Model-Free Pessimistic Q-Learning for Offline RL: Sample Complexity and Efficiency Insights

Chris Junchi Li[◇]

Department of Electrical Engineering and Computer Sciences[◇]
University of California, Berkeley

September 19, 2024

Abstract

In this paper, we investigate pessimistic Q-learning algorithms for offline reinforcement learning (RL) in finite-horizon, non-stationary Markov decision processes (MDPs). Offline RL seeks to leverage historical data, often collected from suboptimal policies, to learn an optimal policy without further environment interaction. We focus on model-free approaches, introducing a pessimistic variant of Q-learning that applies lower confidence bounds to account for uncertainty in Q-value estimation. We provide theoretical guarantees on the sample complexity, demonstrating that our pessimistic Q-learning algorithm can achieve near-optimal sample efficiency. Additionally, we propose a variance-reduced extension, LCB-Q-Advantage, which further closes the gap to the minimax lower bound. Extensive comparisons with prior model-based approaches highlight the flexibility and improved performance of our model-free method in offline settings.

Keywords: Pessimistic Q-Learning; Offline Reinforcement Learning; Sample Complexity; Finite-Horizon MDP; Model-Free Algorithms; Variance Reduction; Batch RL

1 Introduction

Reinforcement Learning (RL) has achieved significant advancements in recent years, particularly in fields such as robotics, autonomous driving, and strategic games, where agents learn to make sequential decisions by interacting with their environment. However, these achievements are often associated with high computational costs, driven by the need for vast amounts of interaction data to reach optimal performance. In many real-world applications, especially where data collection is expensive or limited—such as healthcare, finance, and robotics—direct interaction with the environment is not feasible. This necessitates the study of *offline reinforcement learning*, which leverages pre-collected historical data to learn a near-optimal policy without further interaction.

Offline RL, in contrast to online RL, is subject to unique challenges. The historical data used in offline settings is often collected by a suboptimal behavior policy, which may have poor coverage of the state-action space. This introduces significant uncertainties when estimating the value of unseen state-action pairs, especially when the collected data does not adequately represent the entire environment. To address this issue, a growing body of work has explored the principle of *pessimism*, which applies conservative estimates of value functions in under-explored regions of the state-action space. This paper adopts the pessimism principle to design efficient algorithms for offline RL that balance sample efficiency with computational simplicity.

We focus on model-free approaches to offline RL, particularly Q-learning, due to their simplicity and flexibility. Model-free algorithms avoid the need to estimate the environment's dynamics, which can be computationally expensive in model-based methods. Instead, we develop a *pessimistic Q-learning* algorithm that incorporates lower confidence bounds (LCB) in the Q-value updates to

account for uncertainty in regions where data is sparse. Our algorithm achieves near-optimal sample complexity under standard assumptions and does so with low computational overhead. To further enhance sample efficiency, we introduce a variance-reduced variant, called LCB-Q-Advantage, which leverages the reference-advantage decomposition to reduce variance in the Q-learning updates.

History The success stories of Reinforcement Learning (RL), including matching or surpassing human performance in robotics control and strategy games [SSS+17, MKS+15], often come with nearly prohibitive cost, where an astronomical number of samples are required to train the learning algorithm to a satisfactory level. Scaling up and replicating the RL success in many real-world problems face considerable challenges, due to limited access to large-scale simulation data. In applications such as online advertising and clinical trials, real-time data collection could be expensive, time-consuming, or constrained in sample sizes as a result of experimental limitations.

On the other hand, it is worth noting that tons of samples might have already been accumulated and stored — albeit not necessarily with the desired quality — during previous data acquisition attempts. It is therefore natural to wonder whether such history data can be leveraged to improve performance in future deployments. In reality, the history data was often obtained by executing some (possibly unknown) behavior policy, which is typically not the desired policy. This gives rise to the problem of offline RL or batch RL [LGR12, LKTF20],[1] namely, how to make the best use of history data to learn an improved or even optimal policy, without further exploring the environment. In stark contrast to online RL that relies on active interaction with the environment, the performance of offline RL depends critically not only on the quantity, but also the quality of history data (e.g., coverage over the space-action space), given that the agent is no longer collecting new samples for the purpose of exploring the unknown environment.

