Near-Optimal Model-Free Reinforcement Learning in Non-Stationary Episodic MDPs

Weichao Mao [1] **Kaiqing Zhang** [1] **Ruihao Zhu** [2] **David Simchi-Levi** [2] **Tamer Başar** [1]

Abstract

We consider model-free reinforcement learning (RL) in non-stationary Markov decision processes. Both the reward functions and the state transition functions are allowed to vary arbitrarily over time as long as their cumulative variations do not exceed certain variation budgets. We propose Restarted Q-Learning with Upper Confidence Bounds (RestartQ-UCB), the first model-free algorithm for non-stationary RL, and show that it outperforms existing solutions in terms of dynamic regret. Specifically, RestartQ-UCB with Freedman-type bonus terms achieves a dynamic regret bound of $\widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}HT^{\frac{2}{3}})$, where S and A are the numbers of states and actions, respectively, $\Delta > 0$ is the variation budget, H is the number of time steps per episode, and T is the total number of time steps. We further show that our algorithm is *nearly optimal* by establishing an information-theoretical lower bound of $\Omega(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{2}{3}}T^{\frac{2}{3}})$, the first lower bound in non-stationary RL. Numerical experiments validate the advantages of RestartQ-UCB in terms of both cumulative rewards and computational efficiency. We further demonstrate the power of our results in the context of multi-agent RL, where non-stationarity is a key challenge.

1. Introduction

Reinforcement learning (RL) focuses on the class of problems where an agent maximizes its cumulative reward through sequential interactions with an initially unknown but fixed environment, usually modeled by a Markov Decision Process (MDP). In classical RL problems, the state transition functions and the reward functions are assumed to be time-invariant, i.e., stationary. However, stationary models cannot capture the time-varying environments in a wide range of sequential decision-making problems, such as online advertisement auctions (Cai et al., 2017; Lu et al., 2019), dynamic pricing (Chawla et al., 2016; Mao et al., 2018), traffic management (Chen et al., 2020), healthcare operations (Shortreed et al., 2011), and inventory control (Agrawal & Jia, 2019).

Among others, we want to highlight the connection between non-stationary RL and multi-agent RL (Littman, 1994). In multi-agent RL, a set of agents collaborate or compete by taking actions in a shared environment. Consequently, each agent is facing a non-stationary environment, especially when the agents learn and update policies simultaneously, as the actions of the other agents can alter the environment. We discuss this connection in greater details in Section 7 and also provide more applications of non-stationary RL to other important problems, such as sequential transfer and multi-task RL, in Appendix A.

RL in a non-stationary MDP is highly non-trivial due to the following challenges. First, similar to stationary RL, the agent faces the *exploration vs. exploitation* dilemma: it needs to explore the uncertain environment efficiently while maximizing its rewards along the way. In Jaksch et al. (2010), the authors proposed to leverage the "optimism in the face of uncertain" principle to guide exploration. Another challenge, which is unique to non-stationary RL, is the trade-off between *remembering and forgetting*. On the one hand, since the underlying MDP varies over time, data samples collected in prior interactions can become obsolete. In fact, it has been shown that a standard stationary RL algorithm might incur a linear regret if the non-stationarity is not handled properly (Ortner et al., 2019). On the other hand, the agent needs to extract a sufficient amount of information from historical data to inform future decision-making.

To resolve the aforementioned challenges, Ortner et al. (2019) and Cheung et al. (2020) have proposed algorithms to guide learning in non-stationary MDPs. Although both

[1]Department of Electrical and Computer Engineering & Coordinated Science Laboratory, University of Illinois Urbana-Champaign, Urbana, IL, USA [2]Institute for Data, Systems, and Society, Massachusetts Institute of Technology, Cambridge, MA, USA. Correspondence to: Weichao Mao <weichao2@illinois.edu>, Kaiqing Zhang <kzhang66@illinois.edu>, Ruihao Zhu <rzhu@mit.edu>, David Simchi-Levi <dslevi@mit.edu>, Tamer Başar <basar1@illinois.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

Setting	Algorithm	Regret	Model-free?	Comment
Undis-counted	Jaksch et al. (2010)	$\widetilde{O}(S\;A^{\frac{1}{2}}L^{\frac{1}{3}}D\;T^{\frac{2}{3}})$	✗	only abrupt changes
	Gajane et al. (2018)	$\widetilde{O}(S^{\frac{2}{3}}A^{\frac{1}{3}}L^{\frac{1}{3}}D^{\frac{2}{3}}T^{\frac{2}{3}})$	✗	only abrupt changes
	Ortner et al. (2019)	$\widetilde{O}(S\;A^{\frac{1}{2}}\Delta^{\frac{1}{3}}D\;T^{\frac{2}{3}})$	✗	requires local variations
	Cheung et al. (2020)	$\widetilde{O}(S^{\frac{2}{3}}A^{\frac{1}{2}}\Delta^{\frac{1}{4}}D\;T^{\frac{3}{4}})$	✗	does not require Δ
	Lower bound	$\Omega(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}D^{\frac{2}{3}}T^{\frac{2}{3}})$		
Episodic	Domingues et al. (2020)	$\widetilde{O}(S\;A^{\frac{1}{2}}\Delta^{\frac{1}{3}}H^{\frac{4}{3}}T^{\frac{2}{3}})$	✗	also metric spaces
	RestartQ-UCB	$\widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H\;T^{\frac{2}{3}})$	✓	
	Lower bound	$\Omega(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{2}{3}}T^{\frac{2}{3}})$		

Table 1: Dynamic regret comparisons for RL in non-stationary MDPs. S and A are the numbers of states and actions, L is the number of abrupt changes, D is the maximum diameter, H is the number of steps per episode, and T is the total number of steps. Gray cells denote the results from this paper.

model-based and model-free algorithms have been proposed for stationary RL, existing solutions for non-stationary RL are often built upon model-based methods. Nevertheless, it has been observed that model-based solutions often suffer from the following shortcomings:

• **Time- and space-inefficiency:** Model-based methods are in general more time- and space-consuming, and are less compatible with the design of modern deep RL architectures (Jin et al., 2018; Zhang et al., 2020).

• **Inefficient exploration:** In Cheung et al. (2020), an example was given to show that under non-stationarity, the estimated model can incorrectly indicate that transitioning between states is very unlikely. This suggests that model-based methods, which try to estimate the latent model, might suffer "The Perils of Drift" (Cheung et al., 2020).

• **Limited applicability:** In an important application of nonstationary RL — *decentralized* multi-agent RL, the agents cannot observe the actions taken by the other agents. This information structure precludes model-based methods, as the explicit estimation of the state transition functions is hardly possible without observing all the agents' actions.

These observations have thus motivated us to turn our attention to model-free methods, which, instead of maintaining estimates of the unknown underlying model, directly learn the Q-values.

Main Contributions. In this paper, we focus on the problem of designing model-free algorithms with nearly-optimal performances for non-stationary RL. Our contributions can be summarized as follows:

1. We introduce an algorithm named Restarted Q-Learning with Upper Confidence Bounds (RestartQ-UCB), which is the first model-free algorithm in the general setting of non-stationary RL. Our algorithm adopts a simple but effective restarting strategy (Jaksch et al., 2010; Besbes et al., 2014) that resets the memory of the agent according to a calculated schedule. The restarting strategy ensures that our algorithm only refers to the most up-to-date experience for decision-making. RestartQ-UCB also utilizes an extra optimism term (in addition to the standard Hoeffding/Freedman-based bonus) for exploration to counteract the non-stationarity of the MDP. This additional bonus term, depending on the local variations (i.e., the environmental variation in each restarting interval), guarantees that our optimistic Q-value is still an upper bound of the optimal $Q^\star$-value even when the environment changes. We further show that our algorithm can easily remove the dependence on local variations, an assumption commonly made in the literature (Ortner et al., 2019; Zhou et al., 2020). Our analysis shows that RestartQ-UCB achieves the lowest dynamic regret bound when compared to existing works in the literature;
2. We conduct simulations showing that RestartQ-UCB achieves highly competitive cumulative rewards against a state-of-the-art solution (Zhou et al., 2020), while only taking 0.18% of its computation time;
3. We establish the first lower bounds in non-stationary RL, which suggest that our algorithm is optimal in all parameter dependences except for an $H^{\frac{1}{3}}$ factor, where H is the episode length;
4. To further showcase the flexibility and potential of non-stationary RL, we illustrate how it can be utilized to address the non-stationarity issue inherent in multi-agent RL. Specifically, we show that RestartQ-UCB can be readily applied to a multi-agent RL example against a slowly-changing opponent (Radanovic et al., 2019; Lee

et al., 2020). The setting we consider is a more practical and general decentralized learning setting, which entails model-free solutions.

