Breaking the Sample Complexity Barrier to Regret-Optimal Model-Free Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Achieving sample efficiency in online episodic reinforcement learning (RL) requires optimally balancing exploration and exploitation. When it comes to a finite-horizon episodic Markov decision process with S states, A actions and horizon length H, substantial progress has been made towards characterizing the minimal regret, which scales on the order of $\sqrt{H^2SAT}$ (modulo log factors) with T the total number of samples. While several competing solution paradigms have been proposed to minimize regret, they are either memory-inefficient, or fall short of optimality unless the sample size exceeds an enormous threshold (e.g., $S^6A^4\,\mathrm{poly}(H)$ for the model-free approach). To overcome such a sample size barrier to efficient RL, we design a new model-free algorithm, with space complexity $O(SAH)$, that achieves near-optimal regret as soon as the sample size exceeds the order of $SA\,\mathrm{poly}(H)$. This sample size requirement improves, by at least a factor of S^5A^3, upon any prior algorithm that is simultaneously regret-optimal and memory-efficient. Our algorithm leverages the recently-introduced idea of reference-advantage decomposition, and designs a *double estimator* with the aid of two coupled sequences to further mitigate bias in the sample-limited regime.

1 Introduction

Contemporary reinforcement learning (RL) has to deal with unknown environments with unprecedentedly large dimensionality. How to make the best use of samples in the face of high-dimensional state/action space lies at the core of modern RL practice. An ideal RL algorithm would learn to act favorably even when the number of available data samples scales sub-linearly in the ambient dimension of the model, i.e., the number of parameters needed to describe the transition dynamics of the environment. The challenge is further compounded when this task needs to be accomplished with limited memory.

Simultaneously achieving the desired sample and memory efficiency is particularly challenging when it comes to online episodic RL scenarios. In contrast to the simulator setting that permits sampling of any state-action pair, an agent in online episodic RL is only allowed to draw sample trajectories by executing a policy in the unknown Markov decision process (MDP), where the initial states are pre-assigned and might even be chosen by an adversary. Careful deliberation needs to be undertaken when deciding what policies to use for effective interaction with the unknown environment, how to optimally balance exploitation and exploration, and how to process and store the collected information intelligently without causing redundancy.

1.1 Regret-optimal model-free RL? A sample size barrier

In order to evaluate and compare the effectiveness of RL algorithms in high dimension, a recent body of works seeks to develop a finite-sample theoretical framework to analyze the algorithms of

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

interest, with the aim of delineating the dependency of algorithm performance on all salient problem parameters in a non-asymptotic fashion (Dann et al., 2017; Kakade, 2003). To facilitate discussion, let us take a moment to summarize the state-of-the-art theory for episodic finite-horizon MDPs with non-stationary transition kernels, focusing on minimizing accumulated regret — a metric that quantifies the performance difference between the learned policy and the true optimal policy — with the fewest number of samples. Here and throughout, we use S, A, and H to denote the number of states, the number of actions, and the horizon length of the MDP, respectively, and let T represent the sample size. In addition, the immediate reward gained at each time step is assumed to lie between 0 and 1.

Fundamental regret lower bound. Following the arguments in (Jaksch et al., 2010; Auer et al., 2002), the recent work Jin et al. (2018); Domingues et al. (2021) provided a fundamental lower bound on the expected total regret for this setting. Specifically, it claims that no matter what algorithm to use, one can construct an MDP such that the accumulated regret incurred by the algorithm necessarily exceeds the order of

$$\text{(lower bound)} \qquad \sqrt{H^2SAT}, \tag{1}$$

as long as $T \geq H^2SA$.[1] This sublinear regret lower bound in turn imposes a sampling limit if one wants to achieve ε average regret.

Model-based RL. Moving beyond the lower bound, let us examine the effectiveness of model-based RL — an approach that can be decoupled into a model estimation stage (i.e., estimating the transition kernel using available data) and a subsequent stage of planning/learning using the learned model (Jaksch et al., 2010; Azar et al., 2017; Efroni et al., 2019; Agrawal and Jia, 2017; Pacchiano et al., 2020). In order to ensure a sufficient degree of exploration, (Azar et al., 2017) came up with an algorithm UCB-VI that blends model-based learning and the optimism principle, which achieves a regret bound[2] $\widetilde{O}\big(\sqrt{H^2SAT}\big)$ that nearly attains the lower bound (1) for large T. Caution needs to be exercised, however, that the near optimality of this algorithm does not kick in until the sample size T surpasses

$$T \geq S^3AH^6,$$

a requirement that is significantly larger than the dimension of the underlying model. In addition, model-based algorithms typically require storing the estimated probability transition kernel, resulting in a space complexity that could be as high as $\widetilde{O}(S^2AH)$ (Azar et al., 2017).

Model-free RL. Another competing solution paradigm is the model-free approach, which circumvents the model estimation stage and attempts to learn the optimal values directly (Strehl et al., 2006; Jin et al., 2018; Bai et al., 2019; Yang et al., 2021). In comparison to the model-based approach, the model-free approach brings with it the promise of low space complexity, as it eliminates the need of storing a full description of the model. In fact, in a number of previous works (e.g., (Strehl et al., 2006; Jin et al., 2018)), an algorithm is called model-free only if its space complexity is $o(S^2AH)$ regardless of the sample size T.

• *Memory-efficient methods.* (Jin et al., 2018) proposed the first memory-efficient model-free algorithm — which is an optimistic variant of classical Q-learning — that guarantees a regret proportional to $\sqrt{T}$ and a space complexity $O(SAH)$. Compared to the lower bound (1), however, the regret bound in (Jin et al., 2018) is off by a factor of $\sqrt{H}$ and hence suboptimal for problems with long horizon. This drawback has recently been overcome in (Zhang et al., 2020b) by leveraging the idea of variance reduction (or the so-called "reference-advantage decomposition"). While the resulting regret matches the information-theoretic limit asymptotically, its optimality in the non-asymptotic regime has been settled only after the sample size T exceeds (see (Zhang et al., 2020b, Lemma 7))

$$T \geq S^6A^4H^{28},$$

a requirement that is even far more stringent than the one imposed by (Azar et al., 2017).

• *A memory-inefficient "model-free" variant.* The recent work (Menard et al., 2021) put forward a novel sample-efficient variant of Q-learning called UCB-M-Q, introducing a carefully chosen

[1] Given that a trivial upper bound on the regret is T, one needs to impose a lower bound $T \geq H^2SA$ in order for (1) to be meaningful.

