HSA/\varepsilon) + \frac{H^{\{5,4\}} SA \log(HSA)}{\varepsilon^2 M}\right) = \widetilde{O}\left(H^3 SA + \frac{H^{\{5,4\}} SA}{\varepsilon^2 M}\right),$$

the desired result. □

D.2 Concurrent algorithm with mistake bound

Our concurrent algorithm (Algorithm 4 can be converted straightforwardly to an algorithm with low mistake bound. Indeed, for any given ε, by Theorem 5, we obtain an ε near-optimal policy with high probability by running Algorithm 4 for

$$\widetilde{O}\left(H^3 SA + \frac{H^{\{5,4\}} SA}{\varepsilon^2 M}\right)$$

rounds. We then run this ε near-optimal policy forever and are guaranteed to make no mistake.

For such an algorithm, with high probability, "mistakes" can only happen in the exploration phase. Therefore the total amount of "mistakes" (performing an ε sub-optimal action) is upper bounded by the above number of exploration rounds multiplied by HM, as each round consists of at most M machines[7] each performing H actions. This yields a mistake bound

$$\widetilde{O}\left(H^4SAM + \frac{H^{\{6,5\}}SA}{\varepsilon^2}\right)$$

as desired.

E Proof of Theorem 6

Recall that $\mathcal{M}$ denotes the set of all MDPs with horizon H, state space S, action space A, and deterministic rewards in $[0,1]$. Let K be the number of episodes that we can run, and $\mathcal{A}$ be any RL algorithm satisfying that

$$N_{\text{switch}} = \sum_{(h,x)} n_{\text{switch}}(h,x) \le HSA/2$$

almost surely. We want to show that

$$\sup_{M\in\mathcal{M}} \mathbb{E}_{x_1,M}\left[\sum_{k=1}^{K} V_1^\star(x_1) - V_1^{\pi_k}(x_1)\right] \ge \Omega(K),$$

i.e. the worst case regret is linear in K.

E.1 Construction of prior

Let $a^\star : [H]\times[S]\to[A]$ denote a mapping that maps each (h,x) to an action $a^\star(h,x)\in[A]$. There are A^{HS} such mappings. For each $a^\star$, define an MDP $M_{a^\star}$ where the transition is uniform, i.e.

$$x_1 \sim \mathrm{Unif}([S]), \quad x_{h+1}|x_h = x, a_h = a \sim \mathrm{Unif}([S]) \ \text{ for all } (x,a)\in[S]\times[A],\ h\in[H]$$

and the reward is 1 if $a_h = a^\star(h,x_h)$ and 0 otherwise, that is,

$$r_h(x,a) = \mathbf{1}\left\{a = a^\star(h,x)\right\}.$$

Essentially, $M_{a^\star}$ is just a H-fold connection of S parallel bandits that are A-armed, where $a^\star(h,x)$ is the only optimal action at each (h,x).

For such MDPs, as the transition does not depend on the policy, the value functions can be expressed explicitly as

$$\mathbb{E}_{x_1}[V_1^\pi(x_1)] = \frac{1}{S}\sum_{(h,x)\in[H]\times[S]} \mathbf{1}\left\{\pi_h(x) = a^\star(h,x)\right\},$$

and we clearly have

$$\mathbb{E}_{x_1}[V_1^\star(x_1)] \equiv H.$$

[7]To have a fair comparison with CMBIE, if a round does not utilize all M machines, we still let all M machines run and count their actions as their "mistakes".

E.2 Minimax lower bound

Using the sup to average reduction with the above prior, we have the bound

$$\sup_{M\in\mathcal{M}} \mathbb{E}_{x_1,M}\left[\sum_{k=1}^{K} V_1^{\star}(x_1) - V_1^{\pi_k}(x_1)\right] \geq \mathbb{E}_{a^\star}\mathbb{E}_{M_{a^\star}}\left[KH - \sum_{k=1}^{K} V_1^{\pi_k}(x_1)\right]$$
$$= KH - \sum_{k=1}^{K} \mathbb{E}_{a^\star, M_{a^\star}}[V_1^{\pi_k}(x_1)].$$

It remains to upper bound $\mathbb{E}_{a^\star, M_{a^\star}}[V_1^{\pi_k}(x_1)]$ for each k.