Recently, the principle of pessimism (or conservatism) — namely, being conservative in Q-function estimation when there are not enough samples — has been put forward as an effective way to solve offline RL [BGB20, KZTL20]. This principle has been implemented in, for instance, a model-based offline value iteration algorithm, which modifies classical value iteration [AOM17] by subtracting a penalty term in the estimated Q-values and has been shown to achieve appealing sample efficiency [JYW21, RZM+21, XJW+21]. It is noteworthy that the model-based approach is built upon the construction of an empirical transition kernel, and therefore, requires specific representation of the environment (see, e.g. [AKY20, LWC+20]). It remains unknown whether the pessimism principle can be incorporated into model-free algorithms — another class of popular algorithms that performs learning without model estimation — in a provably effective fashion for offline RL.

Our main contributions are summarized as follows: (i) We propose a pessimistic Q-learning algorithm for finite-horizon, non-stationary MDPs in offline RL. This algorithm modifies the classical Q-learning update with lower confidence bounds and provides theoretical guarantees on its sample complexity. (ii) We introduce a variance-reduced variant, LCB-Q-Advantage, which significantly improves sample efficiency. Our algorithm achieves near-minimax optimality for sufficiently small accuracy levels. (iii) We provide a detailed theoretical analysis, comparing the performance of our model-free approaches to recent model-based algorithms, demonstrating the advantages of our approach in terms of flexibility and computational efficiency.

[1] Throughout this paper, we will be using the term offline RL (resp. dataset) or batch RL (resp. dataset) interchangeably.

1.1 Main contributions

In this paper, we consider finite-horizon non-stationary Markov decision processes (MDPs) with S states, A actions, and horizon length H. The focal point is to pin down the sample efficiency for pessimistic variants of model-free algorithms, under the mild single-policy concentrability assumption (cf. Assumption 1) of the batch dataset introduced in [RZM+21, XJW+21] (in short, this assumption captures how close the batch dataset is to an expert dataset, and will be formally introduced in Section 3.2). Given K episodes of history data each of length H (which amounts to a total number of $T = KH$ samples), our main contributions are summarized as follows.

- We first study a natural pessimistic variant of the Q-learning algorithm, which simply modifies the classical Q-learning update rule by subtracting a penalty term (via certain lower confidence bounds). We prove that pessimistic Q-learning finds an ε-optimal policy as soon as the sample size T exceeds the order of (up to log factor)

$$\frac{H^6SC^\star}{\varepsilon^2}$$

  where $C^\star$ denotes the single-policy concentrability coefficient of the batch dataset. In comparison to the minimax lower bound $\Omega\big(\frac{H^4SC^\star}{\varepsilon^2}\big)$ developed in [XJW+21], the sample complexity of pessimistic Q-learning is at most a factor of H^2 from optimal (modulo some log factor).

- To further improve the sample efficiency of pessimistic model-free algorithms, we introduce a variance-reduced variant of pessimistic Q-learning. This algorithm is guaranteed to find an ε-optimal policy as long as the sample size T is above the order of

$$\frac{H^4SC^\star}{\varepsilon^2} + \frac{H^5SC^\star}{\varepsilon}$$

  up to some log factor. In particular, this sample complexity is minimax-optimal (namely, as low as $\frac{H^4SC^\star}{\varepsilon^2}$ up to log factor) for small enough ε (namely, $\varepsilon \le (0, 1/H]$). The ε-range that enjoys near-optimality is much larger compared to $\varepsilon \le \big(0, 1/H^{2.5}\big]$ in [XJW+21] for model-based algorithms.

Both of the proposed algorithms achieve low computation cost (i.e., $O(T)$) and low memory complexities (i.e., $O(\min\{T, SAH\})$). Additionally, more complete comparisons with prior sample complexities of pessimistic model-based algorithms [XJW+21] are provided in Table 1. In comparison with model-based algorithms, model-free algorithms require drastically different technical tools to handle the complicated statistical dependency between the estimated Q-values at different time steps.

2 Related works

In this section, we discuss several lines of works which are related to ours, with an emphasis on value-based algorithms for tabular settings with finite state and action spaces.