[2] Here and throughout, we use the standard notation $f(n) = O(g(n))$ to indicate that $f(n)/g(n)$ is bounded above by a constant as n grows. The notation $\widetilde{O}(\cdot)$ resembles $O(\cdot)$ except that it hides any logarithmic scaling. The notation $f(n) = o(g(n))$ means that $\lim_{n\to\infty} f(n)/g(n) = 0$.

Algorithm	Regret	Range of sample sizes T that attain optimal regret	Space complexity
UCB-VI (Azar et al., 2017)	$\sqrt{H^2SAT} + H^4S^2A$	$[S^3AH^6, \infty)$	S^2AH
UCB-Q-Hoeffding (Jin et al., 2018)	$\sqrt{H^4SAT}$	never	SAH
UCB-Q-Bernstein (Jin et al., 2018)	$\sqrt{H^3SAT} + \sqrt{H^9S^3A^3}$	never	SAH
UCB2-Q-Bernstein (Bai et al., 2019)	$\sqrt{H^3SAT} + \sqrt{H^9S^3A^3}$	never	SAH
UCB-Q-Advantage (Zhang et al., 2020b)	$\sqrt{H^2SAT} + H^8S^2A^{\frac{3}{2}}T^{\frac{1}{4}}$	$[S^6A^4H^{28}, \infty)$	SAH
UCB-M-Q (Menard et al., 2021)	$\sqrt{H^2SAT} + H^4SA$	$[SAH^6, \infty)$	S^2AH
UCB-Q-Double-Advantage **(this work)**	$\sqrt{H^2SAT} + H^6SA$	$[SAH^{10}, \infty)$	SAH
lower bound (Jin et al., 2018)	$\sqrt{H^2SAT}$	n/a	n/a

Table 1: Comparisons between prior art and our results for non-stationary episodic MDPs when $T \geq H^2SA$. The table includes the order of the regret bound, the range of sample sizes that achieve the optimal regret $\sqrt{H^2SAT}$, and the memory complexity, with all logarithmic factors omitted for simplicity of presentation. The red text highlights the suboptimal part of the algorithms.

momentum term for bias reduction. This algorithm is guaranteed to yield a near-optimal regret $\widetilde{O}\big(\sqrt{H^2SAT}\big)$ as soon as the sample size exceeds $T \geq SAH^6$, which is a remarkable improvement vis-à-vis previous regret-optimal methods (Azar et al., 2017; Zhang et al., 2020b). Nevertheless, akin to the model-based approach, it comes at a price in terms of the space and computation complexities, as the space required to store all bias-value function is $O(S^2AH)$ and the computation required is $O(ST)$, both of which are larger by a factor of S than model-free algorithms like (Zhang et al., 2020b). In view of this memory inefficiency, UCB-M-Q falls short of fulfilling the definition of model-free algorithms in (Strehl et al., 2006; Jin et al., 2018). See (Menard et al., 2021, Section 3.3) for more detailed discussions.

A more complete summary of prior results is provided in Table 1.

1.2 A glimpse of our contributions

In brief, while it is encouraging to see that both model-based and model-free approaches allow for near-minimal regret as T tends to infinity, they are either memory-inefficient, or require the sample size to exceed a threshold substantially larger than the model dimensionality. In fact, no prior algorithms have been shown to be *simultaneously regret-optimal and memory-efficient* unless

$$T \geq S^6A^4\,\mathrm{poly}(H),$$

which constitutes a stringent sample size barrier constraining their utility in the sample-starved and memory-limited regime. The presence of this sample complexity barrier motivates one to pose a natural question: *Is it possible to design an algorithm that accommodates a significantly broader sample size range without compromising regret optimality and memory efficiency?*

In this paper, we answer this question affirmatively, by designing a new model-free algorithm, dubbed as UCB-Q-Double-Advantage, that enjoys the following performance guarantee.

Theorem 1 (informal)**.** *The proposed* UCB-Q-Double-Advantage *algorithm, which has a space complexity $O(SAH)$, achieves a near-optimal regret $\widetilde{O}\big(\sqrt{H^2SAT}\big)$ as soon as the sample size exceeds $T \gtrsim SA\,\mathrm{poly}(H)$.*

As can be seen from Table 1, the space complexity of the proposed algorithm is $O(SAH)$, which is far more memory-efficient than both the model-based approach in (Azar et al., 2017) and the UCB-M-Q algorithm in (Menard et al., 2021), both of which require S^2AH units of space. In addition, the sample size requirement $T \geq SA\,\mathrm{poly}(H)$ of our algorithm substantially improves, by a factor of at least S^5A^3, upon that of any prior algorithm that is simultaneously regret-optimal and memory-efficient. In fact, this requirement is nearly sharp in terms of the dependency on both S and A, and was previously achieved only by the UCB-M-Q algorithm at a price of a higher storage burden.

Let us also briefly point out the key ideas of our algorithm. As an optimistic variant of variance-reduced Q-learning, UCB-Q-Double-Advantage leverages the recently-introduced reference-

advantage decompositions for variance reduction (Zhang et al., 2020b), but employs a *double estimator* with the aid of two coupled sequences to further reduce bias. The design of our novel double estimator, as well as its analysis framework, might be of independent interest to other settings that involve managing intricate exploration-exploitation trade-offs.

2 Problem formulation

In this section, we formally describe the problem setting. Throughout, we denote by $\Delta(\mathcal{S})$ the probability simplex over a set $\mathcal{S}$, and denote $[M] \coloneqq \{1, \cdots, M\}$ for any integer $M > 0$.

Basics of finite-horizon MDPs. Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \{P_h\}_{h=1}^H, \{r_h\}_{h=1}^H)$ represent a finite-horizon MDP, where $\mathcal{S} \coloneqq \{1, \cdots S\}$ is the state space of size S, $\mathcal{A} \coloneqq \{1, \cdots, A\}$ is the action space of size A, H is denotes the horizon, and $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ (resp. $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$) represents the probability transition kernel (resp. reward function) at the h-th time step, $1 \le h \le H$, respectively. More specifically, $P_h(\cdot \mid s, a) \in \Delta(\mathcal{S})$ stands for the transition probability vector from state s when action a is taken at time step h, while $r_h(s, a)$ indicates the immediate reward received at time step h for a state-action pair (s, a) (which is assumed to be deterministic and fall within the range $[0, 1]$). A policy of an agent is represented as $\pi = \{\pi_h\}_{h=1}^H$ with $\pi_h : \mathcal{S} \mapsto \mathcal{A}$ the action selection rule at time step h, so that $\pi_h(s)$ specifies which action to execute in state s at time step h. Throughout this paper, we concentrate on deterministic policies.

