Dynamic Regret of Policy Optimization in Non-Stationary Environments

Yingjie Fei\textsuperscript{1}  Zhuoran Yang\textsuperscript{2}  Zhaoran Wang\textsuperscript{3}  Qiaomin Xie\textsuperscript{1}
\textsuperscript{1} Cornell University, \{yf275, qiaomin.xie\}@cornell.edu
\textsuperscript{2} Princeton University, zy6@princeton.edu
\textsuperscript{3} Northwestern University, zhaoranwang@gmail.com

Abstract

We consider reinforcement learning (RL) in episodic MDPs with adversarial full-information reward feedback and unknown fixed transition kernels. We propose two model-free policy optimization algorithms, POWER and POWER++, and establish guarantees for their dynamic regret. Compared with the classical notion of static regret, dynamic regret is a stronger notion as it explicitly accounts for the non-stationarity of environments. The dynamic regret attained by the proposed algorithms interpolates between different regimes of non-stationarity, and moreover satisfies a notion of adaptive (near-)optimality, in the sense that it matches the (near-)optimal static regret under slow-changing environments. The dynamic regret bound features two components, one arising from exploration, which deals with the uncertainty of transition kernels, and the other arising from adaptation, which deals with non-stationary environments. Specifically, we show that POWER++ improves over POWER on the second component of the dynamic regret by actively adapting to non-stationarity through prediction. To the best of our knowledge, our work is the first dynamic regret analysis of model-free RL algorithms in non-stationary environments.

1 \ Introduction

Classical reinforcement learning (RL) literature often evaluates an algorithm by comparing its performance with that of the best \textit{fixed} (i.e., stationary) policy in hindsight, where the difference is commonly known as regret. Such evaluation metric implicitly assumes that the environment is static so that it is appropriate to compare an algorithm to a single best policy. However, as we advance towards modern and practical RL problems, we face challenges arising in dynamic and non-stationary environments for which comparing against a single policy is no longer sufficient.

Two of the most prominent examples of RL for non-stationary environments are continual RL \cite{WATKINS1989279} and meta RL \cite{Finn2017Model-AgnosticMA, Tripathi_2020} (and more broadly meta learning \cite{Finn2017Model-AgnosticMA, JMLR:v21:20-1083}), which are central topics in the study of generalizability of RL algorithms. In these settings, an agent encounters a stream of tasks throughout time and aims to solve each task with knowledge accrued via solving previous tasks. The tasks can be very different in nature from each other, with potentially increasing difficulties. In particular, the reward mechanism may vary across tasks, and therefore requires the agent to adapt to the change of tasks. Another example of RL under non-stationary environments is human-machine interaction \cite{DBLP:conf/innovations/HoLM18, DBLP:conf/nips/LiangBAA20}. This line of research studies how humans and machines (or robots) should interact or collaborate to accomplish certain goals. In one scenario, a human teaches a robot to complete a task by assigning appropriate rewards to the robot but without intervening its dynamics. The rewards from the human can depend on the stage of the learning process and the rate of improvement in the robot’s behaviors. Therefore, the robot has to adjust its policy over time to maximize the rewards it receives.

In the above examples, it is uninformative to compare an algorithm with a fixed stationary policy, which itself may not perform well given the rapidly changing nature of environments. It is also unclear whether existing algorithms, designed for static environments and evaluated by the standard notion of regret, are sufficient for tackling non-stationary problems.

We aim to address these challenges in this paper. We consider the setting of episodic Markov decision processes (MDPs) with adversarial full-information feedback and unknown transition kernels. We are interested in the notion of dynamic regret, the performance difference between an algorithm and the set of policies optimal for individual episodes in hindsight. For non-stationary RL, dynamic regret is a significantly stronger and more appropriate notion of performance measure than the standard (static) regret, but on the other hand more challenging for algorithm design and analysis.

We propose two efficient, model-free policy optimization algorithms, POWER and POWER++. Under a mild regularity condition of MDPs, we provide dynamic regret analysis for both algorithms and we show that the regret bounds interpolate between different regimes of non-stationarity. In particular, the bounds are of order $\tilde{O}(T^{1/2})$ when the underlying model is nearly stationary, matching with existing near-optimal static regret bounds. In that sense, our algorithms are adaptively near-optimal in slow-varying environments.

To the best of our knowledge, we provide the first dynamic regret analysis for model-free RL algorithms under non-stationary environments.

Our dynamic regret bounds naturally decompose into two terms, one due to maintaining optimism and encouraging exploration in the face of uncertainty associated with the transition kernel, and the other due to the changing nature of reward functions. This decomposition highlights the two main components an RL algorithm needs in order to perform well in non-stationary environments: effective exploration under uncertainty and self-stabilization under drifting reward signals. Our second algorithm, POWER++, takes advantage of active prediction and improves over POWER in terms of the second term in the dynamic regret bounds.

**Our contributions.** The contributions of our work can be summarized as follows:

- We propose two model-free policy optimization algorithms, POWER and POWER++, for non-stationary RL with adversarial rewards;
- We provide dynamic regret analysis for both algorithms, and the regret bounds are applicable across all regimes of non-stationarity of the underlying model;
- When the environment is nearly stationary, our dynamic regret bounds are of order $\tilde{O}(T^{1/2})$ and match the near-optimal static regret bounds, thereby demonstrating the adaptive near-optimality of our algorithms in slow-changing environments.

**Related work.** Dynamic regret has been considered for RL in several papers. The work of [27] considers the setting of online MDP in which the transition kernel and reward functions are allowed to change $l$ times, and the regret compares the algorithm against optimal policies for each of the $l+1$ periods. It proposes UCRL2 with restart, which achieves an $\tilde{O}((l + 1)^{1/3}T^{2/3})$ regret where $T$ is the number of timesteps. The work of [22] considers the same setting and shows that UCRL2 with sliding windows achieves the same regret. Generalizing the previous settings, the work of [59] studies the setting where the changes of model is allowed to take place in every timestep. It proves that UCRL with restart achieves a regret of $\tilde{O}((B_r + B_p)^{1/3}T^{2/3})$ for sufficiently large $B_r, B_p > 0$, where $B_r$ and $B_p$ are the variations of rewards and transition kernels over the $T$ timesteps, respectively. The work of [13] proposes the sliding-window UCRL2 with confidence widening, which achieves an $\tilde{O}((B_r + B_p + 1)^{1/4}T^{3/4})$ regret; under additional regularity conditions, the regret can be improved to $\tilde{O}((B_r + B_p + 1)^{1/3}T^{2/3})$. A Bandit-over-RL algorithm is also provided by [13] to adaptively tune the UCRL2-based algorithm to achieve an $\tilde{O}((B_r + B_p + 1)^{1/4}T^{3/4})$ regret without knowing $B_r$ or $B_p$. The work [54] considers the setting of episodic MDPs in which reward functions and transition kernels get corrupted by an adversary in $K_0$ episodes. It proposes an algorithm called CRANE-RL that achieves a regret of $\tilde{O}(K_0\sqrt{T + K_0^2})$. We remark that all the work discussed so far study model-based algorithms, and we refer interested readers to [40] for an excellent survey on the topic of RL in non-stationary environments. Dynamic regret has also been studied under the settings of multi-armed bandits [3, 6, 8, 11, 12, 31, 33, 48], online convex optimization [1, 24, 26, 44, 47, 49, 52, 55, 62] and games [17]. Interestingly, the notion of dynamic regret is related to the exploitability of strategies in two-player zero-sum games [14]. We would also like to mention a series of papers that consider the
We denote by \( A \) the set of probability distributions supported on a discrete set \( X \). We define

\[
\Delta(X | \mathcal{Y}, H) := \left\{ \{\pi_h(\cdot | \cdot)\}_{h \in [H]} : \pi_h(\cdot | y) \in \Delta(X) \text{ for any } y \in \mathcal{Y} \text{ and } h \in [H] \right\}
\]

for any set \( \mathcal{Y} \) and horizon length \( H \in \mathbb{Z}_{>0} \). For \( p_1, p_2 \in \Delta(X) \), we define \( D_{\text{KL}}(p_1 \| p_2) \) to be the KL divergence between \( p_1 \) and \( p_2 \), that is, \( D_{\text{KL}}(p_1 \| p_2) := \sum_{x \in X} p_1(x) \log \left( \frac{p_1(x)}{p_2(x)} \right) \).

## 2 Preliminaries

### 2.1 Episodic MDPs and dynamic regret

In this paper, we study RL in non-stationary environments via episodic MDPs with adversarial full-information reward feedback and unknown fixed transition kernels. An episodic MDP is defined by the state space \( S \), the action space \( A \), the length \( H \) of each episode, the transition kernels \( \{ P_h(\cdot | \cdot, \cdot)\}_{h \in [H]} \) and the reward functions \( \{ r_h^k : S \times A \rightarrow \{0, 1\} \}_{(k,h) \in [K] \times [H]} \). We assume that the reward functions are deterministic and potentially different across episodes, and that both \( S \) and \( A \) are discrete sets of sizes \( |S| \) and \( |A| \), respectively.

An agent interacts with the MDP through \( K \) episodes without knowledge of \( \{P_h\} \). At the beginning of episode \( k \), the environment provides an arbitrary state \( s^k_1 \) to the agent and chooses reward functions \( \{ r_h^k \}_{h \in [H]} \). The choice of the reward functions is possibly adversarial and may depend on the history of the past \( (k - 1) \) episodes. In step \( h \) of episode \( k \), the agent observes state \( s^k_h \) and then takes an action \( a^k_h \) upon which the environment transitions to the next state \( s^k_{h+1} \sim P(\cdot | s^k_h, a^k_h) \). At the same time, the environment also reveals the reward function \( r^k_h \) to the agent, and the agent receives the reward \( r^k_h(s^k_h, a^k_h) \) (known as the full-information setting). At step \( H + 1 \), the agent observes state \( s^k_{H+1} \) but does not take any action (therefore receiving no reward), and episode \( k \) is completed.

Denote by \( T := KH \) the total number of steps taken throughout the \( K \) episodes.