Related Work. Dynamic regret of non-stationary RL has been mostly studied using model-based solutions. Jaksch et al. (2010) consider the setting where the MDP is allowed to change abruptly for L times. A sliding window approach is proposed in Gajane et al. (2018) under the same setting. Ortner et al. (2019) generalize the previous setting by allowing the MDP to vary either abruptly or gradually at every step, subject to a total variation budget. Cheung et al. (2020) consider the same setting and introduce a Bandit-over-RL technique that adaptively tunes the algorithm without knowing the variation budget. In a setting most similar to ours, Domingues et al. (2020) investigate non-stationary RL in the episodic setting, and propose a kernel-based approach when the state-action set forms a metric space. Their results can be reduced to an $\widetilde{O}(SA^{\frac{1}{2}}\Delta^{\frac{1}{3}}H^{\frac{4}{3}}T^{\frac{2}{3}})$ regret in the tabular case. Fei et al. (2020) assume stationary transitions and adversarial full-information rewards, and their setting is not directly comparable with ours. Two concurrent works Zhou et al. (2020) and Touati & Vincent (2020) consider non-stationary RL in linear MDPs, but their regret bounds, $\widetilde{O}(S^{\frac{4}{3}}A^{\frac{4}{3}}\Delta^{\frac{1}{3}}H^{\frac{4}{3}}T^{\frac{2}{3}})$ and $\widetilde{O}(S^{\frac{7}{6}}A^{\frac{7}{6}}\Delta^{\frac{1}{3}}H^{\frac{4}{3}}T^{\frac{2}{3}})$ when reduced to the tabular RL setting, respectively, are less competitive than ours. Interested readers are referred to Padakandla (2020) for a comprehensive survey on RL in non-stationary environments. Table 1 compares our regret bounds with existing results that tackle similar settings as ours. It can be seen that our result is the first one that achieves the optimal dependence on S and A, and also establishes the tightest dependence on H/D and T among existing solutions in the literature, without relying on their assumptions.

Another related line of research studies online/adversarial MDPs (Yu & Mannor, 2009; Neu et al., 2010; Arora et al., 2012; Yadkori et al., 2013; Dick et al., 2014; Wang et al., 2018; Lykouris et al., 2019; Jin et al., 2019), but they mostly only allow variations in reward functions, and use the static regret as a performance metric. In addition, RL with low switching cost (Bai et al., 2019) also shares a similar spirit as our restarting strategy since it also periodically forgets previous experiences. However, such algorithms do not address the non-stationarity of the environment, and their dynamic regret in terms of the variation budget is unclear.

Non-stationarity has also been considered in bandit problems. Within different non-stationary multi-armed bandit (MAB) settings, various methods have been proposed, including decaying memory and sliding windows (Garivier & Moulines, 2011; Keskin & Zeevi, 2017), as well as restart-based strategies (Auer et al., 2002; Besbes et al., 2014; Allesiardo et al., 2017). These methods largely inspired later research on non-stationary RL. A more recent line of work developed methods that do not require prior knowledge of the variation budget (Karnin & Anava, 2016; Cheung et al., 2019a) or the number of abrupt changes (Auer et al., 2019). Other related settings considered in the literature include Markovian bandits (Tekin & Liu, 2010; Ma, 2018), non-stationary contextual bandits (Luo et al., 2018; Chen et al., 2019), linear bandits (Cheung et al., 2019b; Zhao et al., 2020), continuous-armed bandits (Mao et al., 2020), and bandits with slowly changing rewards (Besbes et al., 2019).

Outline. The rest of the paper is organized as follows: In Section 2, we introduce the mathematical model of our problem and necessary preliminaries. In Section 3, we present our RestartQ-UCB algorithm. A dynamic regret analysis of RestartQ-UCB is provided in Section 4. In Section 5, we establish information-theoretical lower bounds. Simulation results are presented in Section 6. In Section 7, we discuss the application of our method to multi-agent RL. Finally, we conclude the paper in Section 8.

2. Preliminaries

Model: We consider an episodic RL setting where an agent interacts with a non-stationary MDP for M episodes, with each episode containing H steps. We use a pair of integers (m,h) as a *time index* to denote the h-th step of the m-th episode. The environment can be denoted by a tuple $(\mathcal{S},\mathcal{A},H,P,r)$, where $\mathcal{S}$ is the finite set of states with $|\mathcal{S}|=S$, $\mathcal{A}$ is the finite set of actions with $|\mathcal{A}|=A$, H is the number of steps in one episode, $P=\{P_h^m\}_{m\in[M],h\in[H]}$ is the set of transition kernels, and $r=\{r_h^m\}_{m\in[M],h\in[H]}$ is the set of mean reward functions. Specifically, when the agent takes action $a_h^m\in\mathcal{A}$ in state $s_h^m\in\mathcal{S}$ at the time (m,h), it will receive a random reward $R_h^m(s_h^m,a_h^m)\in[0,1]$ with expected value $r_h^m(s_h^m,a_h^m)$, and the environment transitions to a next state s_{h+1}^m following the distribution $P_h^m(\cdot\mid s_h^m,a_h^m)$. It is worth emphasizing that the transition kernel and the mean reward function depend both on m and h, and hence the environment is non-stationary over time. The episode ends when s_{H+1}^m is reached. We further denote $T=MH$ as the total number of steps.

A deterministic policy $\pi:[M]\times[H]\times\mathcal{S}\to\mathcal{A}$ is a mapping from the time index and state space to the action space, and we let $\pi_h^m(s)$ denote the action chosen in state s at time (m,h). Define $V_h^{m,\pi}:\mathcal{S}\to\mathbb{R}$ to be the value function under policy π at time (m,h), i.e.,

$$V_h^{m,\pi}(s)\stackrel{\text{def}}{=}\mathbb{E}\left[\sum_{h'=h}^{H} r_{h'}^m\left(s_{h'},\pi_{h'}^m\left(s_{h'}\right)\right)\mid s_h=s\right],$$

where $s_{h'+1}\sim P_{h'}^m\left(\cdot\mid s_{h'},a_{h'}\right)$. Accordingly, the state-action value function $Q_h^{m,\pi}: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as:

$$Q_h^{m,\pi}(s,a) \stackrel{\text{def}}{=} r_h^m(s,a) + \mathbb{E}\left[\sum_{h'=h+1}^{H} r_{h'}^m\left(s_{h'}, \pi_{h'}^m\left(s_{h'}\right)\right) \mid s_h = s, a_h = a\right]$$

For simplicity of notation, we let $P_h^m V_{h+1}(s,a) \stackrel{\text{def}}{=} \mathbb{E}_{s' \sim P_h^m(\cdot \mid s,a)}\left[V_{h+1}(s')\right]$. Then, the Bellman equation gives $V_h^{m,\pi}(s) = Q_h^{m,\pi}(s, \pi_h^m(s))$ and $Q_h^{m,\pi}(s,a) = (r_h^m + P_h^m V_{h+1}^{m,\pi})(s,a)$, and we also have $V_{H+1}^{m,\pi}(s) = 0, \forall s \in \mathcal{S}$ by definition. Since the state space, the action space, and the length of each episode are all finite, there always exists an optimal policy $\pi^\star$ that gives the optimal value $V_h^{m,\star}(s) \stackrel{\text{def}}{=} V_h^{m,\pi^\star}(s) = \sup_\pi V_h^{m,\pi}(s), \forall s \in \mathcal{S}, m \in [M], h \in [H]$. From the Bellman optimality equation, we have $V_h^{m,\star}(s) = \max_{a \in \mathcal{A}} Q_h^{m,\star}(s,a)$, where $Q_h^{m,\star}(s,a) \stackrel{\text{def}}{=} (r_h^m + P_h^m V_{h+1}^{m,\star})(s,a)$, and $V_{H+1}^{m,\star}(s) = 0, \forall s \in \mathcal{S}$.