Value functions, Q-functions, and Bellman equations. The value function $V_h^\pi(s)$ of a policy π at step h is defined as the expected cumulative rewards received between steps h and H when executing this policy from an initial state s at step h, i.e.,

$$V_h^\pi(s) = \mathop{\mathbb{E}}_{s_{t+1} \sim P_t(\cdot \mid s_t, \pi_t(s_t)), t \ge h} \left[\sum_{t=h}^H r_t\big(s_t, \pi_t(s_t)\big) \,\middle|\, s_h = s \right], \tag{2}$$

where the expectation is taken over the randomness of the MDP trajectory $\{s_t \mid h \le t \le H\}$. The action-value function (a.k.a. the Q-function) $Q_h^\pi(s, a)$ of a policy π at step h can be defined analogously except that the action at step h is fixed to be a, that is,

$$Q_h^\pi(s, a) = r_h(s, a) + \mathop{\mathbb{E}}_{\substack{s_{h+1} \sim P_h(\cdot \mid s, a), \\ s_{t+1} \sim P_t(\cdot \mid s_t, \pi_t(s_t)), t > h}} \left[\sum_{t=h+1}^H r_t(s_t, \pi_t(s_t)) \,\middle|\, s_h = s, a_h = a \right]. \tag{3}$$

In addition, we define $V_{H+1}^\pi(s) = Q_{H+1}^\pi(s, a) = 0$ for any policy π and any $(s, a) \in \mathcal{S} \times \mathcal{A}$. By virtue of basic properties in dynamic programming (Bertsekas, 2017) the value functions and the Q-function satisfy the following Bellman equation:

$$Q_h^\pi(s, a) = r_h(s, a) + \mathbb{E}_{s' \sim P_h(\cdot \mid s, a)} \big[V_{h+1}^\pi(s') \big]. \tag{4}$$

A policy $\pi^\star = \{\pi_h^\star\}_{h=1}^H$ is said to be an optimal policy if it maximizes the value function simultaneously for all states among all policies. The resulting optimal value function $V^\star = \{V_h^\star\}_{h=1}^H$ and optimal Q-functions $Q^\star = \{Q_h^\star\}_{h=1}^H$ satisfy

$$V_h^\star(s) = V_h^{\pi^\star}(s) = \max_\pi V_h^\pi(s) \qquad \text{and} \qquad Q_h^\star(s, a) = Q_h^{\pi^\star}(s, a) = \max_\pi Q_h^\pi(s, a) \tag{5}$$

for any $(s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$. It is well known that the optimal policy always exists Puterman (2014), and satisfies the Bellman optimality equation:

$$\forall (s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H] : \qquad Q_h^\star(s, a) = r_h(s, a) + \mathbb{E}_{s' \sim P_h(\cdot \mid s, a)} \big[V_{h+1}^\star(s') \big]. \tag{6}$$

Online episodic RL. This paper investigates the online episodic RL setting, where the agent is allowed to execute the MDP sequentially in a total number of K episodes each of length H. This amounts to collecting

$$T = KH \text{ samples}$$

in total. More specifically, in each episode $k = 1, \ldots, K$, the agent is assigned an arbitrary initial state s_1^k (possibly by an adversary), and selects a policy $\pi^k = \{\pi_h^k\}_{h=1}^H$ learned based on the information collected up to the $(k-1)$-th episode. The k-th episode is then executed following the policy π^k and the dynamic of the MDP $\mathcal{M}$, leading to an H-length sample trajectory.

Goal: regret minimization. In order to evaluate the quality of the learned policies $\{\pi^k\}_{1\le k\le K}$, a frequently used performance metric is the cumulative regret defined as follows:

$$\mathsf{Regret}(T) := \sum_{k=1}^{K} \big(V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k)\big), \tag{7}$$

where $K = T/H$. In words, the regret reflects the aggregate sub-optimality gap between the values of the optimal policy and those of the learned policies. A natural objective is thus to design a sample-optimal algorithm, namely, an algorithm whose resulting regret scales optimally in the sample size T. Accomplishing this goal requires carefully managing the trade-off between exploration and exploitation, which is particularly challenging in the sample-limited regime.

Notation. Before embarking on our main results, we take a moment to introduce some convenient notation to be used throughout the remainder of this paper. For any vector $x \in \mathbb{R}^{SA}$ that constitutes certain values for each of the state-action pairs, we shall often use $x(s,a)$ to denote the entry associated with the (s,a) pair. We let

$$P_{h,s,a} = P_h(\cdot \mid s,a) \in \mathbb{R}^{1\times S} \tag{8}$$

denote the row of P_h corresponding to the (s,a) pair. Additionally, we denote by e_i the i-th standard basis vector, with the only non-zero element being in the i-th entry.

3 Algorithm and theoretical guarantees

In this section, we present the proposed algorithm called UCB-Q-Double-Advantage, as well as the accompanying theory confirming its sample and memory efficiency.

3.1 Review: Q-learning with UCB exploration and reference advantage

This subsection briefly reviews the Q-learning algorithm with UCB exploration proposed in (Jin et al., 2018), as well as a variant that further exploits the idea of variance reduction (Zhang et al., 2020b). These two model-free algorithms inspire the algorithm design in the current paper.

Q-learning with UCB exploration (UCB-Q). Recall that the classical Q-learning algorithm has been proposed as a stochastic approximation scheme to solve the Bellman optimality equation (6), which consists of the following update rule (Watkins, 1989; Watkins and Dayan, 1992):

$$Q_h(s,a) \leftarrow (1-\eta)Q_h(s,a) + \eta\Big\{r_h(s,a) + \underbrace{\widehat{P}_{h,s,a}V_{h+1}}_{\text{stochastic estimate of } P_{h,s,a}V_{h+1}}\Big\}. \tag{9}$$

Here, Q_h (resp. V_h) indicates the running estimate of $Q_h^\star$ (resp. $V_h^\star$), η is the (possibly iteration-varying) learning rate, and $\widehat{P}_{h,s,a}V_{h+1}$ is a stochastic estimate of $P_{h,s,a}V_{h+1}$ (cf. (8)). For instance, if one has a sample (s,a,s') transitioning from state s at step h to s' at step $h+1$ under action a, then an unbiased estimate of $P_{h,s,a}V_{h+1}$ can be taken as $V_{h+1}(s')$. To further encourage exploration, the algorithm proposed in Jin et al. (2018) — which shall be abbreviated as UCB-Q hereafter — augments the Q-learning update rule (9) in each episode via an additional exploration bonus:

$$Q_h^{\mathsf{UCB}}(s,a) \leftarrow (1-\eta)Q_h^{\mathsf{UCB}}(s,a) + \eta\big\{r_h(s,a) + \widehat{P}_{h,s,a}V_{h+1} + b_h\big\}. \tag{10}$$

The bonus term $b_h \ge 0$ is chosen to be a certain confidence bound for $(\widehat{P}_{h,s,a} - P_{h,s,a})V_{h+1}$, an exploration-efficient scheme originated from the bandit literature (Lai and Robbins, 1985; Lattimore and Szepesvári, 2020). The algorithm then proceeds to the next episode by executing/sampling the MDP using a greedy policy w.r.t. the updated Q-estimate. These steps are repeated until the algorithm is terminated.