For any fixed policy \( \pi = \{\pi_h\}_{h \in [H]} \in \Delta(A | S, H) \) and any \( (k, h, s, a) \in [K] \times [H] \times S \times A \), we define the value function \( V_{\pi, k}^h : S \rightarrow \mathbb{R} \) as

\[
V_{\pi, k}^h(s) := \mathbb{E}_\pi \left[ \sum_{i=h}^H r^k_i(s_i, a_i) \mid s_h = s \right],
\]

and the corresponding action-value function \( Q_{\pi, k}^h : S \times A \rightarrow \mathbb{R} \) as

\[
Q_{\pi, k}^h(s, a) := \mathbb{E}_\pi \left[ \sum_{i=h}^H r^k_i(s_i, a_i) \mid s_h = s, a_h = a \right].
\]

Here, the expectation \( \mathbb{E}_\pi[\cdot] \) is taken over the randomness of the state-action tuples \( \{(s_h, a_h, s_{h+1})\}_{h \in [H]} \), where the action \( a_h \) is sampled from the policy \( \pi_h(\cdot | s_h) \) and the next state \( s_{h+1} \) is sampled from the transition kernel \( P_h(\cdot | s_h, a_h) \). The Bellman equation is given by

\[
Q_{\pi, k}^h(s, a) = r^k_h + \mathbb{P}_h V_{\pi, k}^h, \quad V_{\pi, k}^h(s) := \left\langle Q_{\pi, k}^h, \pi_h \right\rangle_A, \quad V_{\pi, H+1}^\pi(s) = 0. \tag{1}
\]

In Equation (1), we use \( \langle \cdot, \cdot \rangle_A \) to denote the inner product over \( A \) and we will omit the subscript \( A \) in the sequel when appropriate; we also define the operator

\[
(\mathbb{P}_h f)(s, a) := \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot | s, a)}[f(s')]
\]
for any function $f : S \rightarrow \mathbb{R}$.

Under the setting of episodic MDPs, the agent aims to approximate the optimal non-stationary policy by interacting with the environment. Let $\pi^* = \arg\max_{\pi \in \Delta(A|S,H)} V_1^{\pi,k}(s_t^1)$ be the optimal policy of episode $k$, and suppose that the agent executes policy $\pi^k$ in episode $k$. The difference in values between $V_1^{\pi^*,k}(s_t^1)$ and $V_1^{\pi^k,k}(s_t^1)$ serves as the regret or the sub-optimality of the agent’s policy $\pi^k$ in episode $k$. Therefore, the dynamic regret for $K$ episodes is defined as

$$D\text{-Regret}(K) := \sum_{k \in [K]} \left[ V_1^{\pi^*,k}(s_t^1) - V_1^{\pi^k,k}(s_t^1) \right].$$

Dynamic regret is a stronger notion than the classical regret measure found in the literature of online learning and reinforcement learning, which is also known as static regret and defined as

$$\text{Regret}(K) := \sum_{k \in [K]} \left[ V_1^{\pi^*,k}(s_t^1) - V_1^{\pi^k,k}(s_t^1) \right],$$

where $\pi^* = \arg\max_{\pi \in \Delta(A|S,H)} \sum_{k \in [K]} V_1^{\pi,k}(s_t^1)$. In words, dynamic regret compares the agent’s policy to the optimal policy of each individual episode in the hindsight, while static regret compares the agent’s policy to only the optimal fixed policy over all episodes combined. Therefore, the notion of dynamic regret is a more natural measure of performance under non-stationary environments. It is clear that dynamic regret always upper bounds static regret:

$$D\text{-Regret}(K) \geq \max_{\pi \in \Delta(A|S,H)} \sum_{k \in [K]} \left[ V_1^{\pi,k}(s_t^1) - V_1^{\pi^k,k}(s_t^1) \right] = \text{Regret}(K).$$

When $\{\pi^*,k\}$ happen to be identical for all episodes $k \in [K]$, dynamic regret reduces to static regret.

### 2.2 Model assumptions

For any policy $\pi$, step $h \in [H]$ and states $s, s' \in S$, we denote by $P^\pi_h(s' | s)$ the probability of transitioning from $s$ to $s'$ in step $h$ when policy $\pi$ is executed, i.e., $P^\pi_h(s' | s) := \sum_{a \in A} P_h(s' | s, a) \cdot \pi_h(a | s)$. The quantity $P^\pi_h$ is also known as the visitation measure of $\pi$ at state $s$ and step $h$. For any pair of policies $\pi$ and $\pi'$, we define the shorthands

$$\|\pi - \pi'\|_\infty := \max_{s \in S} \|\pi(\cdot | s) - \pi'(\cdot | s)\|_1,$$

$$\|P_h^\pi - P_h^{\pi'}\|_\infty := \max_{s \in S} \|P_h^\pi(\cdot | s) - P_h^{\pi'}(\cdot | s)\|_1.$$ 

The following assumption stipulates that the visitation measures are smooth with respect to policies.

**Assumption 1** (Smooth visitation measures). We assume that there exists a universal constant $C > 0$ such that $\|P_h^\pi - P_h^{\pi'}\|_\infty \leq C \cdot \|\pi - \pi'\|_\infty$ for all $h \in [H]$ and all pairs of policies $\pi, \pi'$.

Assumption 1 states that the visitation measures do not change drastically when similar policies are executed. This notion of smoothness in visitation measures also appears in [41] in the context of two-player games.

**Remark 1.** Assumption 1 can in fact be relaxed to $\|P_h^\pi - P_h^{\pi'}\|_\infty \leq C \cdot \|\pi - \pi'\|_\infty$ for all $h \in [H]$ and $C = O(T^\alpha)$ that holds for all $\alpha > 0$ (i.e., the Lipschitz parameter $C$ is sub-polynomial in $T$), and our algorithms and results remain the same. We choose to instead require $C > 0$ to be a universal constant for clear exposition.

Next, we introduce several measures of changes in MDPs and algorithms. Define

$$P_T := \sum_{k \in [K]} \sum_{h \in [H]} \|\pi^*_h - \pi^*_h\|_\infty,$$

where we set $\pi^*_h = \pi^*_h$ for $h \in [H]$. Note that $P_T$ measures the total variation in the optimal policies of adjacent episodes. Oftentimes, algorithms are designed to estimate the optimal policies...
We present our first algorithm, POWER, in Algorithm 1. Algorithm 1 is inspired by the work of where we set $\phi_k$ for $h \in [H]$. Therefore, the quantity $D_T$ computes total variation in algorithmic iterates $\{Q^k\}$. The notions of $P_T$ and $D_T$ are also used in the work of [7, 22, 27, 39, 42, 43] and are known as variation budgets or path lengths. We assume that we have access to quantities $P_T$ and $D_T$ or their upper bounds via an oracle, but we do not know $\{\pi^{*,k}\}$. Such assumptions are standard in non-stationary RL and online convex optimization [7, 22, 27, 39, 42, 43].

2.3 Connections with popular RL paradigms

We briefly discuss how the setting introduced in Section 2.1 is related to several popular paradigms of RL. In certain settings of continual and meta RL, an agent needs to solve tasks one after another in the same physical environment and receives rewards for each task commensurate to the agent’s performance in solving the task. A task can therefore be seen as an episode in our episodic setting. Since the tasks are presented and solved within the same physical environment, it is sufficient to assume a fixed transition model as we do in Section 2.1. On the other hand, the tasks to be solved by the agent can be substantially different from each other in reward mechanism, as such detail of each task is potentially determined by the agent’s performance in all previous tasks. This suggests that the rewards of the tasks are possibly non-stationary, corresponding to the quantities $\{r^k_h\}$ in our setting.

Our setting can also be viewed as a high-level abstraction for human-machine interaction. As in the example discussed in Section 1, a human guides a robot (the learner) to accomplish certain tasks. Here, we can think of the period in between two presented rewards as an episode in our setting. We may also set the physical state of the robot as the state of our model, thus implying a fixed state transition from the robot’s perspective. Moreover, the rewards are controlled by the human in a way that possibly depends on time and history of the robot’s performance, which corresponds to our assumption on $\{r^k_h\}$.

3 Algorithms

In this section, we present two efficient and model-free algorithms: Policy Optimization With PEriodic Restart (POWER) and its enhanced version, POWER++. Let us introduce some additional notations before proceeding. We set $d = |S||A|$, and let $\phi(s, a)$ be the canonical basis of $\mathbb{R}^d$ corresponding to the state-action pair $(s, a) \in S \times A$: that is, the $(s', a')$-th entry of $\phi(s, a)$ equals to 1 if $(s, a) = (s', a')$ and 0 otherwise.

3.1 POWER

We present our first algorithm, POWER, in Algorithm 1. Algorithm 1 is inspired by the work of [9, 18]. It mainly consists of a policy update and a policy evaluation step. The policy update step in Line 7 is equivalent to solving the following optimization problem:

$$
\pi^k = \arg\max_{\pi \in \Delta(A \mid S, H)} L_{k-1}(\pi) - \frac{1}{\alpha} E_{\pi^k-1} \left[ \sum_{h \in [H]} D_{KL}(\pi_h(\cdot \mid s_h) \mid \pi_{h-1}(\cdot \mid s_h)) \right]_{s_1 = s^k_1},
$$

where

$$
L_{k-1}(\pi) := V_{1}^{\pi^{k-1}, k-1}(s^k_1) + E_{\pi^{k-1}} \left[ \sum_{h \in [H]} \left( Q_{h}^{\pi^{k-1}, k-1}(s_h, \cdot), \pi_h(\cdot \mid s_h) - \pi_{h-1}(\cdot \mid s_h) \right) \right]_{s_1 = s^k_1}
$$

is a local linear approximation of $V_{1}^{\pi, k-1}(s^k_1)$ at $\pi = \pi^{k-1}$. In view of Equation (6), we observe that the policy update step can be seen as a mirror descent (MD) step with KL divergence as the Bregman
Algorithm 1 POWER

Input: Confidence level $\delta$, number of episodes $K$, restart cycle length $\tau$, step size $\alpha$, regularization factor $\lambda$ and bonus multiplier $\beta$