Offline RL. One of the key challenges in offline RL lies in the insufficient coverage of the batch dataset, due to lack of interaction with the environment [LKTF20, LSAB20]. To address this challenge, most of the recent works can be divided into two lines: 1) regularizing the policy to

avoid visiting under-covered state and action pairs [FMP19, DRV+21]; 2) penalizing the estimated values of the under-covered state-action pairs [BGB20, KZTL20]. Our work follows the latter line (also known as the principle of pessimism), which has garnered significant attention recently. In fact, pessimism has been incorporated into recent development of various offline RL approaches, such as policy-based approaches [RDV+21, XCJ+21, ZWB21], model-based approaches [RZM+21, US21, JYW21, YTY+20, KRNJ20, XJW+21, YW21, UZS21, YLCF22b, YKR+21, YDWW22], and model-free approaches [KZTL20, YKC+21, YLCF22a].

Finite-sample guarantees for pessimistic approaches. While model-free approaches with pessimism [KZTL20, YKC+21] have achieved considerable empirical successes in offline RL, prior theoretical guarantees of pessimistic schemes have been confined almost exclusively to model-based approaches. Under the same single-policy concentrability assumption used in prior analyses of model-based approaches [RZM+21, XJW+21, YBW21a], the current paper provides the first finite-sample guarantees for model-free approaches with pessimism in the tabular case without explicit model construction. In addition, [YW21] directly employed the occupancy distributions of the behavior policy and the optimal policy in bounding the performance of a model-based approach, rather than the worst-case upper bound of their ratios as done under the single-policy concentrability assumption.

Non-asymptotic guarantees for variants of Q-learning. Q-learning, which is among the most famous model-free RL algorithms [Wat89, JJS94, WD92], has been adapted in a multitude of ways to deal with different RL settings. Theoretical analyses for Q-learning and its variants have been established in, for example, the online setting via regret analysis [JAZBJ18, BXJW19, ZZJ20a, LSC+21, DWCW19, ZJD20, ZZJ20b, JJWJL20, YYD21], and the simulator setting via probably approximately correct (PAC) bounds [CMSS20, Wai19, LCC+21]. The variant that is most closely related to ours is asynchronous Q-learning, which aims to find the optimal Q-function from Markovian trajectories following some behavior policy [EDM03, BS12, QW20, LWC+21, YBW21b, YBW21a, YBW21a]. Different from ours, these works typically require full coverage of the state-action space by the behavior policy, a much stronger assumption than the single-policy concentrability assumed in our offline RL setting.

Variance reduction in RL. Variance reduction, originally proposed to accelerate stochastic optimization (e.g., the SVRG algorithm proposed by [JZ13]), has been successfully leveraged to improve the sample efficiency of various RL algorithms, including but not limited to policy evaluation [DCL+17, WHY+19, XWZL19, KPR+20], planning [SWW+18, SWWY18], Q-learning and its variants [Wai19, ZZJ20a, LSC+21, LWC+21], and offline RL [XJW+21, YBW21a].

2.1 Notation and paper organization

Let us introduce a set of notation that will be used throughout. We denote by $\Delta(\mathcal{S})$ the probability simplex over a set $\mathcal{S}$, and introduce the notation $[N] \coloneqq \{1, \cdots, N\}$ for any integer $N > 0$. For any vector $x \in \mathbb{R}^{SA}$ (resp. $x \in \mathbb{R}^{S}$) that constitutes certain values for each of the state-action pairs (resp. state), we shall often use $x(s,a)$ (resp. $x(s)$) to denote the entry associated with the (s,a) pair (resp. state s). Similarly, we shall denote by $x \coloneqq \{x_h\}_{h\in[H]}$ the set composed of certain vectors for each of the time step $h \in [H]$. We let e_i represent the i-th standard basis vector, with the only non-zero element being in the i-th entry.