Q-learning with UCB exploration and reference advantage (UCB-Q-Advantage). The regret bounds derived for UCB-Q (Jin et al., 2018), however, as they are at least a factor of $\sqrt{H}$ from the fundamental lower bound, falling short of being optimal . In order to further reduce regret, one strategy is to leverage the idea of variance reduction to accelerate convergence (Johnson and Zhang, 2013; Sidford et al., 2018b; Wainwright, 2019b; Zhang et al., 2020b; Li et al., 2020b). An instantiation of this idea for the regret setting is a variant of UCB-Q based on reference-advantage decomposition,

which was put forward in Zhang et al. (2020b) and shall be abbreviated as UCB-Q-Advantage throughout this paper.

To describe the key ideas of UCB-Q-Advantage, imagine that we are able to maintain a collection of reference values $V^{\mathsf{R}} = \{V_h^{\mathsf{R}}\}_{h=1}^H$, which form reasonable estimates of $V^\star = \{V_h^\star\}_{h=1}^H$ and become increasingly more accurate as the algorithm progresses. The algorithm proceeds in an epoch-based fashion, and the reference values $\{V_h^{\mathsf{R}}\}$ are updated only at the beginning of each epoch (as opposed to each episode). At each time step h, the algorithm adopts the following update rule

$$Q_h^{\mathsf{R}}(s,a) \leftarrow (1-\eta)Q_h^{\mathsf{R}}(s,a) + \eta\Big\{r_h(s,a) + \underbrace{\widehat{P}_{h,s,a}\big(V_{h+1} - V_{h+1}^{\mathsf{R}}\big)}_{\text{stochastic estimate of } P_{h,s,a}\left(V_{h+1} - V_{h+1}^{\mathsf{R}}\right)} + \big[\mathsf{estimate}\big(P_h V_{h+1}^{\mathsf{R}}\big)\big](s,a) + b_h^{\mathsf{R}}\Big\}, \tag{11}$$

two ingredients of which are discussed below.

• Akin to the UCB-Q case, we can take $\widehat{P}_{h,s,a}\big(V_{h+1} - V_{h+1}^{\mathsf{R}}\big)$ to be the stochastic estimate $V_{h+1}(s') - V_{h+1}^{\mathsf{R}}(s')$ if we observe a sample transition (s,a,s') at time step h. If V_{h+1} is fairly close to the reference V_{h+1}^{R}, then this stochastic term can be less volatile than the stochastic term $\widehat{P}_{h,s,a}V_{h+1}$ in (10).

• Additionally, the term $\mathsf{estimate}\big(P_h V_{h+1}^{\mathsf{R}}\big)$ indicates an estimate of the one-step look-ahead value $P_h V_{h+1}^{\mathsf{R}}$, which is computed at the beginning of each epoch using a batch of samples but stays fixed throughout the rest of the epoch. The variability of $\mathsf{estimate}\big(P_h V_{h+1}^{\mathsf{R}}\big)$ can be well-controlled given the use of batch data.

The exploration bonus term b_h^{R} is thus taken to be a confidence bound for these two terms combined. Given that the uncertainty of (11) largely stems from the above two terms (which can both be much smaller than the variability in (10)), the incorporation of the reference term helps accelerate convergence.

3.2 The proposed algorithm: UCB-Q-Double-Advantage

As alluded to previously, however, the sample size required for UCB-Q-Advantage to achieve optimal regret needs to exceed a large polynomial S^6A^4 in the size of the state/action space. To overcome this sample complexity barrier, we introduce an improved variant called UCB-Q-Double-Advantage.

Motivation: early settlement is good. An important insight in the previous algorithm design is that: to retain low regret, it is desirable to maintain an estimate Q function that (i) provides an optimistic view (namely, an over-estimate) of the truth $Q^\star$, and (ii) mitigates the bias $Q - Q^\star$ as much as possible. With two additional optimistic Q-estimates in hand — Q_h^{UCB} based on UCB-Q, and a reference Q_h^{R} — it is natural to combine them as follows to further reduce the bias without violating the optimism principle:

$$Q_h(s_h,a_h) \leftarrow \min\big\{Q_h^{\mathsf{R}}(s_h,a_h), Q_h^{\mathsf{UCB}}(s_h,a_h), Q_h(s_h,a_h)\big\}. \tag{12}$$

This is precisely what is conducted in UCB-Q-Advantage. However, while the estimate Q_h^{R} obtained with the aid of reference-advantage decomposition provides great promise, fully realizing its potential in the sample-limited regime critically relies on the ability to *settle on* a desirable "reference" during the early stage of the algorithm. However, we are facing a dilemma: on the one hand, the reference value V^{R} needs to be updated in a timely manner in order to better control the stochastic estimate of $P_{h,s,a}\big(V_{h+1} - V_{h+1}^{\mathsf{R}}\big)$; on the other hand, updating V^{R} too frequently incurs an overly large sample size burden, since loosely speaking, sufficient samples need to be accumulated from scratch to estimate $P_h V_{h+1}^{\mathsf{R}}$ whenever V^{R} is updated, which contribute to a high sample size barrier.

Built upon the above insights, let us consider a fictitious solution (which shall be solidified by our analysis) that mitigates bias in the sample-limited regime. This consists of updating V^{R} as follows whenever s_h is visited:[3]

$$V_h^{\mathsf{R}}(s_h) \leftarrow \begin{cases} V_h(s_h), & \text{if } V_h^\star(s_h) + 1 < V_h(s_h) < V_h^{\mathsf{R}}(s_h) - 1, \\ V_h^{\mathsf{R}}(s_h), & \text{otherwise,} \end{cases} \tag{13}$$

[3]The margin 1 in (13) is selected arbitrarily and can be changed to a different constant.

