Reviewer #3 & #4 We thank the reviewers for the positive feedback and helpful comments.

Reviewer #5 We thank the reviewer for raising the questions, which we respond to below.

(2) "Why does switching cost make sense in finite-horizon MDPs?"
We believe there is some misunderstanding in the switching cost considered in this paper. First, our switching cost is *not* to measure the change of per-step policy from step to step, but rather the change of the **overall policies from episode to episode**. Concretely, what we meant by a policy π is the aggregation of per-step policies:

$$\pi = \left\{\pi^h : \mathcal{S} \to \mathcal{A}\right\}_{h=1}^{H}, \text{ or equivalently, a stationary mapping from } \mathcal{S} \times [H] \text{ to } \mathcal{A}$$

and our switching cost measures how π_k differs from π_{k+1}, where π_k is the policy deployed in episode k.

If we understand correctly, part of the reviewer's concern is that since optimal policies in finite-horizon MDPs are nonstationary, *to execute the policy one always needs to switch (between time steps)*, so switching is ubiquitous and should not be costly. **This argument confuses the two kinds of switching (between steps vs. between episodes)**: Switching between steps is cheap as it can be eliminated if we specify a full policy that takes the time step as part of its input. In contrast, switching between episodes can only be eliminated if our (augmented) policy takes data from all previous episodes as input, which is much more costly and sometimes impossible. See our response about motivation below for more examples where frequent policy switching between episodes is impractical.

Finally, we agree with the reviewer that switching cost can also be meaningfully studied in infinite-horizon MDPs, but that is asking the same question under a different MDP formulation, and does not differ from the question we consider here in a fundamentally different manner.

(1) "Practical relevance & technical contributions."
Practical Motivation. There are strong practical motivations for designing RL algorithms with low switching cost: it happens whenever changing the policy has a higher cost than using the same policy to gather data. Our introduction lists a variety of such scenarios in practice, but we'd like to make it clearer here by the following concrete example.

In personalized recommendation systems such as video recommendation on YouTube, the policy specifies what videos we recommend to users given their features. Standard (provably efficient) RL algorithms typically require adjusting its policy based on instantaneous feedback, so for example it needs to update the policy for User 2 after obtaining the feedback on User 1. But this is computationally impractical as there are so many users visiting at every second. In contrast, it makes more sense to use the same policy to aggregate data in a certain period before coming up with an improved policy, which is precisely the setting of low switching cost algorithms. We will improve the presentation of our motivation part to make this clearer to the audience.

Technical Contributions. Our key technical contribution is the establishment of finite-sample regret bound under delayed Q updates. As our algorithm plays greedily according to a delayed version of the Q estimate, a priori it may be the case that the errors caused by the delay may blow up and break the regret bound. We provide a tight control of the errors under UCB2 scheduling and show that these errors do not affect the regret bound. Our techniques are novel and we believe of broader interest for understanding the effect of delayed updates in exploration problems.

(4) "What is new about concurrent RL / relationship with asynchronous Q-Learning."
The concerns of concurrent RL and asynchronous Q-Learning are quite different: concurrent RL cares about the improvement in *speedup of exploration* when multiple machines can each play a copy of the MDP at the same time, whereas asynchronous Q-Learning is about the convergence under asynchrony of updating $Q(s, a)$ for different (s, a). Indeed, existing results on asynchronous Q-Learning [e.g., Even-Dar and Mansour, 2003] only show convergence and do not provide an explicit sample complexity bound, and thus do not cover our result. Furthermore, classical Q-Learning analyses (including the asynchronous ones) dodge the challenge of exploration by assuming that all states are visited sufficiently often, but exploration is a key concern in our setting, so the results are generally incomparable. In fact, concurrent PAC RL is first studied by Guo and Brunskill in 2015, and is a relatively new research direction.

(5) "Why constrained setting in the lower bound."
We chose this simplified setting (lower bounding the regret for algorithms with switching cost $\le HSA/2$) as the problem cannot be formulated in a standard fashion and reduced to information-theoretic tools. Indeed, as we assumed deterministic rewards, it is in principle possible for an algorithm with HSA switches to achieve optimal regret. (Think about a bandit with A arms.) We believe our lower bound provides a useful initial step in understanding the limit of switching costs; stronger lower bounds could be possible when rewards are stochastic, which we leave as future work.

References

Eyal Even-Dar and Yishay Mansour. Learning rates for q-learning. *Journal of machine learning Research*, 5(Dec): 1–25, 2003.