1: for episode $k = 1, \ldots, K$ do
2: Receive the initial state $s_1^k$
3: if $k \mod \tau = 1$ then \Comment{periodic restart}
4: Set $\{Q_h^{k-1}\}_{h \in [H]}$ as zero functions and $\{\pi_h^{-1}\}_{h \in [H]}$ as uniform distributions on $A$
5: end if
6: for step $h = 1, 2, \ldots, H$ do \Comment{policy update}
7: Update the policy by $\pi_h^k(\cdot | \cdot) \propto \pi_h^{k-1}(\cdot | \cdot) \cdot \exp\{\alpha \cdot Q_h^{k-1}(\cdot, \cdot)\}$
8: Take action $a_h^k \sim \pi_h^k(\cdot | s_h^k)$
9: Observe the reward function $r_h^k(\cdot, \cdot)$ and receive the next state $s_{h+1}^k$
10: end for
11: Compute $\{Q_h^k\}$ by EvaluatePolicy($k, \{r_h^k\}, \{\pi_h^k\}, \beta$) \Comment{policy evaluation}
12: end for

This step can be efficiently computed by taking the sample mean of $\{V_{h+1}^k(s_h^k)\}_{t \in [k-1]}$. In fact, one has

$$w_h^k(s, a) = \phi(s, a) \top w_h^k = \sum_{s' \in S} \frac{N_h^k(s, a, s')}{N_h^k(s, a)} + \lambda \cdot V_{h+1}(s'),$$

for each $(s, a)$, where the function $N_h^k$ counts the number of times each tuple $(s, a, s')$ or $(s, a)$ has been visited by the algorithm at step $h$ prior to episode $k$. To facilitate exploration in the face of uncertainties, EvaluatePolicy additionally defines a bonus term $\Gamma_h^k(s, a) \propto [N_h^k(s, a)]^{-1/2}$ for each state-action pair $(s, a)$. The estimated action-value function is then set as $Q_h^k = r_h^k + w_h^k + \Gamma_h^k$.

In addition to updating and evaluating policy, Algorithm 1 features a periodic restart mechanism, which resets its policy estimate every $\tau$ episodes. Restart mechanisms have been used to handle non-stationarity in RL \cite{27, 39} and related problems including bandits \cite{6}, online convex optimization \cite{7, 26} and games \cite{17, 41}. Intuitively, by employing the restart mechanism, Algorithm 1 is able to stabilize its iterates against non-stationary drift in the learning process due to adversarial reward functions. We remark that our Algorithm 1 is very different from those used in the existing non-stationary RL literature. Notably, Algorithm 1 is model-free, which is more efficient than the model-based algorithms proposed in e.g., \cite{12, 22, 27, 34, 39}, with respect to both time and space complexities.

3.2 POWER++

Instead of only passively tackling non-stationarity, we may enhance our algorithms with active prediction of the environment. Optimistic mirror descent (OMD) provides exactly such prediction functionality via the so-called predictable sequences. It is well-known in the online learning literature that OMD provides improved regret guarantees than MD algorithm \cite{42, 43}. First proposed by \cite{35} under the name “mirror-prox”, OMD maintains a sequence of main and intermediate iterates. Through the predictable sequences in intermediate iterates, it exploits certain structures of the problem at hand, and therefore achieve better theoretical guarantees. We incorporate predictable sequences into POWER and arrive at an enhanced algorithm, POWER++, which is presented in Algorithm 2.

In Algorithm 2 Lines 8 and 12 together form the OMD steps. Line 10 estimates the intermediate action-value function $Q_h^{k-1/2}$ to be used in the second OMD step (Line 12). The series of iterates $\{Q_h^{k-1}\}$ in Line 8 is the so-called predictable sequence in OMD. Note that we do not execute the
Algorithm 2 POWER++

**Input:** Confidence level \( \delta \), number of episodes \( K \), restart cycle length \( \tau \), step size \( \alpha \), regularization factor \( \lambda \) and bonus multiplier \( \beta \)

1. Set \( \{r_h^k\}_{h \in [H]} \) as zero functions
2. for episode \( k = 1, \ldots, K \) do
3.   Receive the initial state \( s_1^k \)
4.   if \( k \mod \tau = 1 \) then \( \triangleright \) periodic restart
5.     Set \( \{Q_h^{k-1}\}_{h \in [H]} \) as zero functions and \( \{\pi^{k-1}_h\}_{h \in [H]} \) as uniform distributions on \( \mathcal{A} \)
6.   end if
7. for step \( h = 1, 2, \ldots, H \) do \( \triangleright \) intermediate policy update
8.     Update the policy by \( \pi_h^{k-1/2}((\cdot, \cdot)) \propto \pi_h^{k-1}((\cdot, \cdot)) \exp\{\alpha \cdot Q_h^{k-1}((\cdot, \cdot))\} \)
9. end for
10. Compute \( \{Q_h^{k-1/2}\} \) by EvaluatePolicy\((k, \{r_h^k\}, \{\pi_h^k\}, \lambda, \beta)\) \( \triangleright \) intermediate policy evaluation
11. for step \( h = 1, 2, \ldots, H \) do \( \triangleright \) main policy update
12.     Update the policy by \( \pi_h^k((\cdot, \cdot)) \propto \pi_h^{k-1}((\cdot, \cdot)) \exp\{\alpha \cdot Q_h^{k-1}((\cdot, \cdot))\} \)
13.     Take action \( a_h^k \sim \pi_h^k((\cdot, s_h^k)) \)
14.     Observe the reward function \( r_h^k((\cdot, \cdot)) \) and receive the next state \( s_{h+1}^k \)
15. end for
16. Compute \( \{Q_h^k\} \) by EvaluatePolicy\((k, \{r_h^k\}, \{\pi_h^k\}, \lambda, \beta)\) \( \triangleright \) main policy evaluation
17. end for

The result also holds if we replace \( \Pi_{[a,b]}(x) := \max\{\min\{x, b\}, a\} \) and we adopt the convention that \( x/0 = \infty \) for \( x \in \mathbb{R} \). We also define \( L := \lceil K/\tau \rceil \) to be the number of restarts that take place in Algorithm 1 or 2. The following theorem gives an upper bound for the dynamic regret incurred by Algorithm 1.

**Theorem 1** (Upper bound for Algorithm 1). Under Assumption 7 for any \( \delta \in (0, 1] \), with probability at least \( 1 - \delta \) and the choice of \( \lambda = 1, \epsilon = \sqrt{\log A / H^2 T}, \tau = \Pi_{[1,K]}(\lceil T \log A / H^2 T \rceil^{2/3}) \) and \( \beta = C_\beta H \sqrt{S \log (dT/\delta)} \) (for some universal constant \( C_\beta > 0 \)) in Algorithm 2, the dynamic regret of Algorithm 2 is bounded by

\[
\text{D-Regret}(K) \lesssim \sqrt{H^3 S^2 AT \cdot \log^2 (dT/\delta)} + \begin{cases} \sqrt{H^3 T \log A}, & \text{if } 0 \leq P_T \leq \sqrt{\log A}, \\ (H^2 T \sqrt{\log A})^{2/3} P_T^{1/3}, & \text{if } \sqrt{\log A} \leq P_T \leq K \sqrt{\log A}, \\ H^2 P_T, & \text{if } P_T \geq K \sqrt{\log A}. \end{cases}
\]

The result also holds if we replace \( P_T \) in the above with its upper bound. When the upper bounds on \( \text{D-Regret}(K) \) exceed \( T \), we have \( \text{D-Regret}(K) \leq T \).

The proof is given in Appendix C. The regret bound in Theorem 1 interpolates smoothly throughout three regimes of \( P_T \):

- Small \( P_T \): when \( 0 \leq P_T \leq \sqrt{\log A / K} \), the dynamic regret scales as \( \tilde{O}(T^{1/2}) \) and subsumes the static regret results in [9, 18] under the full-information setting. In view of [4], this bound is also nearly optimal (up to polynomial factors of \( H, S \) and \( A \)). Therefore, our bound in Theorem 1 is adaptively near-optimal under small \( P_T \);

- Medium \( P_T \): when \( \sqrt{\log A / K} \leq P_T \leq K \sqrt{\log A} \), the dynamic regret scales as \( {\cal O}(T^{2/3}) \) and is adaptively near-optimal in a medium range of \( P_T \).

- Large \( P_T \): when \( P_T \geq K \sqrt{\log A} \), the dynamic regret scales as \( {\cal O}(T) \) and is adaptively near-optimal in a large range of \( P_T \).

4 Main results

To help with the presentation of our main results, we define the thresholding operator \( \Pi_{[a,b]}(x) := \max\{\min\{x, b\}, a\} \) and we adopt the convention that \( x/0 = \infty \) for \( x \in \mathbb{R} \). We also define \( L := \lceil K/\tau \rceil \) to be the number of restarts that take place in Algorithm 1 or 2. The following theorem gives an upper bound for the dynamic regret incurred by Algorithm 1.

**Theorem 1** (Upper bound for Algorithm 1). Under Assumption 7 for any \( \delta \in (0, 1] \), with probability at least \( 1 - \delta \) and the choice of \( \lambda = 1, \epsilon = \sqrt{\log A / H^2 T}, \tau = \Pi_{[1,K]}(\lceil T \log A / H^2 T \rceil^{2/3}) \) and \( \beta = C_\beta H \sqrt{S \log (dT/\delta)} \) (for some universal constant \( C_\beta > 0 \)) in Algorithm 2, the dynamic regret of Algorithm 2 is bounded by

\[
\text{D-Regret}(K) \lesssim \sqrt{H^3 S^2 AT \cdot \log^2 (dT/\delta)} + \begin{cases} \sqrt{H^3 T \log A}, & \text{if } 0 \leq P_T \leq \sqrt{\log A}, \\ (H^2 T \sqrt{\log A})^{2/3} P_T^{1/3}, & \text{if } \sqrt{\log A} \leq P_T \leq K \sqrt{\log A}, \\ H^2 P_T, & \text{if } P_T \geq K \sqrt{\log A}. \end{cases}
\]

The result also holds if we replace \( P_T \) in the above with its upper bound. When the upper bounds on \( \text{D-Regret}(K) \) exceed \( T \), we have \( \text{D-Regret}(K) \leq T \).