Dynamic Regret: The agent aims to maximize the cumulative expected reward over the entire M episodes, by adopting some policy π. We measure the optimality of the policy π in terms of its *dynamic regret* (Cheung et al., 2020; Domingues et al., 2020), which compares the agent's policy with the optimal policy of each individual episode in hindsight:

$$\mathcal{R}(\pi, M) \stackrel{\text{def}}{=} \sum_{m=1}^{M}\left(V_1^{m,\star}\left(s_1^m\right) - V_1^{m,\pi}\left(s_1^m\right)\right),$$

where the initial state s_1^m of each episode is chosen by an oblivious adversary (Zhang et al., 2020). Dynamic regret is a stronger measure than the standard (static) regret, which only considers the single policy that is optimal over all episodes combined.

Variation: We measure the non-stationarity of the MDP in terms of its *variation* in the mean reward function and transition kernels:

$$\Delta_r \stackrel{\text{def}}{=} \sum_{m=1}^{M-1}\sum_{h=1}^{H} \sup_{s,a} \left|r_h^m(s,a) - r_h^{m+1}(s,a)\right|,$$

$$\Delta_p \stackrel{\text{def}}{=} \sum_{m=1}^{M-1}\sum_{h=1}^{H} \sup_{s,a} \left\|P_h^m(\cdot \mid s,a) - P_h^{m+1}(\cdot \mid s,a)\right\|_1,$$

where $\|\cdot\|_1$ is the L^1-norm. Note that our definition of variation only imposes restrictions on the summation of non-stationarity across two different episodes, and does not put any restriction on the difference between two consecutive steps in the same episode; that is, $P_h^m(\cdot \mid s,a)$ and $P_{h+1}^m(\cdot \mid s,a)$ are allowed to be arbitrarily different. We further let $\Delta = \Delta_r + \Delta_p$, and assume $\Delta > 0$.

3. Algorithm: RestartQ-UCB

We present our algorithm Restarted Q-Learning with Hoeffding Upper Confidence Bounds (RestartQ-UCB Hoeffding) in Algorithm 1. Replacing the Hoeffding term with a Freedman-style one will lead to a tighter regret bound, but the analysis is more involved. For clarity of presentation, we defer the exposition of the Freedman-based algorithm to Appendix E.

RestartQ-UCB breaks the M episodes into D *epochs*, with each epoch containing $K = \lceil \frac{M}{D} \rceil$ episodes (except for the last epoch which possibly has less than K episodes). The optimal value of D (and hence K) will be specified later in our analysis. RestartQ-UCB periodically restarts a Q-learning algorithm with UCB exploration at the beginning of each epoch, thereby addressing the non-stationarity of the environment. For each $d \in [D]$, define $\Delta_r^{(d)}$ to be the *local variation* of the mean reward function within epoch d. By definition, we have $\sum_{d=1}^{D} \Delta_r^{(d)} \le \Delta_r$. Define the local variation of transitions $\Delta_p^{(d)}$ analogously.

Since our algorithm essentially invokes the same procedure for every epoch, in the following, we focus our analysis on what happens inside one epoch only (and without loss of generality, we focus on epoch 1, which contains episodes $1, 2, \ldots, K$). At the end of our analysis, we will merge the results across all epochs.

For each triple $(s,a,h) \in \mathcal{S} \times \mathcal{A} \times [H]$, we divide the visitations (within epoch 1) to the triple into multiple *stages*, where the length of the stages increases exponentially at a rate of $(1 + \frac{1}{H})$. Specifically, let $e_1 = H$, and $e_{i+1} = \lfloor (1 + \frac{1}{H}) e_i \rfloor, i \ge 1$ denote the lengths of the stages. Further, let the partial sums $\mathcal{L} \stackrel{\text{def}}{=} \{\sum_{i=1}^{j} e_i \mid j = 1, 2, 3, \ldots\}$ denote the set of the ending times of the stages. We remark that the stages are defined for each individual triple (s,a,h), and for different triples the starting and ending times of their stages do not necessarily align in time. Such a definition of stages is mostly motivated by the design of the learning rate $\alpha_t = \frac{H+1}{H+t}$ in Jin et al. (2018). It ensures that only the last $O(1/H)$ fraction of samples is given non-negligible weights when used to estimate the optimistic $Q_h(s,a)$ values, while the first $1 - O(1/H)$ fraction is forgotten (Zhang et al., 2020). We set $\iota \stackrel{\text{def}}{=} \log\left(\frac{2}{\delta}\right)$, where δ is the failure probability.

Recall that the time index (k,h) represents the h-th step of the k-th episode. At each step (k,h), we take the optimal action with respect to the optimistic $Q_h(s,a)$ value (Line 6 in Algorithm 1), which is designed as an optimistic estimate of the optimal $Q_h^{k,\star}(s,a)$ value of the corresponding episode. For each triple (s,a,h), we update the optimistic $Q_h(s,a)$ value at the end of each stage, using samples only from this latest stage that is about to end (Line 12 in Algorithm 1). The optimism in $Q_h(s,a)$ comes from two bonus terms b_h^k

Algorithm 1: RestartQ-UCB (Hoeffding)

for *epoch* $d \leftarrow 1$ *to* D **do**
  Initialize: $V_h(s) \leftarrow H - h + 1, Q_h(s,a) \leftarrow H - h + 1, N_h(s,a) \leftarrow 0, \check{N}_h(s,a) \leftarrow 0,$
  $\check{r}_h(s,a) \leftarrow 0, \check{v}_h(s,a) \leftarrow 0$, for all $(s,a,h) \in \mathcal{S} \times \mathcal{A} \times [H]$;
  for *episode* $k \leftarrow (d-1)K + 1$ *to* $\min\{dK, M\}$ **do**
    observe s_1^k;
    for *step* $h \leftarrow 1$ *to* H **do**
      Take action $a_h^k \leftarrow \arg\max_a Q_h(s_h^k, a)$, receive $R_h^k(s_h^k, a_h^k)$, and observe s_{h+1}^k;
      $\check{r}_h(s_h^k, a_h^k) \leftarrow \check{r}_h(s_h^k, a_h^k) + R_h^k(s_h^k, a_h^k), \check{v}_h(s_h^k, a_h^k) \leftarrow \check{v}_h(s_h^k, a_h^k) + V_{h+1}(s_{h+1}^k)$;
      $N_h(s_h^k, a_h^k) \leftarrow N_h(s_h^k, a_h^k) + 1, \check{N}_h(s_h^k, a_h^k) \leftarrow \check{N}_h(s_h^k, a_h^k) + 1$;
      if $N_h(s_h^k, a_h^k) \in \mathcal{L}$ **then**
        `// Reaching the end of the stage`
        $b_h^k \leftarrow \sqrt{\frac{H^2}{\check{N}_h(s_h^k, a_h^k)}\iota} + \sqrt{\frac{1}{\check{N}_h(s_h^k, a_h^k)}\iota}$, $b_\Delta \leftarrow \Delta_r^{(d)} + H\Delta_p^{(d)}$;
        $Q_h(s_h^k, a_h^k) \leftarrow \min\left\{Q_h(s_h^k, a_h^k), \frac{\check{r}_h(s_h^k, a_h^k)}{\check{N}_h(s_h^k, a_h^k)} + \frac{\check{v}_h(s_h^k, a_h^k)}{\check{N}_h(s_h^k, a_h^k)} + b_h^k + 2b_\Delta\right\}$; $(*)$
        $V_h(s_h^k) \leftarrow \max_a Q_h(s_h^k, a)$;
        $\check{N}_h(s_h^k, a_h^k) \leftarrow 0, \check{r}_h(s_h^k, a_h^k) \leftarrow 0, \check{v}_h(s_h^k, a_h^k) \leftarrow 0$;