Algorithm 1: UCB-Q-Double-Advantage

Initialize $Q_h(s,a), Q_h(s,a), Q_h^{\mathsf{R}_1}(s,a), Q_h^{\mathsf{R}_2}(s,a) \leftarrow H$; $N_h(s,a) \leftarrow 0$; and $V_h(s), V_h^{\mathsf{R}_1}(s), V_h^{\mathsf{R}_2}(s) \leftarrow H$ for all $(s,a,h) \in \mathcal{S} \times \mathcal{A} \times [H]$.
for *Episode* $k = 1$ **to** K **do**
  Set initial state $s_1 \leftarrow s_1^k$.
  for *Step* $h = 1$ **to** H **do**
    Take action $a_h = \pi_h^k(s_h) = \arg\max_a Q_h(s_h, a)$, and draw $s_{h+1} \sim P_h(\cdot \mid s_h, a_h)$.
    $N_h(s_h, a_h) \leftarrow N_h(s_h, a_h) + 1$; $n \leftarrow N_h(s_h, a_h)$. `// update the counter`
    $\eta_n \leftarrow \frac{H+1}{H+n}$. `// update the learning rate`
    $Q_h^{\mathsf{UCB}}(s_h, a_h) \leftarrow$ `update-ucb-q`$\big(Q_h^{\mathsf{UCB}}(s_h, a_h), V_{h+1}(s_{h+1})\big)$. `// run UCB-Q; see Algorithm 2`
    for *Index* $i = 1$ **to** 2 **do**
      `/* estimate` $Q_h^{\mathsf{R}_i}$ `and associated parameters` $W_h^{\mathsf{R}_i}$`; see Algorithm 2 */`
      $[Q_h^{\mathsf{R}_i}, W_h^{\mathsf{R}_i}](s_h, a_h) \leftarrow$ `update-ucb-q-advantage`$\big([Q_h^{\mathsf{R}_i}, W_h^{\mathsf{R}_i}](s_h, a_h), [V_{h+1}, V_{h+1}^{\mathsf{R}_i}](s_{h+1})\big)$.
    `/* update Q-estimates using all estimates, and update value estimates */`
    $Q_h(s_h, a_h) \leftarrow \min\big\{Q_h^{\mathsf{R}_1}(s_h, a_h),\ Q_h^{\mathsf{R}_2}(s_h, a_h),\ Q_h^{\mathsf{UCB}}(s_h, a_h),\ Q_h(s_h, a_h)\big\}$;
    $V_h(s_h) \leftarrow \max_a Q_h(s_h, a)$.
    `/* update reference values */`
    if $V_h(s_h) < V_h^{\mathsf{R}_1}(s_h) - 1$ **then**
      $V_h^{\mathsf{R}_2}(s_h) \leftarrow V_h^{\mathsf{R}_1}(s_h)$ and $V_h^{\mathsf{R}_1}(s_h) \leftarrow V_h(s_h)$;
      $[Q_h^{\mathsf{R}_2}, W_h^{\mathsf{R}_2}](s_h, a_h) \leftarrow [Q_h^{\mathsf{R}_1}, W_h^{\mathsf{R}_1}](s_h, a_h)$.

where $V_h(\cdot) = \max_a Q_h(\cdot, a)$ is the current value estimate for time step h. In words, V_h^{R} is updated to the current estimate V_h when

- there is a sufficient gain in doing so (i.e., $V_h^{\mathsf{R}}(s_h) > V_h(s_h) + 1$ so that V_h^{R} is more biased than V_h), and
- V_h is still not too close to the truth (i.e., $V_h(s_h) > V_h^{\star}(s_h) + 1$).

When either of these two conditions is violated, we claim that it is unnecessary to update the reference V_h^{R}, since it is of sufficient quality and further drifting the reference is no longer beneficial.

In truth, (13) suggests that the reference value V_h^{R} generally stops being updated quickly and converges to a reasonable point $V^{\mathsf{R},\mathsf{final}}$; for instance, once V^{R} reaches the region

$$V_h^{\mathsf{R}}(s_h) < V_h^{\star}(s_h) + 2, \tag{14}$$

then the update rule (13) terminates further updates of V_h^{R}. Locking on a reasonable reference value early on means that fewer samples are wasted on estimating a drifting target $(P_h V_{h+1}^{\mathsf{R}})(s_h, a_h)$, and all ensuing samples can be dedicated to estimating the key quantity of interest $\big(P_h V_{h+1}^{\mathsf{R},\mathsf{final}}\big)(s_h, a_h)$.

Unfortunately, the update rule (13) is not practically implementable, given that it relies crucially on the unavailable knowledge of the ground truth $V_h^{\star}$. Therefore, the task boils down to devising a practical scheme that mimics the update rule (13).

The proposed UCB-Q-Double-Advantage **algorithm.** To circumvent the difficulty of implementing (13), we propose to maintain two reference values V^{R_1} and V^{R_2} (which will lead to two Q-estimates Q^{R_1} and Q^{R_2} that are updated similarly except the use of different reference values and associated parameters). The update rule (12) is modified accordingly to

$$Q_h(s_h, a_h) \leftarrow \min\big\{Q_h^{\mathsf{R}_1}(s_h, a_h), Q_h^{\mathsf{R}_2}(s_h, a_h), Q_h^{\mathsf{UCB}}(s_h, a_h), Q_h(s_h, a_h)\big\}, \tag{15}$$

where $Q_h^{\mathsf{UCB}}(s_h, a_h)$ is updated using the subroutine `update-ucb-q` (following the standard Q-learning with Hoeffding bonus (Jin et al., 2018)). Maintaining two value functions provides a sort of diversity that at least one of the reference values is well-behaved in any given iteration. In effect, the

Algorithm 2: Some auxiliary functions

Function `update-ucb-q`$(Q_h^{\mathsf{UCB}}(s_h,a_h), V_{h+1}(s_{h+1}))$**:**

$\quad b_h \leftarrow c_{\mathrm{b}}\sqrt{\frac{H^3 \log \frac{SAT}{\delta}}{n}}$;

$\quad Q_h^{\mathsf{UCB}}(s_h,a_h) \leftarrow (1-\eta_n)Q_h^{\mathsf{UCB}}(s_h,a_h) + \eta_n\big(r(s_h,a_h) + V_{h+1}(s_{h+1}) + b_h\big)$.