The proof is given in Appendix C. The regret bound in Theorem 1 interpolates smoothly throughout three regimes of \( P_T \):

- Small \( P_T \): when \( 0 \leq P_T \leq \sqrt{\log A / K} \), the dynamic regret scales as \( \tilde{O}(T^{1/2}) \) and subsumes the static regret results in [9, 18] under the full-information setting. In view of [4], this bound is also nearly optimal (up to polynomial factors of \( H, S \) and \( A \)). Therefore, our bound in Theorem 1 is adaptively near-optimal under small \( P_T \);

- Medium \( P_T \): when \( \sqrt{\log A / K} \leq P_T \leq K \sqrt{\log A} \), the dynamic regret scales as \( {\cal O}(T^{2/3}) \) and is adaptively near-optimal in a medium range of \( P_T \).

- Large \( P_T \): when \( P_T \geq K \sqrt{\log A} \), the dynamic regret scales as \( {\cal O}(T) \) and is adaptively near-optimal in a large range of \( P_T \).
• Moderate $P_T$: when $\sqrt{\log A} \leq P_T \leq K \sqrt{\log A}$, we obtain a dynamic regret of order $O(T^{2/3} P_T^{3/2})$, which is $O(T^{2/3})$ if $P_T = O(1)$ and sub-linear in $T$ if $P_T = o(K)$. Similar $O(T^{2/3})$ bounds have been achieved by model-based algorithms in $[13, 22, 27, 39]$, which are less efficient than our model-free algorithms in both time and space complexities;

• Large $P_T$: when $P_T \geq K \sqrt{\log A}$, the model is highly non-stationary and Algorithm 1 incurs a linear regret in $T$.

In addition, the dynamic regret bound in Theorem 1 can be seen as a combination of two parts. The first is the cost paid for being optimistic and due to sum of bonus terms $\{B_k^T\}$ in Algorithm 5 (see Equation (15) in the proof for details). This part is necessary to enforce optimism in the face of uncertainty generated by the transition kernels and is key to effective exploration. The second part is the error caused by non-stationarity of reward functions and depends on $P_T$. Such decomposition is not available in the dynamic regret analysis of online convex optimization problems where MD/OMD-based algorithms have been widely applied. In particular, the dynamic regret bound for online optimization lacks the term due to bonus as it does not require exploration, which is nevertheless a key component underlying RL algorithms that provably explore.

Next we present a result for Algorithm 2.

Theorem 2 (Upper bound for Algorithm 2). Under Assumption 7, for any $\delta \in [0, 1]$, with probability at least $1 - \delta$ and the choice of $\lambda = 1$, $\alpha = \sqrt{L H \log A / D_T}$, $\tau = \Pi_{[1, K]}\left(\left(\frac{\sqrt{D_T \cdot \log A}}{H^2 P_T}\right)^{2/3}\right)$ and $\beta = C_\beta H \sqrt{S \log (d T / \delta)}$ (for some universal constant $C_\beta > 0$) in Algorithm 2, the dynamic regret of Algorithm 2 is bounded by

$$\text{D-Regret}(K) \leq \sqrt{H^3 S^2 A T \log^2(d T / \delta)} + \left\{ \begin{array}{ll}
\sqrt{D_T \cdot H \log A}, & \text{if } 0 \leq P_T \leq \sqrt{D_T \log A / K^2 H^2}, \\
(H \sqrt{D_T \cdot \log A})^{2/3} P_T^{1/3}, & \text{if } \sqrt{D_T \log A / K^2 H^2} \leq P_T \leq \frac{D_T \log A}{H^2}, \\
H^2 P_T, & \text{if } P_T \geq \frac{D_T \log A}{H^2}. 
\end{array} \right.$$ 

The result also holds if we replace $P_T$ and $D_T$ in the above with their upper bounds. When the upper bounds on $\text{D-Regret}(K)$ exceed $T$, we have $\text{D-Regret}(K) \leq T$.

The proof is given in Appendix D. A few remarks about Theorem 2 are in order. Similar to Theorem 1, the result in Theorem 2 interpolates across three regimes depending on the magnitude of $P_T$, and decomposes into two terms respectively arising from the uncertainties of transition kernels and non-stationarity of reward functions. Moreover, thanks to the OMD steps in Algorithm 2 that actively make predictions via predictable sequence $\{Q_h^{k-1}\}$, the bound in Theorem 2 is strictly better than that in Theorem 1 in view of the fact that $D_T \leq KH^3$. When $P_T$ is moderate, i.e., $\sqrt{D_T \log A / K^2 H^2} \leq P_T \leq \frac{D_T \log A}{H^2}$, the dynamic regret bound in Theorem 2 is of order $O(T^{1/3} D_T^{1/3} P_T^{1/3})$, which is similar to the result of [26] Theorem 3] obtained for online optimization problems. Regret bounds that depend on $D_T$, the variation of predictable sequences, have also appeared in [42, 43], although for static regret and online optimization problems.

Technical highlights. A central step of our dynamic regret analysis is to control the expected performance difference between the estimated policies $\{\pi^k\}$ and the optimal $\{\pi^*, k\}$, defined as

$$\sum_{k \in [K]} \sum_{k \in [H]} \mathbb{E}_{\pi^*, k} \left[ \left( Q^k_h(s_h, \cdot), \pi^*_h(\cdot \mid s_h) - \pi^k_h(\cdot \mid s_h) \right) \mid s_1 = s^1_1 \right].$$

Note the the expectation is taken over $\{\pi^*, k\}$ which may vary over episodes $k$. For static regret, i.e., when $\pi^*, k \equiv \pi^k$ for $k \in [K]$, we may control the above term by a standard telescoping argument, which is not viable for dynamic regret analysis. Instead, we decompose the above expectation into $\mathbb{E}_{\pi^*, k} \left[ \right] = \mathbb{E}_{\pi^*, k_0} \left[ \right] + \mathbb{E}_{\pi^*, k - k_0, k_0} \left[ \right]$. Here, $k_0 < k$ is the episode in which restart takes place most recently prior to episode $k$. The first expectation $\mathbb{E}_{\pi^*, k_0} \left[ \right]$ is taken over $\pi^*, k_0$, which stays constant for the period from $k_0$ to the next restart. Therefore, we may apply a customized telescoping argument to each period between restarts. The second expectation $\mathbb{E}_{\pi^*, k - k_0, k_0} \left[ \right]$ from the decomposition

\[ \mathbb{E}_{\pi^*, k - k_0, k_0} \left[ \right] = \mathbb{E}_{\pi^*, k - k_0, k_0} \left[ \right] \]
involves the difference $\pi^*, k - \pi^*, k_0$ and can be bounded by $P_T$. See Lemmas 3 and 4 in Appendices, respectively, for details of controlling the two expectations. Furthermore, it is noteworthy that the restart cycle length $\tau$ plays an important role of balancing the tradeoffs that 1) the optimal policies between two adjacent restarts are relatively stationary among themselves so that the algorithm is compared to stable benchmarks, and that 2) there are not too many restarts so that the sub-optimality of algorithm do not grow too fast when combined over periods in between restarts.

**Comparison with existing results.** We compare the results in Theorems 1 and 2 to those in [13], which is so far state-of-the-art in dynamic regret analysis for non-stationary RL. First, our model-free algorithms are more efficient than the model-based algorithm in [13] that is adapted from UCRL2 and requires solving linear programs in each timestep. Second, our bounds in Theorems 1 and 2 are on the near-optimal order $\tilde{O}(T^{1/2})$ when $P_T$ is sufficiently small, whereas the results in [13] are of order $\tilde{O}(T^{2/3})$. On the other hand, [13] studies a more general setting where the transition kernel of the MDP is allowed to vary adversarially in each timestep. It also provides a procedure to adaptively tune its UCRL2-based algorithm to achieve an $\tilde{O}(T^{3/4})$ regret without knowledge of variations such as $P_T$.

**Broader Impact**

This work provides novel algorithms and analysis for non-stationary RL, which is the foundation of several important RL paradigms including continual/meta RL and human-machine interaction. We present two efficient and model-free policy optimization algorithms for episodic MDPs with adversarial reward functions and fixed unknown transitions. For both algorithms, we provide dynamic regret bounds that interpolate between different regimes of non-stationarity of the underlying model. We show that our bounds achieve the near-optimal $\tilde{O}(T^{1/2})$ order and are adaptively near-optimal in slow-changing environments. To the best of our knowledge, our work provides the first dynamic regret analysis for model-free algorithms in non-stationary RL.

**Acknowledgments and Disclosure of Funding**

Q. Xie is partially supported by NSF grant 1955997.

**References**


Appendices

A Implementation of EvaluatePolicy

Algorithm 3 EvaluatePolicy

**Input:** Episode index $k$, reward functions $\{r_h\}$, policies $\{\pi_h\}$, regularization factor $\lambda$ and bonus multiplier $\beta$

**Output:** Updated Q-values $\{Q_h\}$

1: Initialize $V_{H+1}$ as a zero function
2: for step $h = H, H-1, \ldots, 1$ do
3: \hspace{1cm} $\Lambda_h \leftarrow \sum_{\ell \in [k-1]} \phi(s^\ell_h, a^\ell_h) \phi(s^\ell_h, a^\ell_h)^\top + \lambda \cdot \mathbf{I}$
4: \hspace{1cm} $w_h \leftarrow (\Lambda_h)^{-1} \sum_{\ell \in [k-1]} \phi(s^\ell_h, a^\ell_h) \cdot V_{h+1}(s^\ell_{h+1})$
5: \hspace{1cm} $\Gamma_h(\cdot, \cdot) \leftarrow \beta \cdot \langle \phi(\cdot, \cdot)^\top (\Lambda_h)^{-1} \phi(\cdot, \cdot) \rangle^{1/2}$
6: \hspace{1cm} $Q_h(\cdot, \cdot) \leftarrow r_h(\cdot, \cdot) + \min_{\pi} \phi(\cdot, \cdot) \tau w_h + \Gamma_h(\cdot, \cdot), H-h \rangle^+$
7: \hspace{1cm} $V_h(\cdot) \leftarrow \langle Q_h(\cdot, \cdot), \pi_h(\cdot | \cdot) \rangle_A$
8: end for

In Algorithm 3 the tuples $\{(s^\ell_h, a^\ell_h)\}_{\ell \in [k-1]}$ are state-action pairs visited by Algorithm 1 or 2 before episode $k$.