For all $k \geq 1$, let

$$n_{\text{switch}}^k(h,x) := \sum_{j=1}^{k-1} \mathbf{1}\left\{\pi_j^h(x) \neq \pi_{j+1}^h(x)\right\} \quad \text{and} \quad N_{\text{switch}}^k = \sum_{h,x} n_{\text{switch}}^k(h,x)$$

denote respectively the switching cost at a single (h,x) and the total (local) switching cost. We use the switching cost to upper bound $\mathbb{E}_{a^\star, M_{a^\star}}[V_1^{\pi_k}]$.

Let

$$A_k(h,x) := \left\{\pi_1^h(x), \dots, \pi_k^h(x)\right\} \subseteq [A]$$

denote the set of visited actions at timestep h and state x. Observe that

$$\mathbb{E}_{a^\star, M_{a^\star}}[V_1^{\pi_k}] = \frac{1}{S}\sum_{h,x} \mathbb{E}\left[\mathbf{1}\left\{a^\star(h,x) = \pi_k^h(x)\right\}\right] \leq \frac{1}{S}\sum_{h,x} \underbrace{\mathbb{E}[\mathbf{1}\left\{a^\star(h,x) \in A_k(h,x)\right\}]}_{:=\Phi_k(h,x)}.$$

Therefore it suffices to bound $\Phi_k(h,x)$.

It is clear that algorithms that only switch to unseen actions can maximize the value function, so we henceforth restrict attention on these algorithms. Let $a^\star = a^\star(h,x)$ and $n_{\text{switch}}^k = n_{\text{switch}}^k(h,x)$ for convenience. Let

$$A_k(h,x) = \left\{a^1, a^2, \dots, a^{n_{\text{switch}}^k + 1}\right\}$$

be the ordered set of unique actions that have been taken at (h,x) throughout the execution of the algorithm. We have

$$\Phi_k(h,x) = \mathbb{P}(a^\star \in A_k(h,x)) = \mathbb{P}\left(\bigcup_{j\geq 1}\left\{n_{\text{switch}}^k + 1 \geq j, a^\star \notin \left\{a^1, a^2, \dots, a^{j-1}\right\}, a^\star = a^j\right\}\right)$$
$$= \sum_{j\geq 1} \mathbb{P}(n_{\text{switch}}^k + 1 \geq j) \cdot \mathbb{P}(a^\star \notin \left\{a^1, a^2, \dots, a^{j-1}\right\}, a^\star = a^j \mid n_{\text{switch}}^k + 1 \geq j).$$

Now, suppose we know that $n_{\text{switch}}^k + 1 \geq j$, then the algorithm have seen the reward on $a^1, \dots, a^{j-1}$. By the uniform prior of $a^\star$, if the algorithm has observed the rewards for all $a \in S$ and found that $a^\star \notin S$, the corresponding posterior for $a^\star$ would be uniform on $[A] \setminus S$. Therefore, we have recursively that

$$\mathbb{P}(a^\star \notin \left\{a^1, \dots, a^{j-1}\right\}, a^\star = a^j \mid n_{\text{switch}}^k + 1 \geq j) = \prod_{\ell=1}^{j-1} \frac{A-\ell}{A-\ell+1} \cdot \frac{1}{A-j+1} = \frac{1}{A}.$$

Substituting this into the preceding bound gives

$$\Phi_k(h,x) = \frac{1}{A}\sum_{j\geq 1}\mathbb{P}(n^k_{\text{switch}} + 1 \geq j) = \frac{\mathbb{E}[n^k_{\text{switch}} + 1]}{A}$$

and thus

$$\mathbb{E}_{a^\star, M_{a^\star}}[V_1^{\pi_k}] \leq \frac{1}{S}\sum_{h,x}\Phi_k(h,x) \leq \frac{1}{S}\sum_{h,x}\frac{\mathbb{E}[n^k_{\text{switch}}(h,x) + 1]}{A} \leq \frac{H}{A} + \frac{\mathbb{E}[N^k_{\text{switch}}]}{SA}$$

As $N^k_{\text{switch}} \leq N^K_{\text{switch}} \leq HSA/2$ almost surely, we have for all k that

$$\mathbb{E}_{a^\star, M_{a^\star}}[V_1^{\pi_k}] \leq H/A + H/2 \leq 3H/4$$

when $A \geq 4$ and thus the regret can be lower bounded as

$$KH - \sum_{k=1}^{K}\mathbb{E}_{a^\star, M_{a^\star}}[V_1^{\pi_k}] \geq KH/4,$$

concluding the proof. □