Let $\mathcal{X} \coloneqq (S, A, H, T)$. The notation $f(\mathcal{X}) \lesssim g(\mathcal{X})$ (resp. $f(\mathcal{X}) \gtrsim g(\mathcal{X})$) means that there exists a universal constant $C_0 > 0$ such that $|f(\mathcal{X})| \le C_0|g(\mathcal{X})|$ (resp. $|f(\mathcal{X})| \ge C_0|g(\mathcal{X})|$). In addition, we often overload scalar functions and expressions to take vector-valued arguments, with the interpretation that they are applied in an entrywise manner. For example, for a vector $x = [x_i]_{1\le i\le n}$, we have $x^2 = [x_i^2]_{1\le i\le n}$. For any two vectors $x = [x_i]_{1\le i\le n}$ and $y = [y_i]_{1\le i\le n}$, the notation $x \le y$ (resp. $x \ge y$) means $x_i \le y_i$ (resp. $x_i \ge y_i$) for all $1 \le i \le n$.

Paper organization. The rest of this paper is organized as follows. Section 3 introduces the backgrounds on finite-horizon MDPs and formulates the offline RL problem. Section 4 starts by introducing a natural pessimistic variant of Q-learning along with its sample complexity bound, and further enhances the sample efficiency via variance reduction in Section 4.3. Section 5 presents the proof outline and key lemmas. Finally, we conclude in Section 6 with a discussion and defer the proof details to the supplementary material.

3 Background and problem formulation

3.1 Tabular finite-horizon MDPs

Basics. This work focuses on an episodic finite-horizon MDP as represented by

$$\mathcal{M} = \big(\mathcal{S}, \mathcal{A}, H, \{P_h\}_{h=1}^H, \{r_h\}_{h=1}^H\big),$$

where H is the horizon length, $\mathcal{S}$ is a finite state space of cardinality S, $\mathcal{A}$ is a finite action space of cardinality A, 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 ($1 \le h \le H$). Throughout this paper, we shall adopt the following convenient notation

$$P_{h,s,a} \coloneqq P_h(\cdot \,|\, s, a) \in [0,1]^{1\times S} \tag{1}$$

which stands for the transition probability vector given the current state-action pair (s, a) at time step h. The parameters S, A and H can all be quite large, allowing one to capture the challenges arising in MDPs with large state/action space and long horizon.

A policy (or action selection rule) of an agent is represented by $\pi = \{\pi_h\}_{h=1}^H$, where $\pi_h : \mathcal{S} \to \Delta(\mathcal{A})$ specifies the associated selection probability over the action space at time step h (or more precisely, we let $\pi_h(a \,|\, s)$ represent the probability of selecting action a in state s at step h). When π is a deterministic policy, we abuse the notation and let $\pi_h(s)$ denote the action selected by policy π in state s at step h. In each episode, the agent generates an initial state $s_1 \in \mathcal{S}$ drawn from an initial state distribution $\rho \in \Delta(\mathcal{S})$, and rolls out a trajectory over the MDP by executing a policy π as follows:

$$\{s_h, a_h, r_h\}_{h=1}^H = \{s_1,\, a_1,\, r_1,\, \ldots, s_H,\, a_H,\, r_H\} \tag{2}$$

where at time step h, $a_h \sim \pi_h(\cdot \,|\, s_h)$ indicates the action selected in state s_h, $r_h = r_h(s_h, a_h)$ denotes the deterministic immediate reward, and s_{h+1} denotes the next state drawn from the transition probability vector $P_{h,s_h,a_h} \coloneqq P_h(\cdot \,|\, s_h, a_h)$. In addition, let $d_h^\pi(s)$ and $d_h^\pi(s,a)$ denote respectively the occupancy distribution induced by π at time step $h \in [H]$, namely,

$$d_h^\pi(s) \coloneqq \mathbb{P}(s_h = s \,|\, s_1 \sim \rho, \pi), \qquad d_h^\pi(s,a) \coloneqq \mathbb{P}(s_h = s \,|\, s_1 \sim \rho, \pi)\, \pi_h(a \,|\, s) \tag{3}$$

here and throughout, we denote $[H] \coloneqq \{1, \cdots, H\}$. Given that the initial state s_1 is drawn from ρ, the above definition gives

$$d_1^\pi(s) = \rho(s) \qquad \text{for any policy } \pi \tag{4}$$

Value function, Q-function, and optimal policy. The value function $V_h^\pi(s)$ of policy π in state s at step h is defined as the expected cumulative rewards when this policy is executed starting from state s at step h, i.e.,