B Proofs of technical lemmas

Recall that $L := \lceil \frac{K}{H} \rceil$. Algorithm 1 divides $K$ episodes into $L$ periods, and at the the beginning of each period it resets its Q-value and policy estimates. Each period contains $\tau$ episodes, except for the last one, which consists of at most $\tau$ episodes. For ease of notations, we assume that the last period has exactly $\tau$ episodes. Our proof can be easily extended to the case where the last period has fewer than $\tau$ episodes.

B.1 Regret decomposition

For any $(k, h, s) \in [K] \times [H] \times \mathcal{S}$, we define the model prediction error

$$e^k_h := r^k_h + \mathbb{P}_h V^k_{h+1} - Q^k_h,$$

We have the following decomposition of the dynamic regret (2).

**Lemma 1.** We have

$$\text{D-Regret}(K) = \sum_{\ell \in [L]} \sum_{k=(l-1)\tau+1}^{l \tau} \sum_{h \in [H]} \mathbb{E}_{\pi^\ast, k} \left[ \left\langle Q^k_h(\cdot, \cdot), \pi^\ast_h(\cdot | s_h) - \pi^k_h(\cdot | s_h) \right\rangle \mid s_1 = s^k_1 \right]$$

$$+ \sum_{\ell \in [L]} \sum_{k=(l-1)\tau+1}^{l \tau} \sum_{h \in [H]} \mathbb{E}_{\pi^\ast, k} \left[ \left| e^k_h(s_h, a_h) \mid s_1 = s^k_1 \right| - e^k_h(s^k_h, a^k_h) \right] + M_{K,H},$$

where $M_{K,H} := \sum_{k \in [K]} \sum_{h \in [H]} M^k_h$ is a martingale that satisfies $|M^k_h| \leq 4H$ for $(k, h) \in [K] \times [H]$.

We defer its proof to Section B.5.

B.2 Performance difference bound

We may further decompose the first term on the RHS of Lemma 1 as

$$\sum_{\ell \in [L]} \sum_{k=(l-1)\tau+1}^{l \tau} \sum_{h \in [H]} \mathbb{E}_{\pi^\ast, k} \left[ \left\langle Q^k_h(\cdot, \cdot), \pi^\ast_h(\cdot | s_h) - \pi^k_h(\cdot | s_h) \right\rangle \mid s_1 = s^k_1 \right]$$
We first introduce a “one-step descent” result.

Lemma 2. For any $s \in \mathcal{S}$, state $s \in \mathcal{S}$, and function $Q : \mathcal{S} \times \mathcal{A} \rightarrow [0, H]$, it holds for a distribution $p'$ supported on $A$ with $p'(\cdot) \propto p(\cdot) \cdot \exp\{\alpha \cdot Q(s, \cdot)\}$ that

$$
\langle Q(s, \cdot), p^*(\cdot) - p(\cdot) \rangle \leq \frac{1}{2} \alpha H^2 + \frac{1}{\alpha} \left[ D_{KL}(p^*(\cdot) \parallel p(\cdot)) - D_{KL}(p^*(\cdot) \parallel p'(\cdot)) \right].
$$

The next lemma controls the performance difference for any initial state.

Lemma 3. For any $s^k_1 \in \mathcal{S}$, we have

$$
\sum_{l \in [L]} \sum_{k=(l-1)\tau+1}^{\tau} \sum_{h \in [H]} \mathbb{E}_{\pi^*,(l-1)\tau+1} \left[ \langle Q_h^k(s_h, \cdot), \pi_h^{*,k}(\cdot | s_h) - \pi_h^k(\cdot | s_h) \rangle \right] s_1 = s^k_1
\leq \frac{1}{2} \alpha KH^3 + \frac{1}{\alpha} LH \log A + \tau HP_T.
$$

Proof. For each $l \in [L]$, we let $\nu^l = \{\nu^l_h\}_{h \in [H]}$ where each $\nu^l_h$ is a policy (or a distribution supported on $A$) to be specified. We have the decomposition

$$
\sum_{l \in [L]} \sum_{k=(l-1)\tau+1}^{\tau} \sum_{h \in [H]} \mathbb{E}_{\pi^*,(l-1)\tau+1} \left[ \langle Q_h^k(s_h, \cdot), \pi_h^{*,k}(\cdot | s_h) - \pi_h^k(\cdot | s_h) \rangle \right] s_1 = s^k_1
= \sum_{l \in [L]} \sum_{k=(l-1)\tau+1}^{\tau} \sum_{h \in [H]} \mathbb{E}_{\pi^*,(l-1)\tau+1} \left[ \langle Q_h^k(s_h, \cdot), \nu^l_h(\cdot | s_h) - \pi_h^k(\cdot | s_h) \rangle \right] s_1 = s^k_1
+ \sum_{l \in [L]} \sum_{k=(l-1)\tau+1}^{\tau} \sum_{h \in [H]} \mathbb{E}_{\pi^*,(l-1)\tau+1} \left[ \langle Q_h^k(s_h, \cdot), \pi_h^{*,k}(\cdot | s_h) - \nu^l_h(\cdot | s_h) \rangle \right] s_1 = s^k_1
=: E_1 + E_2.
\tag{9}
$$

By Lemma 2 we have

$$
\left| E_1 \right| \leq \frac{1}{2} \alpha KH^3 + \sum_{h \in [H]} \frac{1}{\alpha}
\times \sum_{l \in [L]} \mathbb{E}_{\pi^*,(l-1)\tau+1} \left[ \sum_{k=(l-1)\tau+1}^{\tau} \left[ D_{KL}(\nu^l_h(\cdot | s_h) \parallel \pi_h^{(l-1)\tau+1}(\cdot | s_h)) - D_{KL}(\nu^l_h(\cdot | s_h) \parallel \pi_h^{(l-1)\tau+1}(\cdot | s_h)) \right] \right] s_1 = s^k_1
\leq \frac{1}{2} \alpha KH^3 + \sum_{h \in [H]} \frac{1}{\alpha}
\times \sum_{l \in [L]} \mathbb{E}_{\pi^*,(l-1)\tau+1} \left[ D_{KL}(\nu^l_h(\cdot | s_h) \parallel \pi_h^{(l-1)\tau+1}(\cdot | s_h)) - D_{KL}(\nu^l_h(\cdot | s_h) \parallel \pi_h^{(l-1)\tau+1}(\cdot | s_h)) \right] s_1 = s^k_1
\leq \frac{1}{2} \alpha KH^3 + \sum_{h \in [H]} \frac{1}{\alpha}
\sum_{l \in [L]} \mathbb{E}_{\pi^*,(l-1)\tau+1} \left[ D_{KL}(\nu^l_h(\cdot | s_h) \parallel \pi_h^{(l-1)\tau+1}(\cdot | s_h)) \right] s_1 = s^k_1
$$

14
\begin{align*}
&\leq \frac{1}{2} \alpha KH^3 + \frac{1}{\alpha} LH \log A,
\end{align*}

where the second step holds by telescoping, the third step holds since the KL divergence is non-negative, and the last step holds since by construction \( \pi_h^{(l-1)\tau+1} \mid \cdot, s \) in Algorithm 1 is a uniform distribution on \( A \) and for any policy \( \nu \) and state \( s \in S \) we have

\begin{align*}
D_{KL}(\nu(\cdot \mid s) || \pi_h^{(l-1)\tau+1}(\cdot \mid s)) &= \sum_{a \in A} \nu(a \mid s) \cdot \log (A \cdot \nu(a \mid s)) \\
&= \log A + \sum_{a \in A} \nu(a \mid s) \cdot \log (\nu(a \mid s)) \\
&\leq \log A
\end{align*}

given the fact that the entropy of any distribution is non-negative.

Now for each \((l, h) \in [L] \times [H]\), we set

\[ \nu^l_h := \pi^*_{h}^{(l-1)\tau+1}, \]

that is, \( \nu^l_h \) is the policy after one update in step \( h \) of period \( l \). For \( D_2 \), we have

\begin{align*}
E_2 &\leq \sum_{l \in [L]} \sum_{t=1}^{l\tau} \sum_{k=1}^{(l-1)\tau+1} \sum_{h \in [H]} E_{\pi^*,(l-1)\tau+1} \left[ H \cdot \| \pi^*_{h}^{(l-1)\tau+1}(\cdot \mid s_h) - \nu^l_h(\cdot \mid s_h) \|_1 \mid s_1 = s_1^k \right] \\
&= H \cdot \sum_{l \in [L]} \sum_{t=1}^{l\tau} \sum_{k=1}^{(l-1)\tau+1} \sum_{h \in [H]} \sum_{i=1}^{t\tau+1} \sum_{i=1}^{\tau+1} \max_{s' \in S} \| \pi^*_{h}^{t\tau+i}(\cdot \mid s') - \pi^*_{h}^{t\tau+i-1}(\cdot \mid s') \|_1 \\
&= H \cdot \tau \cdot \sum_{l \in [L]} \sum_{i=1}^{l\tau+1} \tau \max_{s' \in S} \| \pi^*_{h}^{t\tau+i}(\cdot \mid s') - \pi^*_{h}^{t\tau+i-1}(\cdot \mid s') \|_1 \\
&= H \cdot \tau \cdot P_T
\end{align*}

where the first step holds by Holder’s inequality and the fact that \( \| Q_h^k(s, \cdot) \|_\infty \leq H \), the second step holds by the definition of \( \{ \nu^l_h \} \), the third step follows from telescoping, and the last step holds by the definition \( P_T := \sum_{k \in [K]} \sum_{h \in [H]} \| \pi^*_{h}^{k} - \pi^*_{h}^{k-1} \|_\infty \).

**B.2.2 Second term in Equation (8)**

The following lemma controls the performance difference due to varying optimal policies across episodes.