and b_Δ, where b_h^k is a standard Hoeffding-based optimism that is commonly used in upper confidence bounds (Jin et al., 2018; Zhang et al., 2020), and b_Δ is the extra optimism that we need to take into account the non-stationarity of the environment. The definition of b_Δ requires knowledge of the local variation budget in each epoch, which is a rather strong assumption in practice. However, we can further show (later in Theorem 2) that if we simply replace Equation $(*)$ in Algorithm 1 with the following update rule:

$$Q_h(s_h^k, a_h^k) \leftarrow \min\left\{\frac{\check{r}_h(s_h^k, a_h^k)}{\check{N}_h(s_h^k, a_h^k)} + \frac{\check{v}_h(s_h^k, a_h^k)}{\check{N}_h(s_h^k, a_h^k)} + b_h^k, Q_h(s_h^k, a_h^k)\right\}, \quad (1)$$

then our algorithm can achieve the same regret bound without the assumption on the local variation budget.

4. Analysis

In this section, we present our main result—a dynamic regret analysis of the RestartQ-UCB algorithm. Our first result on RestartQ-UCB with Hoeffding-style bonus terms is summarized in the following theorem. Complete proofs of its supporting lemmas are given in Appendix B.

Theorem 1. *(Hoeffding) For $T = \Omega(SA\Delta H^2)$, and for any $\delta \in (0,1)$, with probability at least $1-\delta$, the dynamic regret of RestartQ-UCB with Hoeffding bonuses (Algorithm 1) is bounded by $\widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{5}{3}}T^{\frac{2}{3}})$, where $\widetilde{O}(\cdot)$ hides poly-logarithmic factors of T and $1/\delta$.*

Our proof relies on the following technical lemma, stating that for any triple (s,a,h), the difference of their optimal Q-values at two different episodes $1 \le k_1 < k_2 \le K$ is bounded by the variation of this epoch.

Lemma 1. *For any triple (s,a,h) and any $1 \le k_1 < k_2 \le K$, it holds that $|Q_h^{k_1,\star}(s,a) - Q_h^{k_2,\star}(s,a)| \le \Delta_r^{(1)} + H\Delta_p^{(1)}$.*

Let $Q_h^k(s,a)$ denote the value of $Q_h(s,a)$ at the *beginning* of the k-th episode in Algorithm 1. The following lemma states that the optimistic Q-value $Q_h^k(s,a)$ is an upper bound of the optimal Q-value $Q_h^{k,\star}(s,a)$ with high probability. Note that we only need to show that the event holds with probability $1 - \mathtt{poly}(K,H)\delta$, because we can replace δ with $\delta/\mathtt{poly}(K,H)$ in the end to get the desired high probability bound without affecting the polynomial part of the regret bound.

Lemma 2. *(Hoeffding) For $\delta \in (0,1)$, with probability at least $1 - 2KH\delta$, it holds that $Q_h^{k,\star}(s,a) \le Q_h^{k+1}(s,a) \le Q_h^k(s,a), \forall (s,a,h,k) \in \mathcal{S} \times \mathcal{A} \times [H] \times [K]$.*

Building upon Lemmas 1 and 2, a complete proof of Theorem 1 is given in Appendix C. We remark that Algorithm 1 relies on the assumption that the local variations b_Δ are known a priori, which is a strong but commonly made assumption in the literature on non-stationary RL (Ortner et al., 2019; Zhou et al., 2020). To the best of our knowledge, existing restart-based solutions either crucially rely on this local variation assumption (Ortner et al., 2019), or suffer a severe regret degeneration after removing this assumption (Zhou et al., 2020). Interestingly, in the following theorem, we show that this assumption can be safely removed in our approach without affecting the regret bound. The only modification to the algorithm is to replace the Q-value update rule in Equation $(*)$ of Algorithm 1 with the new update rule in Equation (1).

Theorem 2. *(Hoeffding, no local budgets) For $T = \Omega(SA\Delta H^2)$, and for any $\delta \in (0,1)$, with probability at least $1-\delta$, the dynamic regret of RestartQ-UCB with Hoeffding bonuses and no knowledge of local budgets is bounded by $\widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{5}{3}}T^{\frac{2}{3}})$, where $\widetilde{O}(\cdot)$ hides poly-logarithmic factors of T and $1/\delta$.*

To understand why this simple modification works, notice that in $(*)$ we are adding exactly the same value $2b_\Delta$ to the upper confidence bounds of all (s,a) pairs in the same epoch. Subtracting the same value from all optimistic Q-values simultaneously should not change the choice of actions in future steps. The only difference is that the new "optimistic" $Q_h^k(s,a)$ values would no longer be strict upper bounds of the optimal $Q_h^{k,\star}(s,a)$ anymore, but only an "upper bound" subject to some error term of the order b_Δ. This further requires a slightly different analysis on how this error term propagates over time, which is presented as a variant of Lemma 2 as follows.

Lemma 3. *(Hoeffding, no local budgets) Suppose that we have no prior knowledge of the local variations and replace the update rule $(*)$ in Algorithm 1 with Equation (1). For $\delta \in (0,1)$, with probability at least $1-2KH\delta$, it holds that $Q_h^{k,\star}(s,a) - 2(H-h+1)b_\Delta \le Q_h^{k+1}(s,a) \le Q_h^k(s,a), \forall (s,a,h,k) \in \mathcal{S}\times\mathcal{A}\times[H]\times[K]$.*

Remark 1. *The easy removal of the local budget assumption is non-trivial in the design of the algorithm, and to the best of our knowledge is absent in the non-stationary RL literature with restarts. In fact, it has been shown in a concurrent work (Zhou et al., 2020) that removing this assumption could lead to a much worse regret bound (cf. Corollary 2 and Corollary 3 therein).*

Replacing the Hoeffding-based upper confidence bound with a Freedman-style one will lead to a tighter regret bound, summarized in Theorem 3 below. The proof of the theorem follows a similar procedure as in the proof of Theorem 1, and is given in Appendix F. It relies on a reference-advantage decomposition technique for variance reduction as coined in Zhang et al. (2020). The intuition is to first learn a reference value function V^{ref} that serves as a roughly accurate estimate of the optimal value function $V^\star$. The goal of learning the optimal value function $V^\star = V^{\text{ref}} + (V^* - V_{\text{ref}})$ can hence be decomposed into estimating two terms V^{ref} and $V^* - V_{\text{ref}}$, each of which can be more accurately estimated due to the reduced variance. For ease of exposition, we proceed again with the assumption that the local variation budgets are known. The reader should bear in mind that this assumption can be easily removed using a similar technique as in Theorem 2.

Theorem 3. *(Freedman) For T greater than some polynomial of S, A, Δ and H, and for any $\delta \in (0,1)$, with probability at least $1-\delta$, the dynamic regret of RestartQ-UCB with Freedman bonuses (presented in Algorithm 2) is upper bounded by $\widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}HT^{\frac{2}{3}})$, where $\widetilde{O}(\cdot)$ hides poly-logarithmic factors.*

5. Lower Bounds

In this section, we provide information-theoretical lower bounds of the dynamic regret to characterize the fundamental limits of any algorithm in non-stationary RL.