Function `update-ucb-q-advantage`$([Q_h^{\mathsf{R}}, W_h^{\mathsf{R}}](s_h,a_h), [V_{h+1}, V_{h+1}^{\mathsf{R}}](s_{h+1}))$**:**

`/* update the moment statistics of` V_h^{R} `*/`

$\quad [\mu_h^{\mathsf{ref,R}}, \sigma_h^{\mathsf{ref,R}}, \mu_h^{\mathsf{adv,R}}, \sigma_h^{\mathsf{adv,R}}](s_h,a_h) \leftarrow$
$\quad$ `update-moments`$\big([\mu_h^{\mathsf{ref,R}}, \sigma_h^{\mathsf{ref,R}}, \mu_h^{\mathsf{adv,R}}, \sigma_h^{\mathsf{adv,R}}](s_h,a_h), [V_{h+1}, V_{h+1}^{\mathsf{R}}](s_{h+1})\big)$;

`/* update the accumulative bonus and bonus difference */`

$\quad [\delta_h^{\mathsf{R}}, B_h^{\mathsf{R}}](s_h,a_h) \leftarrow$ `update-bonus`$\big([\mu_h^{\mathsf{ref,R}}, \sigma_h^{\mathsf{ref,R}}, \mu_h^{\mathsf{adv,R}}, \sigma_h^{\mathsf{adv,R}}, B_h^{\mathsf{R}}](s_h,a_h)\big)$;

$\quad b_h^{\mathsf{R}} \leftarrow B_h^{\mathsf{R}}(s_h,a_h) + (1-\eta_n)\frac{\delta_h^{\mathsf{R}}(s_h,a_h)}{\eta_n} + c_{\mathrm{b}}\frac{H^3 \log \frac{SAT}{\delta}}{n}$;

`/* update the Q-estimate based on reference-advantage decomposition */`

$\quad Q_h^{\mathsf{R}}(s_h,a_h) \leftarrow$
$\quad (1-\eta_n)Q_h^{\mathsf{R}}(s_h,a_h) + \eta_n\big(r_h(s_h,a_h) + V_{h+1}(s_{h+1}) - V_{h+1}^{\mathsf{R}}(s_{h+1}) + \mu_h^{\mathsf{ref,R}}(s_h,a_h) + b_h^{\mathsf{R}}\big)$;

`/* update the associated parameters */`

$\quad W_h^{\mathsf{R}}(s_h,a_h) \leftarrow [\mu_h^{\mathsf{ref,R}}, \sigma_h^{\mathsf{ref,R}}, \mu_h^{\mathsf{adv,R}}, \sigma_h^{\mathsf{adv,R}}, \delta_h^{\mathsf{R}}, B_h^{\mathsf{R}}](s_h,a_h)$.

updates of the references (lines 14-15 of Algorithm 1) are designed to terminate early (which will be made rigorous in our analysis); for instance, if

$$V_h^{\mathsf{R}_2}(s_h) < V_h^{\star}(s_h) + 2, \tag{16}$$

then further updates of $V_h^{\mathsf{R}_2}$ terminate by checking the condition of $V_h^{\mathsf{R}_1}(s_h)$ instead of $V_h^{\star}(s_h)$ (see Remark 1). Therefore, this double estimator mimics the ideal design V^{R} discussed above (cf. (14)).

Remark 1. To elaborate a bit more, we provide an intuitive argument as follows. Suppose for the moment that $V_h \geq V_h^{\star}$ is always satisfied due to the optimism principle. At the end of each time step, denoting by $V_h^{\mathsf{R}_1,\mathsf{past}}$ the value of $V_h^{\mathsf{R}_1}$ before being updated, once the update condition for the reference values (line 13 of Algorithm 1) is activated, then the update rules (cf. lines 14-15 of Algorithm 1) guarantee (16) by virtue of the following fact: according to the update rules, when $V_h^{\mathsf{R}_2}(s_h) = V_h^{\mathsf{R}_1,\mathsf{past}}(s_h) < V_h^{\star}(s_h) + 2$, one has $V_h^{\mathsf{R}_1}(s_h) = V_h(s_h) < V_h^{\mathsf{R}_1,\mathsf{past}}(s_h) - 1 < V_h^{\star}(s_h) + 1$. This leads to the reference values not being updated afterwards, given that the update condition (line 13 of Algorithm 1) $V_h(s_h) < V_h^{\mathsf{R}_1}(s_h) - 1 < V_h^{\star}(s_h)$ will never be satisfied from now on (otherwise it violates the optimism principle).

Q-learning with reference-advantage decomposition. The rest of this subsection is devoted to explaining the subroutine `update-ucb-q-advantage`, which produces a Q-estimate Q^{R} based on the reference-advantage decomposition similar to (Zhang et al., 2020b). To facilitate the implementation, let us introduce the parameters associated with a reference value V^{R}, which will be denoted by $W^{\mathsf{R}} = \{W_h^{\mathsf{R}}\}_{h=1}^H$. Each W_h^{R} comprises six different components, i.e.,

$$W_h^{\mathsf{R}} := \big[\mu_h^{\mathsf{ref,R}}, \sigma_h^{\mathsf{ref,R}}, \mu_h^{\mathsf{adv,R}}, \sigma_h^{\mathsf{adv,R}}, \delta_h^{\mathsf{R}}, B_h^{\mathsf{R}}\big] \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times 6}, \tag{17}$$

where each column is a vector constituting values over all state-action pairs. For any $(s,a,h) \in \mathcal{S}\times\mathcal{A}\times[H]$, we denote by $W_h^{\mathsf{R}}(s,a)$ the (s,a)-th row of W_h^{R}, i.e.,

$$W_h^{\mathsf{R}}(s,a) = \big[\mu_h^{\mathsf{ref,R}}(s,a), \sigma_h^{\mathsf{ref,R}}(s,a), \mu_h^{\mathsf{adv,R}}(s,a), \sigma_h^{\mathsf{adv,R}}(s,a), \delta_h^{\mathsf{R}}(s,a), B_h^{\mathsf{R}}(s,a)\big], \tag{18}$$

where $\mu_h^{\mathsf{ref,R}}(s,a)$ and $\sigma_h^{\mathsf{ref,R}}(s,a)$ estimate the running mean and 2nd moment of the reference $\big[P_h V_{h+1}^{\mathsf{R}}\big](s,a)$; $\mu_h^{\mathsf{adv,R}}(s,a)$ and $\sigma_h^{\mathsf{adv,R}}(s,a)$ estimate the running mean and 2nd moment of the advantage $\big[P_h(V_{h+1} - V_{h+1}^{\mathsf{R}})\big](s,a)$; $B_h^{\mathsf{R}}(s,a)$ aggregates the empirical standard deviations of the reference and the advantage; and last but not least, $\delta_h^{\mathsf{R}}(s,a)$ is the temporal difference between $B_h^{\mathsf{R}}(s,a)$ and its previous value.

As alluded to earlier, the Q-value estimation follows the scheme (11) at a high level. Upon observing a sample transition (s_h, a_h, s_{h+1}), we compute the following estimates to update $Q^{\mathsf{R}}(s_h,a_h)$:

• the term $\widehat{P}_{h,s,a}\big(V_{h+1} - V_{h+1}^{\mathsf{R}}\big)$ is estimated via $V_{h+1}(s_{h+1}) - V_{h+1}^{\mathsf{R}}(s_{h+1})$.