**Lemma 4.** Under Assumption 1, we have

\begin{align*}
\sum_{l \in [L]} \sum_{k=1}^{(l-1)\tau+1} \sum_{h \in [H]} (E_{\pi^*,k} - E_{\pi^*,(l-1)\tau+1}) \left[ \left\langle Q^k_h(s_h, \cdot), \pi^*_{h}^{k}(\cdot \mid s_h) - \pi^*_{h}^{k-1}(\cdot \mid s_h) \right\rangle \mid s_1 = s_1^k \right] \\
&\leq C \cdot \tau H^2 P_T,
\end{align*}

where \( C > 0 \) is a universal constant.

**Proof.** We denote by \( \mathbb{I}(s_h) \) the indicator function for state \( s_h \), and we have

\begin{align*}
\sum_{l \in [L]} \sum_{k=1}^{(l-1)\tau+1} \sum_{h \in [H]} (E_{\pi^*,k} - E_{\pi^*,(l-1)\tau+1}) \left[ \left\langle Q^k_h(s_h, \cdot), \pi^*_{h}^{k}(\cdot \mid s_h) - \pi^*_{h}^{k-1}(\cdot \mid s_h) \right\rangle \mid s_1 = s_1^k \right]
\end{align*}
where the first step follows from $\langle Q^k_h(s_h, \cdot), \pi^k_h(\cdot | s_h) - \pi^k_h(\cdot | s_h) \rangle \leq 2H \cdot \|s_h\|$ and the last step holds by telescoping. Let $P_i(s)$ be the visitation measure of state $s$ in step $i$ under policy $\pi$, and let us fix an $h \in [H]$. Under Assumption 1, for any pair of policies $\pi_i$ and $\pi'_i$, and $P_i(s | s') := \sum_{a' \in A} P_i(s | s', a') \cdot \pi_h(a' | s')$ is the transition kernel in step $h$ when policy $\pi$ is executed. We have the following smoothness property for the (conditional) visitation measure $P_1, P_2, \ldots, P_{h-1}$ of $(s_h | s_1)$ thanks to Assumption 1.

**Lemma 5.** Under Assumption 1, for any $h \in [H], j \in [h-1], s_h, s_1 \in S$, and policies $\{\pi_i\}_{i \in [H]} \cup \{\pi'_i\}$ we have

$$\left| P_1 \cdot P_2 \cdots P_{h-1}(s_h | s_1) - P'_1 \cdot P'_2 \cdots P'_{h-1}(s_h | s_1) \right| \leq C \cdot \|\pi_j - \pi'_j\|_\infty,$$

where $C > 0$ is a universal constant.

**Proof.** We have

$$\left| P_1 \cdot P_2 \cdots P_{h-1}(s_h | s_1) - P_1 \cdot P_2 \cdots P_{h-1}(s_h | s_1) \right| \leq \sum_{s_2, s_3, \ldots, s_{h-1}} \left| P_2 \cdot P_3 \cdots P_{h-1}(s_j+1 | s_j) - P_2 \cdot P_3 \cdots P_{h-1}(s_j+1 | s_j) \right| \cdot \prod_{i \in [h-1] \setminus \{j\}} P_i(s_{i+1} | s_i)$$

$$\leq \sum_{s_2, s_3, \ldots, s_{h-1}} \sum_{j \in [h-1] \setminus \{j\}} \left| P_2 \cdot P_3 \cdots P_{h-1}(s_j+1 | s_j) - P_2 \cdot P_3 \cdots P_{h-1}(s_j+1 | s_j) \right| \cdot \sum_{s_j} \max_{s_{j+1} \in S} \prod_{i \in [h-1] \setminus \{j\}} P_i(s_{i+1} | s_i)$$

$$\leq C \cdot \|\pi_j - \pi'_j\|_\infty \cdot \sum_{s_2, s_3, \ldots, s_{h-1}} \sum_{j \in [h-1] \setminus \{j\}} \prod_{i \in [h-1] \setminus \{j\}} P_i(s_{i+1} | s_i)$$

$$= C \cdot \|\pi_j - \pi'_j\|_\infty \cdot \sum_{s_2, s_3, \ldots, s_{h-1}} \max_{s_{j+1} \in S} \prod_{i=1}^{j-1} P_i(s_{i+1} | s_i) \cdot \sum_{j=1}^{h-1} \prod_{i=1}^{j-1} P_i(s_{i+1} | s_i)$$

$$\leq C \cdot \|\pi_j - \pi'_j\|_\infty,$$

where steps (i) and (ii) hold by H"older's inequality, and step (iii) holds under Assumption 1. □

Therefore, for $(k, t, h) \in [K]^2 \times [H]$ such that $k \leq t - 1$, we have

$$\left| \left( \mathbb{E}_{\pi_t} - \mathbb{E}_{\pi_{t-1}} \right) \left[ I(s_h) \right] \right| \leq \left\| P_1^* \cdot P_2^* \cdots \cdot P_{h-1}^*(\cdot | s_1^k) - P_1^* \cdot P_2^* \cdots \cdot P_{h-1}^*(\cdot | s_1^k) \right\|_\infty$$
where in the last step we used the definition

\[ \| P_{\pi_i^t}^{t} \|_{\infty} \]

Proof.

\[ \text{Lemma 6. Recall that } \]

\[ \text{Finally, we establish the following result on the performance difference.} \]

\[ \text{B.2.3 Putting together} \]

By applying Lemmas 3 and 4, we have

\[ \sum_{l \in [L]} \sum_{k=(l-1)+1}^{\tau} \sum_{h \in [H]} (E_{\pi^{*},h} - E_{\pi^{*},(l-1)+1}) \left[ \left\langle Q_h^k(s_h, \cdot), \pi_h^{*,k}(\cdot | s_h) - \pi_h^{k}(\cdot | s_h) \right\rangle \bigg| s_1 = s_1^k \right] \]

\[ \leq 2H \cdot C \cdot \sum_{h \in [H]} \left( \sum_{l \in [L]} \sum_{k=(l-1)+1}^{\tau} \sum_{i \in [H]} \| \pi_i^{*,t} - \pi_i^{*,t-1} \|_{\infty} \right) \]

\[ = 2H \cdot C \cdot \sum_{l \in [L]} \sum_{i \in [H]} \| \pi_{i}^{*,k} - \pi_{i}^{*,k-1} \|_{\infty} \]

where in the last step we used the definition \( P_T := \sum_{k \in [K]} \sum_{i \in [H]} \| \pi_{i}^{*,k} - \pi_{i}^{*,k-1} \|_{\infty} \).

\[ \text{Lemma 6. Recall that } P_T := \sum_{k \in [K]} \sum_{i \in [H]} \| \pi_{i}^{*,k} - \pi_{i}^{*,k-1} \|_{\infty}. \text{ Under Assumption 4} \]

\[ \alpha = \sqrt{\frac{\log A}{KH^2}} \text{ in Algorithm 4 and we have} \]

\[ \sum_{l \in [L]} \sum_{k=(l-1)+1}^{\tau} \sum_{h \in [H]} (E_{\pi^{*},h} - E_{\pi^{*},(l-1)+1}) \left[ \left\langle Q_h^k(s_h, \cdot), \pi_h^{*,k}(\cdot | s_h) - \pi_h^{k}(\cdot | s_h) \right\rangle \bigg| s_1 = s_1^k \right] \]

\[ = 2H^2 \sqrt{K \log A + C \cdot \tau H^2 P_T}, \]

for some universal constant \( C > 0. \)

Proof. Recall from Equation 8 that for any \( l \in [L] \), we have

\[ \sum_{l \in [L]} \sum_{k=(l-1)+1}^{\tau} \sum_{h \in [H]} (E_{\pi^{*},h} - E_{\pi^{*},(l-1)+1}) \left[ \left\langle Q_h^k(s_h, \cdot), \pi_h^{*,k}(\cdot | s_h) - \pi_h^{k}(\cdot | s_h) \right\rangle \bigg| s_1 = s_1^k \right] \]

By applying Lemmas 3 and 4 we have

\[ \sum_{l \in [L]} \sum_{k=(l-1)+1}^{\tau} \sum_{h \in [H]} (E_{\pi^{*},h} - E_{\pi^{*},(l-1)+1}) \left[ \left\langle Q_h^k(s_h, \cdot), \pi_h^{*,k}(\cdot | s_h) - \pi_h^{k}(\cdot | s_h) \right\rangle \bigg| s_1 = s_1^k \right] \]

\[ \leq \alpha K H^3 + \frac{1}{\alpha} L H \log A + \tau HP_T + C' \cdot \tau H^2 P_T \]

\[ = 2H^2 \sqrt{K \log A + \tau HP_T + C' \cdot \tau H^2 P_T} \]

\[ \leq 2H^2 \sqrt{K \log A + C \cdot \tau H^2 P_T}, \]

where the equality above holds by our choice of \( \alpha \), and \( C, C' > 0 \) are universal constants.
B.3 Model prediction error

We need the following results to control the bonus $\Gamma^k_h(\cdot, \cdot)$ (defined in Line 5 of Algorithm 3) accumulated over episodes.

**Lemma 7.** Let $\lambda = 1$ and $\beta = C \cdot H \sqrt{S \log(dT/p)}$ in Algorithm 4, where $C > 0$ is a universal constant and $p \in (0, 1]$. With probability at least $1 - p/2$ and for all $(k, h, s, a) \in [K] \times [H] \times S \times A$, it holds that

$$-2\Gamma^k_h(s, a) \leq i^k_h(s, a) \leq 0.$$

**Proof.** The proof follows that of [9, Lemma 4.3] specialized to the tabular setting by replacing Lemma D.2 therein with [5, Lemma 12].