Theorem 4. *For any algorithm, there exists an episodic non-stationary MDP such that the dynamic regret of the algorithm is at least $\Omega(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{2}{3}}T^{\frac{2}{3}})$.*

Proof sketch. The proof of our lower bound relies on the construction of a "hard instance" of non-stationary MDPs. The instance we construct is essentially an MDP with piecewise constant dynamics on each *segment* of the horizon, and its dynamics experience an abrupt change at the beginning of each new segment. Specifically, we divide the horizon T into L segments[1], where each segment has $T_0 \overset{\text{def}}{=} \lfloor \frac{T}{L} \rfloor$ steps and contains $M_0 \overset{\text{def}}{=} \lfloor \frac{M}{L} \rfloor$ episodes. Within each segment, the system dynamics of the MDP do not vary, and we construct the dynamics for each segment in a way such that the instance is a hard instance of stationary MDPs on its own. The MDP within each segment is essentially similar to the hard instances constructed in Osband & Van Roy (2016); Jin et al. (2018). Between two consecutive segments, the dynamics of the MDP change abruptly, and we let the dynamics vary in a way such that no information learned from previous interactions with the MDP can be used in the new segment. In this sense, the agent needs to learn a new hard stationary MDP in each segment. Finally, optimizing the value of L and the variation magnitude between consecutive segments (subject to the constraints of the total variation budget) leads to our lower bound. $\square$

A useful side result of our proof is the following lower bound for non-stationary RL in the un-discounted setting, which is the same setting as studied in Gajane et al. (2018), Ortner et al. (2019) and Cheung et al. (2020).

Proposition 1. *Consider a reinforcement learning problem in un-discounted non-stationary MDPs with horizon length T, total variation budget Δ, and maximum MDP diameter D (Cheung et al., 2020). For any learning algorithm, there exists a non-stationary MDP such that the dynamic regret of the algorithm is at least $\Omega(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}D^{\frac{2}{3}}T^{\frac{2}{3}})$.*

[1]The definition of segments is irrelevant to, and should not be confused with, the notion of epochs we previously defined.

6. Simulations

In this section, we empirically evaluate RestartQ-UCB on reinforcement learning tasks with various types of non-stationarity. We compare RestartQ-UCB with three baseline algorithms: LSVI-UCB-Restart (Zhou et al., 2020), Q-Learning UCB, and Epsilon-Greedy (Watkins, 1989). LSVI-UCB-Restart is a state-of-the-art non-stationary RL algorithm that combines optimistic least-squares value iteration with periodic restarts. Q-Learning UCB is simply our RestartQ-UCB algorithm with no restart. It is a Q-learning based algorithm that uses upper confidence bounds to guide the exploration. Epsilon-Greedy is a restart-based algorithm that uses an epsilon-greedy strategy for action selection.

We evaluate the cumulative rewards of the four algorithms on a variant of a reinforcement learning task named Bidirectional Diabolical Combination Lock (Agarwal et al., 2020; Misra et al., 2020). This task is designed to be particularly difficult for *exploration*. We introduce two types of non-stationarity to the task, namely *abrupt* variations and *gradual* variations. A detailed discussion on the task settings as well as the configuration of the hyper-parameters is deferred to Appendix I. The cumulative rewards of the four algorithms in the abruptly-changing and gradually-changing environments are shown in Figures 1(a) and 1(b), respectively. All results are averaged over 30 runs.

As we can see, RestartQ-UCB outperforms Q-Learning UCB and Epsilon-Greedy under both types of environment variations. For the abruptly-changing environment as an example, RestartQ-UCB achieves 1.36 and 2.52 times of the cumulative rewards of Q-Learning UCB and Epsilon-Greedy, respectively. This demonstrates the importance of both addressing the environment variations (using restarts) and actively exploring the environment (using UCB-based bonus terms) in non-stationary RL. LSVI-UCB-Restart nearly matches the performance of RestartQ-UCB, which is unsurprising because both of them use the restarting strategy and optimistic exploration. Nevertheless, LSVI-UCB-Restart requires a higher time and space complexity. It needs to store all the history information in one epoch and solve a regularized least-squares minimization problem at every time step. This is indeed evidenced by our simulation results (shown in Figure 1(c)) that RestartQ-UCB only takes 0.18% of the computation time of LSVI-UCB-Restart.

Remark 2. *The heavy computation in LSVI-UCB-Restart mostly comes from the usage of a high-dimensional feature. In our simulations, we followed Example 2.1 in Jin et al. (2019) to convert a linear MDP algorithm to a tabular one, which results in a feature dimension of $d = S \times A$. This is essentially the most efficient feature encoding when no special structure is imposed on the tabular MDP. We believe that designing low-dimensional features for specific MDP instances can possibly reduce the computations for LSVI-UCB-Restart by a large amount, and is an interesting future direction for learning in linear MDPs per se.*

7. Application to Multi-Agent RL

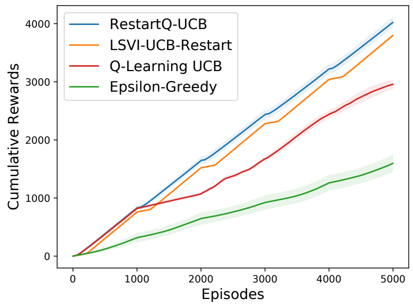
In this section, we discuss the application of our non-stationary RL method to multi-agent RL in episodic stochastic games (Shapley, 1953), which by nature leads to a non-stationary RL problem from one-agent's perspective.

7.1. Problem Setup

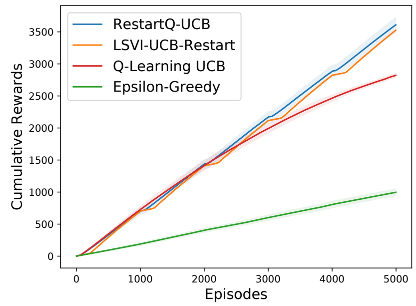
In general, an N-player episodic stochastic game is defined by a tuple $(\mathcal{N}, H, \mathcal{S}, \{\mathcal{A}^i\}_{i=1}^N, \{r^i\}_{i=1}^N, P)$, where (1) $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of agents; (2) $H \in \mathbb{N}_+$ is the number of time steps in each episode; (3) $\mathcal{S}$ is the finite state space; (4) $\mathcal{A}^i$ is the finite action space for agent $i \in \mathcal{N}$; (5) $r_h^i : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function at step $h \in [H]$ for agent $i \in \mathcal{N}$, where $\mathcal{A} = \times_{i=1}^N \mathcal{A}^i$; and (6) $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel at step $h \in [H]$, where the next state depends on the current state and the joint actions of all the agents. The game lasts for M episodes, and we let $T = MH$ be the total number of time steps. At each time step (m, h), the agents observe the state $s_h^m \in \mathcal{S}$, and take actions $a_h^{i,m} \in \mathcal{A}^i, i \in \mathcal{N}$ simultaneously. We let $a_h^m = (a_h^{1,m}, \ldots, a_h^{N,m})$. Agent i receives a reward with an expected value of $r_h^i(s_h^m, a_h^m)$, and the environment transitions to the next state $s_{h+1}^m \sim P_h(\cdot | s_h^m, a_h^m)$. For each agent i, a policy is a mapping from the time index and state space to (possibly a distribution over) the action space. We denote the set of policies for agent i by $\Pi^i = \{\pi^i : [M] \times [H] \times \mathcal{S} \to \Delta(\mathcal{A}^i)\}$. The set of joint policies are denoted by $\Pi = \times_{i=1}^N \Pi^i$. Each agent seeks to find a policy that maximizes its own reward.