• the term $\big[P_h V_{h+1}^{\mathsf{R}}\big](s,a)$ is estimated via $\mu_h^{\mathsf{ref},\mathsf{R}}(s_h, a_h)$. Given that this is estimated using all previous samples, we expect the variability of this term to be well-controlled as the sample size increases especially after V^{R} is locked.

• the exploration bonus $b_h^{\mathsf{R}}(s,a)$ is updated using $B_h^{\mathsf{R}}(s_h, a_h)$ and $\delta_h^{\mathsf{R}}(s_h, a_h)$ (cf. lines 6-7) that accounts for the confidence bounds of both the reference and the advantage.

With the above updates properly implemented, Q_h^{R} provides the advantage-based update of the Q-function at time step h, based on the update rule (11).

3.3 Regret of the proposed algorithm

Encouragingly, the proposed algorithm manages to achieve near-optimal regret even in the sample-limited and memory-limited regime, as formalized by the following theorem.

Theorem 2. *Suppose that $c_{\mathrm{b}} > 0$ is chosen to be a sufficiently large constant. Then there exists some absolute constant $C_0 > 0$ such that, for any $\delta \in (0,1)$, the regret (cf. (7)) of Algorithm 1 satisfies*

$$\mathsf{Regret}(T) \le C_0 \left(\sqrt{H^2 SAT \log \frac{SAT}{\delta}} + H^6 SA \log^3 \frac{SAT}{\delta} \right) \tag{19}$$

with probability at least $1 - \delta$.

Theorem 2 delivers a non-asymptotic characterization of the performance of our algorithm UCB-Q-Double-Advantage. Several appealing features of the algorithm are noteworthy.

• *Regret optimality.* Our regret bound (19) simplifies to $\mathsf{Regret}(T) \le \widetilde{O}\big(\sqrt{H^2 SAT}\big)$ as long as the sample size T exceeds $T \ge SA\,\mathrm{poly}(H)$. This sublinear regret bound is essentially optimal, as it coincides with the existing lower bound (1) modulo some logarithmic factor.

• *Sample complexity and a broadened sample size range.* As an interpretation of our theory, our algorithm attains ε average regret (i.e., $\frac{1}{K}\mathsf{Regret}(T) \le \varepsilon$) with a sample complexity $\widetilde{O}\Big(\frac{SAH^4}{\varepsilon^2}\Big)$. Crucially, the applicable sample size range $T \ge SA\,\mathrm{poly}(H)$ significantly broadens the range covered by the state-of-the-art memory-efficient model-free algorithm (Zhang et al., 2020b) (whose optimality is guaranteed only in the range $T \ge S^6 A^4\,\mathrm{poly}(H)$).

• *Memory efficiency.* Our algorithm, which is model-free in nature, achieves a low space complexity on the order of $O(SAH)$. This is basically un-improvable for the tabular case, since even storing the optimal Q-values alone takes $O(SAH)$ space. In comparison, while (Menard et al., 2021) also accommodates the sample size range $T \ge SA\,\mathrm{poly}(H)$, the algorithm proposed therein incurs a space complexity of $O(S^2AH)$ that is S times higher than ours.

• *Computational complexity.* An additional intriguing feature of our algorithm is its low computational complexity. The runtime of UCB-Q-Double-Advantage is no larger than $O(T)$, which is proportional to the time taken to read the samples. This matches the computational cost of the model-free algorithm UCB-Q proposed in (Jin et al., 2018), and is considerably lower than than of the UCB-M-Q algorithm in (Menard et al., 2021) (which has a computational cost of at least $O(ST)$).

4 Discussion

In this paper, we have proposed a novel model-free algorithm, tailored to online episodic RL, that attains near-optimal regret $\widetilde{O}(\sqrt{H^2 SAT})$ and near-minimal memory complexity $O(SAH)$ at once. Remarkably, the near-optimality of the algorithm comes into effect as soon as the sample size rises above $O(SA\,\mathrm{poly}(H))$, which significantly improves upon prior sample size requirements for any regret-optimal model-free algorithm (based on the definition of the model-free algorithm in (Jin et al., 2018)). We hope that the paradigms developed herein might inspire further works that allow one to overcome sample size barriers, in settings that include model-based RL (Azar et al., 2017), RL for discounted infinite-horizon MDPs (Zhang et al., 2020a), and the case with low-complexity function approximation (Jin et al., 2020), to name just a few. Additionally, our sample size range is not yet optimal in terms of its dependency on the horizon H, which we leave for furture investigation.

References

Agarwal, A., Kakade, S., and Yang, L. F. (2020). Model-based reinforcement learning with a generative model is minimax optimal. *Conference on Learning Theory*, pages 67–83.

Agrawal, S. and Jia, R. (2017). Posterior sampling for reinforcement learning: worst-case regret bounds. *arXiv preprint arXiv:1705.07041*.

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

Auer, P. and Ortner, R. (2010). UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65.

Azar, M. G., Kappen, H. J., Ghavamzadeh, M., and Munos, R. (2011). Speedy Q-learning. In *Advances in neural information processing systems*, pages 2411–2419.

Azar, M. G., Munos, R., and Kappen, H. J. (2013). Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model. *Machine learning*, 91(3):325–349.

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

Bai, Y., Xie, T., Jiang, N., and Wang, Y.-X. (2019). Provably efficient Q-learning with low switching cost. *arXiv preprint arXiv:1905.12849*.

Bartlett, P. and Tewari, A. (2009). Regal: a regularization based algorithm for reinforcement learning in weakly communicating MDPs. In *Uncertainty in Artificial Intelligence: Proceedings of the 25th Conference*, pages 35–42. AUAI Press.

Beck, C. L. and Srikant, R. (2012). Error bounds for constant step-size Q-learning. *Systems & control letters*, 61(12):1203–1208.

Bertsekas, D. P. (2017). *Dynamic programming and optimal control (4th edition)*. Athena Scientific.

Chen, Z., Maguluri, S. T., Shakkottai, S., and Shanmugam, K. (2020). Finite-sample analysis of stochastic approximation using smooth convex envelopes. *arXiv preprint arXiv:2002.00874*.

Chen, Z., Maguluri, S. T., Shakkottai, S., and Shanmugam, K. (2021). A Lyapunov theory for finite-sample guarantees of asynchronous Q-learning and TD-learning variants. *arXiv preprint arXiv:2102.01567*.

Dann, C., Lattimore, T., and Brunskill, E. (2017). Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning. *arXiv preprint arXiv:1703.07710*.

Domingues, O. D., Ménard, P., Kaufmann, E., and Valko, M. (2021). Episodic reinforcement learning in finite MDPs: Minimax lower bounds revisited. In *Algorithmic Learning Theory*, pages 578–598. PMLR.