$$V_h^\pi(s) \coloneqq \mathbb{E}\left[\sum_{t=h}^{H} r_t\big(s_t, a_t\big) \,\Big|\, s_h = s\right] \tag{5}$$

where the expectation is taken over the randomness of the trajectory (2) induced by the policy π as well as the MDP transitions. Similarly, the Q-function $Q_h^\pi(\cdot,\cdot)$ of a policy π at step h is defined as

$$Q_h^\pi(s,a) \coloneqq r_h(s,a) + \mathbb{E}\left[\sum_{t=h+1}^{H} r_t(s_t, a_t) \,\Big|\, s_h = s, a_h = a\right] \tag{6}$$

where the expectation is again over the randomness induced by π and the MDP except that the state-action pair at step h is now conditioned to be (s,a). By convention, we shall also set

$$V_{H+1}^\pi(s) = Q_{H+1}^\pi(s,a) = 0 \quad \text{for any } \pi \text{ and } (s,a) \in \mathcal{S} \times \mathcal{A} \tag{7}$$

A policy $\pi^\star = \{\pi_h^\star\}_{h=1}^H$ is said to be an optimal policy if it maximizes the value function (resp. Q-function) *simultaneously* for all states (resp. state-action pairs) among all policies, whose existence is always guaranteed [Put14]. The resulting optimal value function $V^\star = \{V_h^\star\}_{h=1}^H$ and optimal Q-functions $Q^\star = \{Q_h^\star\}_{h=1}^H$ are denoted respectively by

$$V_h^\star(s) \coloneqq V_h^{\pi^\star}(s) = \max_\pi V_h^\pi(s), \qquad Q_h^\star(s,a) \coloneqq Q_h^{\pi^\star}(s,a) = \max_\pi Q_h^\pi(s,a)$$

for any $(s,a,h) \in \mathcal{S} \times \mathcal{A} \times [H]$. Throughout this paper, we assume that $\pi^\star$ is a *deterministic optimal policy*, which always exists [Put14].

Additionally, when the initial state is drawn from a given distribution ρ, the expected value of a given policy π and that of the optimal policy at the initial step are defined respectively by

$$V_1^\pi(\rho) \coloneqq \mathop{\mathbb{E}}_{s_1 \sim \rho}\big[V_1^\pi(s_1)\big] \qquad \text{and} \qquad V_1^\star(\rho) \coloneqq \mathop{\mathbb{E}}_{s_1 \sim \rho}\big[V_1^\star(s_1)\big] \tag{8}$$

Bellman equations. The Bellman equations play a fundamental role in dynamic programming [Ber17]. Specifically, the value function and the Q-function of any policy π satisfy the following Bellman consistency equation:

$$Q_h^\pi(s,a) = r_h(s,a) + \mathop{\mathbb{E}}_{s' \sim P_{h,s,a}}\big[V_{h+1}^\pi(s')\big] \tag{9}$$

for all $(s,a,h) \in \mathcal{S} \times \mathcal{A} \times [H]$. Moreover, the optimal value function and the optimal Q-function satisfy the Bellman optimality equation:

$$Q_h^\star(s,a) = r_h(s,a) + \mathop{\mathbb{E}}_{s' \sim P_{h,s,a}}\big[V_{h+1}^\star(s')\big] \tag{10}$$

for all $(s,a,h) \in \mathcal{S} \times \mathcal{A} \times [H]$.

3.2 Offline RL under single-policy concentrability

Offline RL assumes the availability of a history dataset $\mathcal{D}_\mu$ containing K episodes each of length H. These episodes are independently generated based on a certain policy $\mu = \{\mu_h\}_{h=1}^H$ — called the *behavior policy*, resulting in a dataset

$$\mathcal{D}_\mu := \left\{ \left(s_1^k,\, a_1^k,\, r_1^k,\, \ldots,\, s_H^k,\, a_H^k,\, r_H^k\right) \right\}_{k=0}^{K-1}$$

Here, the initial states $\{s_1^k\}_{k=1}^K$ are independently drawn from $\rho \in \Delta(\mathcal{S})$ such that $s_1^k \overset{\text{i.i.d.}}{\sim} \rho$, while the remaining states and actions are generated by the MDP induced by the behavior policy μ. The total number of samples is thus given by

$$T = KH.$$

With the notation (8) in place, the goal of offline RL amounts to finding an ε-optimal policy $\widehat{\pi} = \{\widehat{\pi}_h\}_{h=1}^H$ satisfying

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

with as few samples as possible, and ideally, in a computationally fast and memory-efficient manner.