For notational convenience, we consider two-player games, i.e., $N = 2$. We consider the problem where we can control the policy of agent 1, while agent 2 is an opponent that is adapting its own policy in an unknown way. Achieving sublinear regret in the face of an arbitrarily changing opponent is known to be computationally hard (Radanovic et al., 2019). Therefore, existing works (Radanovic et al., 2019; Lee et al., 2020) often focus on a setting where the opponent is only "slowly changing" its policy over time. One such example is when the opponent is using a relatively stable learning algorithm. We also focus on the *decentralized* setting[2], where each agent *cannot* observe the actions and rewards of the other agent. This is generally considered

[2]This setting has been studied under various names in the literature, including individual learning (Leslie & Collins, 2005), decentralized learning (Arslan & Yüksel, 2016), online agnostic learning (Tian et al., 2020), and independent learning (Daskalakis et al., 2020). It is also related to the broader category of teams and games with decentralized information structure (Ho, 1980; Nayyar et al., 2013a;b).



(a) Abrupt variations



(b) Gradual variations

Algorithm	Time per episode
RestartQ-UCB	0.102 ms
LSVI-UCB-Restart	57.65 ms
Q-Learning UCB	0.098 ms
Epsilon-Greedy	0.123 ms

(c) Time usage

Figure 1: Cumulative rewards of the four algorithms under (a) abrupt variations, and (b) gradual variations, respectively, as well as their (c) time usage. Shaded areas denote the standard deviations of rewards. Note that RestartQ-UCB significantly outperforms Q-Learning UCB and Epsilon-Greedy, and matches LSVI-UCB-Restart while being *much more* time-efficient.

to be a more practical multi-agent RL paradigm, and also more challenging than those that we will compare with in the literature (Radanovic et al., 2019; Lee et al., 2020).

A joint policy induces a probability measure on the sequence of states and joint actions. For a joint policy $\pi = (\pi^1, \pi^2) \in \Pi$, and for each time step $(m, h) \in [M] \times [H]$, state $s \in \mathcal{S}$, we define the state value function *for agent 1* as follows:

$$V_h^{m,\pi}(s) \stackrel{\text{def}}{=} \mathbb{E}\left[\sum_{h'=h}^{H} r^1\left(s_{h'}, \pi_{h'}^{1,m}\left(s_{h'}\right), \pi_{h'}^{2,m}\left(s_{h'}\right)\right) \mid s_h = s\right].$$

For a joint policy (π^1, π^2), we again evaluate the optimality of agent 1's policy π^1 in terms of its *dynamic regret*, which compares the agent's policy with the optimal policy of each individual episode in hindsight:

$$\mathcal{R}^{\pi^2}(\pi^1, M) \stackrel{\text{def}}{=} \sum_{m=1}^{M}\left(\sup_{\pi^{1\star}} V_1^{m,(\pi^{1\star},\pi^2)}(s_1^m) - V_1^{m,(\pi^1,\pi^2)}(s_1^m)\right).$$

The initial state of each episode s_1^m is again chosen by an oblivious adversary.

7.2. Regret Against a Slowly-Changing Opponent

We model the slowly-changing behavior of agent 2 by requiring it to have a low *switching cost* (Bai et al., 2019; Gao et al., 2021). This is a standard notion in the literature to measure the changing behavior of an RL algorithm. We consider the following definition of the (local) switching cost from Bai et al. (2019).

Definition 1. *The switching cost between any pair of policies (π, π') is the number of (h, s) pairs on which π and π' act differently:*

$$n_{\mathit{switch}}(\pi, \pi') \stackrel{\text{def}}{=} \left|\{(h, s) \in [H] \times \mathcal{S} : \pi_h(s) \neq \pi'_h(s)\}\right|.$$

For a policy trajectory $(\pi^1, \ldots, \pi^M)$ across M episodes[3]*, its switching cost is defined as* $N_{\mathit{switch}} \stackrel{\text{def}}{=} \sum_{m=1}^{M} n_{\mathit{switch}}(\pi^m, \pi^{m+1})$.

Bai et al. (2019) develops a learning algorithm that achieves a switching cost of $O(SAH^3 \log T)$, while Zhang et al. (2020) improves the switching cost to $O(SAH^2 \log T)$. For the sake of generality, we characterize the behavior of agent 2 by assuming that the switching cost of its policy trajectory is upper bounded by $O(T^\beta)$ for some $0 < \beta < 1$. Clearly, the two state-of-the-art RL algorithms mentioned above satisfy this upper bound. A direct application of RestartQ-UCB leads to the following result for agent 1:

Theorem 5. *Suppose that the switching cost of agent 2 satisfies $N_{\mathit{switch}} = O(T^\beta)$ for $0 < \beta < 1$. Let agent 1 run the RestartQ-UCB (Hoeffding/Freedman) algorithm. For T large enough, the dynamic regret of agent 1 is upper bounded by $\widetilde{O}(T^{\frac{\beta+2}{3}})$.*

7.3. Learning Team-Optimality

Theorem 5 can be readily applied to learning team-optimal policies in "smooth games", which is the setting considered in Radanovic et al. (2019). This corresponds to the setting where a team of agents learn to collaborate. Before we present our results, a few definitions are in order.

Definition 2. *A stochastic game is called a stochastic team (or simply a* team*) if there exists a reward function $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$ such that $r_h^i = r_h, \forall i \in \mathcal{N}, h \in [H]$.*

Definition 3. *A joint policy $\pi^\star = (\pi^{1\star}, \pi^{2\star}) \in \Pi$ is called* team-optimal *if*

$$V_h^{(\pi^{1\star},\pi^{2\star})}(s) = \sup_{\pi^1,\pi^2} V_h^{(\pi^1,\pi^2)}(s), \forall s \in \mathcal{S}, h \in [H],$$

where $V_h^{(\pi^1,\pi^2)}(s) \stackrel{\text{def}}{=} \mathbb{E}[\sum_{h'=h}^{H} r_{h'}(s_{h'}, \pi_{h'}^1(s_{h'}), \pi_{h'}^2(s_{h'})) \mid s_h = s]$ is the value function.

In a stochastic team, the agents share the same objective, and aim to maximize the team accumulated reward. Team

[3]Here, the superscript of π denotes the index of an episode, rather than the index of an agent.

optimality is achieved when the joint policy of the agents induces the highest possible accumulated reward.

Since we cannot control the behavior of agent 2, its behavior might be sub-optimal and drive us away from team-optimality. To avoid such scenarios, we impose a structural assumption that allows us to quantify the distance from optimality. In particular, we assume that the team is (λ, μ)-smooth, following the definition in Radanovic et al. (2019).

Definition 4. *(Adapted from Definition 1 in Radanovic et al. (2019)) A two-player stochastic team is (λ, μ)-smooth if there exists a pair of policies $(\pi^{1\star}, \pi^{2\star})$ such that for every policy pair (π^1, π^2) and every $h \in [H], s \in \mathcal{S}$:*

$$V_h^{(\pi^{1\star},\pi^{2\star})}(s) \geq V_h^{(\pi^1,\pi^2)}(s),$$
$$V_h^{(\pi^{1\star},\pi^2)}(s) \geq \lambda \cdot V_h^{(\pi^{1\star},\pi^{2\star})}(s) - \mu \cdot V^{(\pi^1,\pi^2)}(s).$$

The (λ, μ)-smoothness ensures that agent 2's sub-optimal behavior only has a bounded negative impact on the joint value. Our definition of smoothness is adapted from Radanovic et al. (2019), where the infinite-horizon average-reward setting is considered. We adapt it to the finite-horizon case. This notion of smoothness is motivated by the definition of smooth games in Roughgarden (2009); Syrgkanis et al. (2015), as stated in Radanovic et al. (2019).

Applying our RestartQ-UCB algorithm would lead to the following theorem, which implies that the time-average return of the agents converges to a $\frac{\lambda}{1+\mu}$ factor of the team-optimal value as T grows. This is the same factor as has been achieved in Radanovic et al. (2019).