**Lemma 8 ([9] Lemma D.6; [29] Lemma D.2).** Let $\{\phi_t\}_{t \geq 0}$ be a bounded sequence in $\mathbb{R}^d$ satisfying $\sup_{t \geq 0} ||\phi_t|| \leq 1$. Let $\Lambda_0 \in \mathbb{R}^{d \times d}$ be a positive definite matrix with $\lambda_{\min}(\Lambda_0) \geq 1$. For any $t \geq 0$, we define $\Lambda_t := \Lambda_0 + \sum_{i \in [t-1]} \phi_i \phi_i^\top$. Then, we have

$$\log \left[ \frac{\det(\Lambda_{t+1})}{\det(\Lambda_0)} \right] \leq \sum_{i \in [t]} \phi_i^\top \Lambda_i^{-1} \phi_i \leq 2 \log \left[ \frac{\det(\Lambda_{t+1})}{\det(\Lambda_0)} \right].$$

**Lemma 9.** We have

$$\sum_{k \in [K]} \sum_{h \in [H]} \Gamma^k_h(s_h^k, a_h^k) \leq \beta H \sqrt{2dK \log((K + \lambda)/\lambda)}.$$

**Proof.** Given the construction of $\Lambda^k_h$ in Algorithm 1, we have for any $h \in [H]$,

$$\sum_{k \in [K]} \phi(s_h^k, a_h^k)^\top (\Lambda_h^k)^{-1} \phi(s_h^k, a_h^k) \leq 2 \log \left[ \frac{\det(\Lambda_h^{k+1})}{\det(\Lambda_h^k)} \right] \leq 2d \log \left[ \frac{K + \lambda}{\lambda} \right],$$

where the last step holds since the construction of Algorithm 1 implies that $\Lambda_h^1 = \lambda \cdot I$ and $\Lambda_h^{k+1} = \sum_{i \in [k]} \phi(s_h^i, a_h^i) \phi(s_h^i, a_h^i)^\top + \lambda \cdot I \preceq (k + \lambda) \cdot I$, which yields

$$\log \left[ \frac{\det(\Lambda_h^{k+1})}{\det(\Lambda_h^k)} \right] \leq \log \left[ \frac{\det((K + \lambda) \cdot I)}{\det(\lambda \cdot I)} \right] = d \log \left[ \frac{K + \lambda}{\lambda} \right].$$

Therefore, by the Cauchy-Schwarz inequality and Lemma 9, we have

$$\sum_{k \in [K]} \sum_{h \in [H]} \Gamma^k_h(s_h^k, a_h^k) \leq \beta \cdot \sum_{k \in [H]} \left( K \cdot \sum_{k \in [K]} \phi(s_h^k, a_h^k)^\top (\Lambda_h^k)^{-1} \phi(s_h^k, a_h^k) \right)^{1/2} = \beta H \sqrt{2dK \log((K + \lambda)/\lambda)}.$$

**B.4 Martingale bound**

**Lemma 10.** Consider $M_{K, H}$ in Lemma 4. With probability $1 - \delta/2$, we have

$$|M_{K, H}| \leq \sqrt{16H^2T \cdot \log(4/\delta)}.$$

**Proof.** From Lemma 4 and by the Azuma Hoeffding inequality, we have for any $t \geq 0$,

$$\mathbb{P}(\{|M_{K, H}| \geq t\}) \leq 2 \exp \left( - \frac{t^2}{16H^2T} \right).$$

Setting $t = \sqrt{16H^2T \cdot \log(4/\delta)}$, we have

$$|M_{K, H}| \leq \sqrt{16H^2T \cdot \log(4/\delta)}$$

with probability at least $1 - \delta/2$. 

\[\square\]
B.5 Proof of Lemma 4.2

For any function \( f : \mathcal{S} \times \mathcal{A} \to \mathbb{R} \) and any \( (k, h, s) \in [K] \times [H] \times \mathcal{S} \), define the operators

\[
\mathbb{J}_{k,h}^* f(s) = \left\langle f(s, \cdot), \pi_h^*(\cdot | s) \right\rangle,
\]

\[
\mathbb{J}_{k,h} f(s) = \left\langle f(s, \cdot), \pi_h^k(\cdot | s) \right\rangle.
\]

and the function

\[
\xi_h^k(s) := (\mathbb{J}_{k,h}^* Q_h^k)(s) - (\mathbb{J}_{k,h} Q_h^k)(s) = \left\langle Q_h^k(s, \cdot), \pi_h^*(\cdot | s) - \pi_h^k(\cdot | s) \right\rangle.
\]

The proof mostly follows that of [9, Lemma 4.2], except that we replace \( \pi^* \) and \( \mathbb{J}_h \) therein by \( \pi^*, k \) and \( \mathbb{J}_{k,h} \), respectively. Therefore, we outline the key steps only and refer the readers to the proof of [9, Lemma 4.2] for full details.

Recall that \( \pi^*, k \) is the optimal policy in episode \( k \). We have

\[
\text{D-Regret}(K) = \sum_{k \in [K]} \left[ V_1^{\pi^*, k, h}(s_1^k) - V_1^{\pi^*, k, h}(s_1^k) \right]
\]

\[
= \sum_{l \in [L]} \sum_{k = (l-1) + 1} \left[ V_1^{\pi^*, k, h}(s_1^k) - V_1^{\pi^*, k, h}(s_1^k) \right].
\]

We have

\[
V_1^{\pi^*, k, h}(s_1^k) - V_1^{\pi^*, k, h}(s_1^k) = \frac{V_1^{\pi^*, k, h}(s_1^k) - V_1^{\pi^*, k, h}(s_1^k)}{G_1} + \frac{V_1^{\pi^*, k, h}(s_1^k) - V_1^{\pi^*, k, h}(s_1^k)}{G_2}.
\]

(12)

From [9, Section B.1], we have for any \( k \in [K] \),

\[
G_1 = \sum_{h \in [H]} \mathbb{E}_{\pi^*, k} \left[ \mathbb{E}_{\pi^*, k} \left[ Q_h^k(s_h, a_h) \mid s_1 = s_1^k \right] \right]
\]

\[
+ \sum_{h \in [H]} \mathbb{E}_{\pi^*, k} \left[ \left\langle Q_h^k(s_h, \cdot), \pi_h^*(\cdot | s_h) - \pi_h^k(\cdot | s_h) \right\rangle \mid s_1 = s_1^k \right],
\]

(13)

and

\[
G_2 = -\sum_{h \in [H]} t_h^k(s_h^k, a_h^k) + \sum_{h \in [H]} (D_{h,1}^k + D_{h,2}^k),
\]

(14)

where

\[
D_{h,1}^k := \left( \mathbb{J}_{h,k} (Q_h^k - Q_h^{\pi_h^k,k}) \right)(s_h^k) - (Q_h^k - Q_h^{\pi_h^k,k})(s_h^k, a_h^k),
\]

\[
D_{h,2}^k := \mathbb{P}_h(V_h^{k+1} - V_h^{k+1}) (s_h^k, a_h^k) - (V_h^{k+1} - V_h^{\pi_h^k,k})(s_h^{k+1}).
\]

From Line 11 of Algorithm 1, we have

\[
Q_h^k, Q_h^{\pi_h^k,k}, V_h^{k+1}, V_h^{\pi_h^k,k} \in [0, H],
\]

which implies \( |D_{h,1}^k|, |D_{h,2}^k| \leq 2H \) for any \( (k, h) \in [K] \times [H] \). Writing \( M_h^k := D_{h,1}^k + D_{h,2}^k \), we have that

\[
M_{K,H} := \sum_{k \in [K]} \sum_{h \in [H]} M_h^k
\]

is a martingale where \( |M_h^k| \leq 4H \). The proof is completed in view of Equations (12), (13) and (14).
C Proof of Theorem 1

By Lemmas 7 and 9, we have
\[
\sum_{l \in \{L\}} \sum_{k=-(l-1)\tau+1}^{l\tau} \sum_{h \in [H]} [\mathbb{E}_{s_{1}, s_{s_{1}}} \left[ \Gamma_{h}^{k}(s_{h}, a_{h}) | s_{1} = s_{1}^{k} \right] - \psi_{h}^{k}(s_{h}, a_{h})] \leq 2 \sum_{k \in \{K\}} \sum_{h \in [H]} \leq 2 + 2K \sqrt{2dK \log((K + \lambda)/\lambda)}.
\]
(15)

We apply Lemmas 6 and 10 as well as Equation (15) to the conclusion of Lemma 1. With the choice of \( \lambda = 1 \) and \( \beta = C_{\beta} H \sqrt{S \log(dT/\delta)} \) and the identity \( K = L \tau \), we have
\[
\text{D-Regret}(K) \leq 2H^{2} \sqrt{KL \log A} + C \cdot \tau H^{2} P_{T} + 2 \beta H \sqrt{2dK \log((K + \lambda)/\lambda)}
\]
\[
+ \sqrt{16H^{2}T \cdot \log(4/\delta)}
\]
\[
\leq 2H^{2} \sqrt{KL \log A} + C \cdot \tau H^{2} P_{T} + 2C_{\beta} H^{2} \sqrt{S \log(dT/\delta)} \sqrt{2dK \log(K + 1)}
\]
\[
+ \sqrt{16H^{2}T \cdot \log(4/\delta)}
\]
\[
= 2H^{2} \sqrt{KL \log A} + C \cdot \tau H^{2} P_{T} + C_{\beta} H^{2} \sqrt{S \log(dT/\delta)} \cdot \log(4/\delta) \cdot \log(K + 1)
\]
\[
+ \sqrt{16H^{2}T \cdot \log(4/\delta)}
\]
\[
\leq 2H^{2} \sqrt{KL \log A} + C \cdot \tau H^{2} P_{T} + C_{\beta} H^{2} \sqrt{S \log(dT/\delta)} \cdot \log(4/\delta)
\]
\[
\leq 2H^{2} \sqrt{KL \log A} + C \cdot \tau H^{2} P_{T} + C_{\beta} H^{2} \sqrt{S \log(dT/\delta)} \cdot \log(4/\delta)
\]
(16)

where \( C, C' > 0 \) are universal constants, the second step above holds by the definition of \( \beta \), and the third step holds by the identity \( T = KH \).

We discuss several cases.