Dong, K., Wang, Y., Chen, X., and Wang, L. (2019). Q-learning with UCB exploration is sample efficient for infinite-horizon MDP. *arXiv preprint arXiv:1901.09311*.

Du, S. S., Chen, J., Li, L., Xiao, L., and Zhou, D. (2017). Stochastic variance reduction methods for policy evaluation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1049–1058. JMLR. org.

Du, S. S., Luo, Y., Wang, R., and Zhang, H. (2019). Provably efficient Q-learning with function approximation via distribution shift error checking oracle. In *Advances in Neural Information Processing Systems*, pages 8058–8068.

Efroni, Y., Merlis, N., Ghavamzadeh, M., and Mannor, S. (2019). Tight regret bounds for model-based reinforcement learning with greedy policies. *arXiv preprint arXiv:1905.11527*.

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

Fan, J., Wang, Z., Xie, Y., and Yang, Z. (2019). A theoretical analysis of deep Q-learning. *arXiv e-prints*, pages arXiv–1901.

He, J., Zhou, D., and Gu, Q. (2020). Nearly minimax optimal reinforcement learning for discounted MDPs. *arXiv preprint arXiv:2010.00587*.

Jaakkola, T., Jordan, M. I., and Singh, S. P. (1994). Convergence of stochastic iterative dynamic programming algorithms. In *Advances in neural information processing systems*, pages 703–710.

Jafarnia-Jahromi, M., Wei, C.-Y., Jain, R., and Luo, H. (2020). A model-free learning algorithm for infinite-horizon average-reward MDPs with near-optimal regret. *arXiv preprint arXiv:2006.04354*.

Jaksch, T., Ortner, R., and Auer, P. (2010). Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(4).

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

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. (2020). Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR.

Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323.

Kakade, S. (2003). *On the sample complexity of reinforcement learning*. PhD thesis, University of London.

Khamaru, K., Pananjady, A., Ruan, F., Wainwright, M. J., and Jordan, M. I. (2020). Is temporal difference learning optimal? an instance-dependent analysis. *arXiv preprint arXiv:2003.07337*.

Lai, T. L. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22.

Lattimore, T. and Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press.

Li, G., Cai, C., Chen, Y., Gu, Y., Wei, Y., and Chi, Y. (2021). Is Q-learning minimax optimal? a tight sample complexity analysis. *arXiv preprint arXiv:2102.06548*.

Li, G., Wei, Y., Chi, Y., Gu, Y., and Chen, Y. (2020a). Breaking the sample size barrier in model-based reinforcement learning with a generative model. *Advances in Neural Information Processing Systems*, 33.

Li, G., Wei, Y., Chi, Y., Gu, Y., and Chen, Y. (2020b). Sample complexity of asynchronous Q-learning: Sharper analysis and variance reduction. *Advances in Neural Information Processing Systems (NeurIPS)*.

Liu, S. and Su, H. (2020). γ-regret for non-episodic reinforcement learning. *arXiv:2002.05138*.

Menard, P., Domingues, O. D., Shang, X., and Valko, M. (2021). UCB momentum Q-learning: Correcting the bias without forgetting. *arXiv preprint arXiv:2103.01312*.

Murphy, S. (2005). A generalization error for Q-learning. *Journal of Machine Learning Research*, 6:1073–1097.

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

Pacchiano, A., Ball, P., Parker-Holder, J., Choromanski, K., and Roberts, S. (2020). On optimism in model-based reinforcement learning. *arXiv preprint arXiv:2006.11911*.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Qu, G. and Wierman, A. (2020). Finite-time analysis of asynchronous stochastic approximation and Q-learning. *Conference on Learning Theory*, pages 3185–3205.

Sidford, A., Wang, M., Wu, X., Yang, L., and Ye, Y. (2018a). Near-optimal time and sample complexities for solving Markov decision processes with a generative model. In *Advances in Neural Information Processing Systems*, pages 5186–5196.

Sidford, A., Wang, M., Wu, X., and Ye, Y. (2018b). Variance reduced value iteration and faster algorithms for solving Markov decision processes. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 770–787. SIAM.

Strehl, A. L., Li, L., Wiewiora, E., Langford, J., and Littman, M. L. (2006). PAC model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 881–888.

Szepesvári, C. (1997). The asymptotic convergence-rate of Q-learning. In *NIPS*, volume 10, pages 1064–1070. Citeseer.

Tsitsiklis, J. N. (1994). Asynchronous stochastic approximation and Q-learning. *Machine learning*, 16(3):185–202.

Wai, H.-T., Hong, M., Yang, Z., Wang, Z., and Tang, K. (2019). Variance reduced policy evaluation with smooth function approximation. *Advances in Neural Information Processing Systems*, 32:5784–5795.

Wainwright, M. J. (2019a). Stochastic approximation with cone-contractive operators: Sharp ℓ_∞-bounds for Q-learning. *arXiv preprint arXiv:1905.06265*.

Wainwright, M. J. (2019b). Variance-reduced Q-learning is minimax optimal. *arXiv preprint arXiv:1906.04697*.

Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4):279–292.

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

Weng, B., Xiong, H., Zhao, L., Liang, Y., and Zhang, W. (2020). Momentum Q-learning with finite-sample convergence guarantee. *arXiv preprint arXiv:2007.15418*.

Xiong, H., Zhao, L., Liang, Y., and Zhang, W. (2020). Finite-time analysis for double Q-learning. *Advances in Neural Information Processing Systems*, 33.

Xu, T., Wang, Z., Zhou, Y., and Liang, Y. (2019). Reanalysis of variance reduced temporal difference learning. In *International Conference on Learning Representations*.

Yang, K., Yang, L., and Du, S. (2021). Q-learning with logarithmic regret. In *International Conference on Artificial Intelligence and Statistics*, pages 1576–1584. PMLR.

Yin, M., Bai, Y., and Wang, Y.-X. (2021). Near-optimal offline reinforcement learning via double variance reduction. *arXiv preprint arXiv:2102.01748*.

Zanette, A. and Brunskill, E. (2019). Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In *International Conference on Machine Learning*, pages 7304–7312. PMLR.

Zhang, Z., Ji, X., and Du, S. S. (2020a). Is reinforcement learning more difficult than bandits? a near-optimal algorithm escaping the curse of horizon. *arXiv preprint arXiv:2009.13503*.

Zhang, Z., Zhou, Y., and Ji, X. (2020b). Almost optimal model-free reinforcement learning via reference-advantage decomposition. *Advances in Neural Information Processing Systems*, 33.

Zhang, Z., Zhou, Y., and Ji, X. (2020c). Model-free reinforcement learning: from clipped pseudo-regret to sample complexity. *arXiv preprint arXiv:2006.03864*.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]