Theorem 6. *Let π^2 denote the policy of agent 2, and suppose that the switching cost of agent 2 satisfies $N_{switch} = O(T^\beta)$ for $0 < \beta < 1$. Assume that the team problem is (λ, μ)-smooth. Let agent 1 run the RestartQ-UCB algorithm, and let π^1 denote its induced policy. For T large enough, the return of the algorithm is lower bounded by:*

$$\sum_{m=1}^{M} V_1^{(\pi^1,\pi^2)}(s_1^m) \geq \frac{\lambda}{1+\mu}\left[\sum_{m=1}^{M} V_1^{(\pi^{1\star},\pi^{2\star})}(s_1^m) - \widetilde{O}(T^{\frac{\beta+2}{3}})\right].$$

Remark 3. *(Comparison with Radanovic et al. (2019) and Lee et al. (2020).) It might first appear to the reader that our regret guarantee is weaker than the bounds of $O(T^{\max\{1-\frac{3}{7}\alpha,\frac{1}{4}\}})$ and $O(T^{\max\{1-\frac{3}{2}\alpha,0\}})$ given in Radanovic et al. (2019) and Lee et al. (2020), respectively, where α can be essentially translated[4] to $1-\beta$. However, we would like to emphasize that our setting significantly generalizes the other two works and is inherently more challenging due to the following facts: First, we are considering* a learning *problem where the transition and reward functions are unknown; the other two works essentially consider* planning *with a known MDP model. Second, we are using the more challenging* dynamic regret *as a measure of optimality, while the other two use the static regret. Third, we study* decentralized *learning, where each agent cannot observe the actions and rewards of the other agent; the algorithms proposed in the other two works critically rely on the* observation *of the other agent's* policies*.*

[4]The other two works model the slowly-changing behavior of agent 2 using the small "policy change magnitude" criterion. Our setting is in this sense not completely comparable with theirs.

Remark 4. *(Significance of model-freeness.) Decentralized multi-agent RL is generally only possible with model-free approaches (see, e.g., Arslan & Yüksel (2016); Tian et al. (2020); Daskalakis et al. (2020)); model-based methods proceed by explicitly estimating the transition and reward functions, which crucially relies on observing the other agents' actions. This further demonstrates the flexibility and significance of model-free methods, when one addresses the non-stationarity issues in multi-agent RL through the lens of non-stationary RL.*

8. Concluding Remarks

In this paper, we have considered model-free reinforcement learning in non-stationary episodic MDPs. We have proposed an algorithm named RestartQ-UCB that adopts a simple restarting strategy. RestartQ-UCB with Freedman-type bonus terms achieves a dynamic regret of $\widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}HT^{\frac{2}{3}})$, which nearly matches the information-theoretical lower bound $\Omega(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{2}{3}}T^{\frac{2}{3}})$. Numerical experiments have validated the advantages of RestartQ-UCB in terms of both cumulative rewards and computational efficiency. A multi-agent RL example has been considered as an application to illustrate the power of our method. An interesting future direction would be to close the $\widetilde{O}(H^{\frac{1}{3}})$ factor gap between the upper and lower bounds that we have established for the non-stationary RL problem. It would also be interesting to explore if non-stationary RL can be helpful in other multi-agent RL scenarios.

Acknowledgements

We thank Zihan Zhang for helpful discussions. Research of W.M, K.Z, and T.B. was supported in part by the US Army Research Laboratory (ARL) Cooperative Agreement W911NF-17-2-0196, in part by the Office of Naval Research (ONR) MURI Grant N00014-16-1-2710, and in part by the Air Force Office of Scientific Research (AFOSR) Grant FA9550-19-1-0353.

References

Abel, D., Jinnai, Y., Guo, S. Y., Konidaris, G., and Littman, M. Policy and value transfer in lifelong reinforcement learning. In *International Conference on Machine Learn-*

ing, pp. 20–29, 2018.

Agarwal, A., Henaff, M., Kakade, S., and Sun, W. PC-PG: Policy cover directed exploration for provable policy gradient learning. *arXiv preprint arXiv:2007.08459*, 2020.

Agrawal, S. and Jia, R. Learning in structured MDPs with convex cost functions: Improved regret bounds for inventory management. In *ACM Conference on Economics and Computation*, pp. 743–744, 2019.

Allesiardo, R., Féraud, R., and Maillard, O.-A. The non-stationary stochastic multi-armed bandit problem. *International Journal of Data Science and Analytics*, 3(4): 267–283, 2017.

Arora, R., Dekel, O., and Tewari, A. Deterministic MDPs with adversarial rewards and bandit feedback. In *Conference on Uncertainty in Artificial Intelligence*, pp. 93–101, 2012.

Arslan, G. and Yüksel, S. Decentralized Q-learning for stochastic teams and games. *IEEE Transactions on Automatic Control*, 62(4):1545–1558, 2016.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

Auer, P., Gajane, P., and Ortner, R. Adaptively tracking the best bandit arm with an unknown number of distribution changes. In *Conference on Learning Theory*, pp. 138–158, 2019.

Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pp. 263–272, 2017.

Bai, Y., Xie, T., Jiang, N., and Wang, Y.-X. Provably efficient Q-learning with low switching cost. In *Advances in Neural Information Processing Systems*, pp. 8004–8013, 2019.

Besbes, O., Gur, Y., and Zeevi, A. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in Neural Information Processing Systems*, pp. 199–207, 2014.

Besbes, O., Gur, Y., and Zeevi, A. Optimal exploration–exploitation in a multi-armed bandit problem with non-stationary rewards. *Stochastic Systems*, 9(4):319–337, 2019.

Bowling, M. and Veloso, M. Rational and convergent learning in stochastic games. In *International Joint Conference on Artificial Intelligence*, volume 17, pp. 1021–1026. Lawrence Erlbaum Associates Ltd, 2001.

Brunskill, E. and Li, L. Sample complexity of multi-task reinforcement learning. In *Uncertainty in Artificial Intelligence*, pp. 122, 2013.

Busoniu, L., Babuska, R., and De Schutter, B. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

Cai, H., Ren, K., Zhang, W., Malialis, K., Wang, J., Yu, Y., and Guo, D. Real-time bidding by reinforcement learning in display advertising. In *International Conference on Web Search and Data Mining*, pp. 661–670, 2017.

Chawla, S., Devanur, N. R., Karlin, A. R., and Sivan, B. Simple pricing schemes for consumers with evolving values. In *ACM-SIAM Symposium on Discrete Algorithms*, pp. 1476–1490, 2016.

Chen, C., Wei, H., Xu, N., Zheng, G., Yang, M., Xiong, Y., Xu, K., and Li, Z. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *AAAI Conference on Artificial Intelligence*, pp. 3414–3421, 2020.

Chen, Y., Lee, C.-W., Luo, H., and Wei, C.-Y. A new algorithm for non-stationary contextual bandits: Efficient, optimal, and parameter-free. *arXiv preprint arXiv:1902.00980*, 2019.

Cheung, W. C., Simchi-Levi, D., and Zhu, R. Hedging the drift: Learning to optimize under non-stationarity. *arXiv preprint arXiv:1903.01461*, 2019a.

Cheung, W. C., Simchi-Levi, D., and Zhu, R. Learning to optimize under non-stationarity. In *International Conference on Artificial Intelligence and Statistics*, pp. 1079–1087, 2019b.

Cheung, W. C., Simchi-Levi, D., and Zhu, R. Reinforcement learning for non-stationary Markov decision processes: The blessing of (more) optimism. *arXiv preprint arXiv:2006.14389*, 2020.

Daskalakis, C., Foster, D. J., and Golowich, N. Independent policy gradient methods for competitive reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Davis, T., Burch, N., and Bowling, M. Using response functions to measure strategy strength. In *AAAI Conference on Artificial Intelligence*, 2014.

Dick, T., Gyorgy, A., and Szepesvari, C. Online learning in Markov decision processes with changing cost sequences. In *International Conference on Machine Learning*, pp. 512–520, 2014.