- If \( 0 \leq P_{T} \leq \sqrt{\frac{\log A}{K}} \), then by elementary calculation we have \( \left( \frac{\sqrt{\log A}}{HT^{2}} \right)^{2} \geq K \). This implies \( \tau = K \) by our choice of \( \tau \), and therefore \( L = 1 \). Then Equation (16) yields

\[
\text{D-Regret}(K) \leq 2H^{2} \sqrt{KL \log A} + C \cdot H^{2} \sqrt{K \log A} + C_{\beta} H^{2} \sqrt{S \log(dT/\delta)} \cdot \log^{2}(dT/\delta)
\]
\[
= (2 + C) \sqrt{H^{2} \log A} + C_{\beta} H^{2} \sqrt{S \log(dT/\delta)} \cdot \log^{2}(dT/\delta).
\]

- If \( \sqrt{\frac{\log A}{K}} \leq P_{T} \leq 2^{3/2} \cdot K \sqrt{\log A} \), we have and \( 2 \leq \tau \leq K \) and Equation (16) implies

\[
\text{D-Regret}(K) \leq 2H^{2} \sqrt{KL \log A} + C \cdot H^{2} P_{T} + C_{\beta} H^{2} S^{2} \sqrt{A T} \cdot \log^{2}(dT/\delta)
\]
\[
\leq (4 + C) \cdot \left( \frac{\sqrt{\log A}}{HT^{2}} \right)^{2} \cdot \left( \frac{\sqrt{\log A}}{HT^{2}} \right)^{2/3} P_{T}^{1/3} + C_{\beta} H^{2} S^{2} \sqrt{AT} \cdot \log^{2}(dT/\delta),
\]

where the first step holds by \( K = L \tau \), and in the last step we applied the choice of \( \tau = \left( \frac{\sqrt{\log A}}{HT^{2}} \right)^{2/3} \).

- If \( P_{T} > 2^{3/2} \cdot K \sqrt{\log A} \), we have \( \left( \frac{\sqrt{\log A}}{HT^{2}} \right)^{2/3} < 2 \) and therefore \( \tau = 1 \) and \( L = K \). Then Equation (16) yields

\[
\text{D-Regret}(K) \leq 2H^{2} \sqrt{KL \log A} + C \cdot H^{2} P_{T} + C_{\beta} H^{2} S^{2} \sqrt{A T} \cdot \log^{2}(dT/\delta)
\]
\[
\leq (8 + C) H^{2} P_{T} + C' H^{2} S^{2} \sqrt{AT} \cdot \log^{2}(dT/\delta),
\]

It is not hard to see that all of the above arguments also go through if we replace \( P_{T} \) with its upper bound. The proof is completed by combining the last case above with the trivial bound \( \text{D-Regret}(K) \leq T \).
D Proof of Theorem

The proof follows the same reasoning as in Appendix C except that Lemmas no longer applies. In the following, we provide an alternative to Lemmas adapted for Algorithm 2.

Lemma 11. For any \( s \in S \), we have

\[
\sum_{l \in [L]} \sum_{k=(l-1)\tau+1}^{l\tau} \sum_{h \in [H]} \left\langle Q_h^k(s, \cdot), \pi_h^* \pi_h^k \cdot | s) - \pi_h^k(\cdot | s) \right\rangle \\
\leq \alpha D_T + \frac{1}{\alpha} LH \log A + \tau HP_T.
\]

Proof. Let us fix an \( s \in S \). For each \( l \in [L] \), we let \( \nu_l^1 = \{\nu_l^h\}_{h \in [H]} \) where each \( \nu_l^h \) is a policy (or a distribution supported \( A \)) that depends only on \( l \) and \( h \). We have the decomposition

\[
\sum_{l \in [L]} \sum_{k=(l-1)\tau+1}^{l\tau} \sum_{h \in [H]} \left\langle Q_h^k(s, \cdot), \pi_h^* \pi_h^k \cdot | s) - \pi_h^k(\cdot | s) \right\rangle \\
\leq \sum_{l \in [L]} \sum_{k=(l-1)\tau+1}^{l\tau} \sum_{h \in [H]} \left\langle Q_h^k(s, \cdot), \nu_l^h(\cdot | s) - \pi_h^k(\cdot | s) \right\rangle \\
+ \sum_{l \in [L]} \sum_{k=(l-1)\tau+1}^{l\tau} \sum_{h \in [H]} \left\langle Q_h^k(s, \cdot), \pi_h^* \pi_h^k \cdot | s) - \nu_l^h(\cdot | s) \right\rangle \\
=: E_1 + E_2.
\]

The term \( E_2 \) can be controlled in exactly the same way as in the proof of Lemma 3. Therefore, we only control \( E_1 \). Note that the policy update steps in Algorithm Lines 8 and 12 essentially follow the update steps of OMD (see e.g. [50] Section 3.1.1 for details). This observation enables us to take advantage of the following lemma, which is a version of [50] Proposition 5 adapted to our case.

Lemma 12. For any \( (l, h, s) \in [L] \times [H] \times S \), we have

\[
\sum_{k=(l-1)\tau+1}^{l\tau} \left\langle Q_h^k(s, \cdot), \nu_l^h(\cdot | s) - \pi_h^k(\cdot | s) \right\rangle \\
\leq \frac{\log A}{\alpha} + \alpha \cdot \sum_{k=(l-1)\tau+1}^{l\tau} \|Q_h^k(s, \cdot) - Q_h^{k-1}(s, \cdot)\|^2_\infty - \frac{1}{8\alpha} \cdot \sum_{k=(l-1)\tau+1}^{l\tau} \|\pi_h^k - \pi_h^{k-1}\|^2_\infty.
\]

Proof. The result follows from [50] Proposition 5 and we note that the quantity \( R \) defined therein is upper bounded by \( \log A \).

By Lemma 12 and the definition of \( D_T \) in Equation 5, we have

\[
E_1 \leq L \cdot H \cdot \frac{\log A}{\alpha} + \alpha \cdot D_T,
\]

We have the following result on the performance difference, similar to Lemma 6.

Lemma 13. Recall that \( P_T := \sum_{k \in [K]} \sum_{s \in [H]} \|\pi_i^{s_k} - \pi_i^{s_{k-1}}\|_\infty \). Under Assumption 1 we choose \( \alpha = \sqrt{\frac{\text{LHlog} A}{D_T}} \) in Algorithm 2 and we have

\[
\sum_{l \in [L]} \sum_{k=(l-1)\tau+1}^{l\tau} \sum_{h \in [H]} \mathbb{E}_{\pi_i^{s_k}} \left[ \left\langle Q_h^k(s_h, \cdot), \pi_h^* \pi_h^k \cdot | s_h) - \pi_h^k(\cdot | s_h) \right\rangle \right| s_1 = s_1^k \\
= 2\sqrt{D_T LH \log A} + C \cdot \tau H^2 P_T,
\]

for some universal constant \( C > 0 \).
Proof. Now, for any $l \in [L]$ we have the decomposition

\[
\sum_{t \in [L]} \sum_{k = (l-1)+1}^{l+1} \sum_{h \in [H]} \mathbb{E}_{\pi^{\tau,k}} \left[ \left< Q_h^k(s_h, \cdot), \pi_h^{\tau,k}(\cdot | s_h) - \pi_h^k(\cdot | s_h) \right> \mid s_1 = s_1^k \right]
\]

\[
= \sum_{t \in [L]} \sum_{k = (l-1)+1}^{l+1} \sum_{h \in [H]} \mathbb{E}_{\pi^{\tau,k}} \left[ \left< Q_h^k(s_h, \cdot), \pi_h^{\tau,k}(\cdot | s_h) - \pi_h^k(\cdot | s_h) \right> \mid s_1 = s_1^k \right]
\]

\[
+ \sum_{t \in [L]} \sum_{k = (l-1)+1}^{l+1} \sum_{h \in [H]} (\mathbb{E}_{\pi^{\tau,k}} - \mathbb{E}_{\pi^{\tau,(l-1)+1}}) \left[ \left< Q_h^k(s_h, \cdot), \pi_h^{\tau,k}(\cdot | s_h) - \pi_h^k(\cdot | s_h) \right> \mid s_1 = s_1^k \right]
\]

By applying Lemmas [1] and [4] we have

\[
\sum_{t \in [L]} \sum_{k = (l-1)+1}^{l+1} \sum_{h \in [H]} \mathbb{E}_{\pi^{\tau,k}} \left[ \left< Q_h^k(s_h, \cdot), \pi_h^{\tau,k}(\cdot | s_h) - \pi_h^k(\cdot | s_h) \right> \mid s_1 = s_1^k \right]
\]

\[
\leq \alpha D_T + \frac{LH \log A}{\alpha} + \tau H P_T + C' \cdot \tau H^2 P_T
\]

\[
= 2 \sqrt{D_T LH \log A} + \tau H P_T + C' \cdot \tau H^2 P_T
\]

\[
\leq 2 \sqrt{D_T LH \log A} + C \cdot \tau H^2 P_T,
\]

where the last equality holds by our choice of $\alpha$, and $C, C' > 0$ are universal constants.

We apply Lemmas [13] and [10] and Equation (15) to the conclusion of Lemma 1. With the choice of $\lambda = 1$ and $\beta = C_\beta H \sqrt{S \log(dT/\delta)}$, we have

\[
\text{D-Regret}(K) \leq 2 \sqrt{D_T LH \log A} + C \cdot \tau H^2 P_T + 2\beta H \sqrt{2dK \log((K + \lambda)/\lambda)}
\]

\[
+ \sqrt{16H^2 T \cdot \log(4/\delta)}
\]

\[
\leq 2 \sqrt{D_T LH \log A} + C \cdot \tau H^2 P_T + 2C_\beta H^2 \sqrt{S \log(dT/\delta)} \sqrt{2dK \log(K + 1)}
\]

\[
+ \sqrt{16H^2 T \cdot \log(4/\delta)}
\]

\[
= 2 \sqrt{D_T LH \log A} + C \cdot \tau H^2 P_T + 2C_\beta \sqrt{2H^3 S^2 A T \cdot \log(dT/\delta) \cdot \log(K + 1)}
\]

\[
+ \sqrt{16H^2 T \cdot \log(4/\delta)}
\]

\[
\leq 2 \sqrt{D_T LH \log A} + C \cdot \tau H^2 P_T + C' \sqrt{2H^3 S^2 A T \cdot \log^2(dT/\delta)}
\]

where $C, C' > 0$ are universal constants, the second step holds by the definition of $\beta$, and the third step holds by the identity $T = KH$. Analyzing Equation (17) in the same way as Equation (16) in Section C (for different regimes of $P_T$) yields the result.