Domingues, O. D., Ménard, P., Pirotta, M., Kaufmann, E., and Valko, M. A kernel-based approach to non-stationary reinforcement learning in metric spaces. *arXiv preprint arXiv:2007.05078*, 2020.

Fei, Y., Yang, Z., Wang, Z., and Xie, Q. Dynamic regret of policy optimization in non-stationary environments. *arXiv preprint arXiv:2007.00148*, 2020.

Freedman, D. A. On tail probabilities for martingales. *The Annals of Probability*, pp. 100–118, 1975.

Gajane, P., Ortner, R., and Auer, P. A sliding-window algorithm for Markov decision processes with arbitrarily changing rewards and transitions. *arXiv preprint arXiv:1805.10066*, 2018.

Gao, M., Xie, T., Du, S. S., and Yang, L. F. A provably efficient algorithm for linear Markov decision process with low switching cost. *arXiv preprint arXiv:2101.00494*, 2021.

Garivier, A. and Moulines, E. On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*, pp. 174–188, 2011.

Ho, Y.-C. Team decision theory and information structures. *Proceedings of the IEEE*, 68(6):644–654, 1980.

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

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pp. 4863–4873, 2018.

Jin, C., Jin, T., Luo, H., Sra, S., and Yu, T. Learning adversarial MDPs with bandit feedback and unknown transition. *arXiv preprint arXiv:1912.01192*, 2019.

Kaplanis, C., Shanahan, M., and Clopath, C. Continual reinforcement learning with complex synapses. In *International Conference on Machine Learning*, pp. 2497–2506, 2018.

Karnin, Z. S. and Anava, O. Multi-armed bandits: Competing with optimal sequences. In *Advances in Neural Information Processing Systems*, pp. 199–207, 2016.

Keskin, N. B. and Zeevi, A. Chasing demand: Learning and earning in a changing environment. *Mathematics of Operations Research*, 42(2):277–307, 2017.

Lee, C.-W., Luo, H., Wei, C.-Y., and Zhang, M. Linear last-iterate convergence for matrix games and stochastic games. *arXiv preprint arXiv:2006.09517v1*, 2020. Available at https://arxiv.org/pdf/2006.09517v1.pdf.

Leslie, D. S. and Collins, E. J. Individual Q-learning in normal form games. *SIAM Journal on Control and Optimization*, 44(2):495–514, 2005.

Littman, M. L. Markov games as a framework for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 157–163. 1994.

Lu, J., Yang, C., Gao, X., Wang, L., Li, C., and Chen, G. Reinforcement learning with sequential information clustering in real-time bidding. In *International Conference on Information and Knowledge Management*, pp. 1633–1641, 2019.

Luo, H., Wei, C.-Y., Agarwal, A., and Langford, J. Efficient contextual bandits in non-stationary worlds. In *Conference On Learning Theory*, pp. 1739–1776, 2018.

Lykouris, T., Simchowitz, M., Slivkins, A., and Sun, W. Corruption robust exploration in episodic reinforcement learning. *arXiv preprint arXiv:1911.08689*, 2019.

Ma, W. Improvements and generalizations of stochastic knapsack and Markovian bandits approximation algorithms. *Mathematics of Operations Research*, 43(3):789–812, 2018.

Mao, W., Zheng, Z., Wu, F., and Chen, G. Online pricing for revenue maximization with unknown time discounting valuations. In *International Joint Conference on Artificial Intelligence*, pp. 440–446, 2018.

Mao, W., Zhang, K., Xie, Q., and Başar, T. POLY-HOOT: Monte-Carlo planning in continuous space MDPs with non-asymptotic analysis. *arXiv preprint arXiv:2006.04672*, 2020.

Misra, D., Henaff, M., Krishnamurthy, A., and Langford, J. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. In *International Conference on Machine Learning*, pp. 6961–6971, 2020.

Nayyar, A., Gupta, A., Langbort, C., and Başar, T. Common information based Markov perfect equilibria for stochastic games with asymmetric information: Finite games. *IEEE Transactions on Automatic Control*, 59(3):555–570, 2013a.

Nayyar, A., Mahajan, A., and Teneketzis, D. Decentralized stochastic control with partial history sharing: A common information approach. *IEEE Transactions on Automatic Control*, 58(7):1644–1658, 2013b.

Neu, G., Antos, A., György, A., and Szepesvári, C. Online Markov decision processes under bandit feedback. In *Advances in Neural Information Processing Systems*, pp. 1804–1812, 2010.

Ortner, R., Gajane, P., and Auer, P. Variational regret bounds for reinforcement learning. In *Uncertainty in Artificial Intelligence*, pp. 81–90, 2019.

Osband, I. and Van Roy, B. On lower bounds for regret in reinforcement learning. *arXiv preprint arXiv:1608.02732*, 2016.

Padakandla, S. A survey of reinforcement learning algorithms for dynamically varying environments. *arXiv preprint arXiv:2005.10619*, 2020.

Radanovic, G., Devidze, R., Parkes, D., and Singla, A. Learning to collaborate in Markov decision processes. In *International Conference on Machine Learning*, pp. 5261–5270, 2019.

Roughgarden, T. Intrinsic robustness of the price of anarchy. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, pp. 513–522, 2009.

Shapley, L. S. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953.

Shortreed, S. M., Laber, E., Lizotte, D. J., Stroup, T. S., Pineau, J., and Murphy, S. A. Informing sequential clinical decision-making through reinforcement learning: An empirical study. *Machine learning*, 84(1-2):109–136, 2011.

Sun, Y., Yin, X., and Huang, F. Temple: Learning template of transitions for sample efficient multi-task RL. *arXiv preprint arXiv:2002.06659*, 2020.

Syrgkanis, V., Agarwal, A., Luo, H., and Schapire, R. E. Fast convergence of regularized learning in games. *Advances in Neural Information Processing Systems*, 28: 2989–2997, 2015.

Tekin, C. and Liu, M. Online algorithms for the multi-armed bandit problem with Markovian rewards. In *48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1675–1682, 2010.

Tian, Y., Wang, Y., Yu, T., and Sra, S. Provably efficient online agnostic learning in Markov games. *arXiv preprint arXiv:2010.15020*, 2020.

Tirinzoni, A., Poiani, R., and Restelli, M. Sequential transfer in reinforcement learning with a generative model. *arXiv preprint arXiv:2007.00722*, 2020.

Touati, A. and Vincent, P. Efficient learning in non-stationary linear markov decision processes. *arXiv preprint arXiv:2010.12870*, 2020.

Wang, J., Liu, Y., and Li, B. Reinforcement learning with perturbed rewards. *arXiv preprint arXiv:1810.01032*, 2018.

Watkins, C. J. C. H. Learning from delayed rewards. *PhD thesis, King's College, University of Cambridge*, 1989.

Yadkori, Y. A., Bartlett, P. L., Kanade, V., Seldin, Y., and Szepesvári, C. Online learning in Markov decision processes with adversarially chosen transition probability distributions. In *Advances in Neural Information Processing Systems*, pp. 2508–2516, 2013.

Yu, J. Y. and Mannor, S. Online learning in Markov decision processes with arbitrarily changing rewards and transitions. In *International Conference on Game Theory for Networks*, pp. 314–322. IEEE, 2009.

Zhang, K., Yang, Z., and Başar, T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635*, 2019.

Zhang, Z., Zhou, Y., and Ji, X. Almost optimal model-free reinforcement learning via reference-advantage decomposition. *arXiv preprint arXiv:2004.10019*, 2020.

Zhao, P., Zhang, L., Jiang, Y., and Zhou, Z.-H. A simple approach for non-stationary linear bandits. In *International Conference on Artificial Intelligence and Statistics*, volume 2020, 2020.

Zhou, H., Chen, J., Varshney, L. R., and Jagmohan, A. Nonstationary reinforcement learning with linear function approximation. *arXiv preprint arXiv:2010.04244*, 2020.