Obviously, efficient offline RL cannot be accomplished without imposing proper assumptions on the behavior policy, which also provide means to gauge the difficulty of the offline RL task through the quality of the history dataset. Following the recent works [RZM+21, XJW+21], we assume that the behavior policy μ satisfies the following property called *single-policy concentrability*.

Assumption 1 (single-policy concentrability)**.** *The single-policy concentrability coefficient $C^\star \in [1, \infty)$ of a behavior policy μ is defined to be the smallest quantity that satisfies*

$$\max_{(h,s,a)\in[H]\times\mathcal{S}\times\mathcal{A}} \frac{d_h^{\pi^\star}(s,a)}{d_h^{\mu}(s,a)} \le C^\star \tag{11}$$

where we adopt the convention $0/0 = 0$.

Intuitively, the single-policy concentrability coefficient measures the discrepancy between the optimal policy $\pi^\star$ and the behavior policy μ in terms of the resulting density ratio of the respective occupancy distributions. It is noteworthy that a finite $C^\star$ does not necessarily require μ to cover the entire state-action space; instead, it can be attainable when its coverage subsumes that of the optimal policy $\pi^\star$. This is in stark contrast to, and in fact much weaker than, other assumptions that require either full coverage of the behavior policy (i.e., $\min_{(h,s,a)\in[H]\times\mathcal{S}\times\mathcal{A}} d_h^\mu(s,a) > 0$ [LWC+21, YBW21b, YBW21a]), or uniform concentrability over all possible policies [CJ19]. Additionally, the single-policy concentrability coefficient is minimized (i.e., $C^\star = 1$) when the behavior policy μ coincides with the optimal policy $\pi^\star$, a scenario closely related to imitation learning or behavior cloning [RYJR20].

4 Pessimistic Q-learning: algorithms and theory

In the current paper, we present two model-free algorithms — namely, LCB-Q and LCB-Q-Advantage — for offline RL, along with their respective theoretical guarantees. The first algorithm can be viewed as a pessimistic variant of the classical Q-learning algorithm, while the second one further leverages the idea of variance reduction to boost the sample efficiency. In this section, we begin by introducing LCB-Q.

4.1 LCB-Q: a natural pessimistic variant of Q-learning

Before proceeding, we find it convenient to first review the classical Q-learning algorithm [Wat89, WD92], which can be regarded as a stochastic approximation scheme to solve the Bellman optimality equation (10). Upon receiving a sample transition (s_h, a_h, r_h, s_{h+1}) at time step h, Q-learning updates the corresponding entry in the Q-estimate as follows

$$Q_h(s_h, a_h) \;\leftarrow\; (1-\eta)Q_h(s_h, a_h) + \eta\Big\{r_h(s_h, a_h) + V_{h+1}(s_{h+1})\Big\} \tag{12}$$

where Q_h (resp. V_h) indicates the running estimate of $Q_h^\star$ (resp. $V_h^\star$), and $0 < \eta < 1$ is the learning rate. In comparison to model-based algorithms that require estimating the probability transition kernel based on all the samples, Q-learning, as a popular kind of model-free algorithms, is simpler and enjoys more flexibility without explicitly constructing the model of the environment. The wide applicability of Q-learning motivates one to adapt it to accommodate offline RL.

Inspired by recent advances in incorporating the pessimism principle for offline RL [RZM+21, JYW21], we study a pessimistic variant of Q-learning called LCB-Q, which modifies the Q-learning update rule as follows

$$Q_h(s_h, a_h) \leftarrow (1-\eta_n)Q_h(s_h, a_h) + \eta_n\Big\{r_h(s_h, a_h) + V_{h+1}(s_{h+1}) - b_n\Big\} \tag{13}$$

where η_n is the learning rate depending on the number of times n that the state-action pair (s_h, a_h) has been visited at step h, and the penalty term $b_n > 0$ (cf. line 8 of Algorithm 1) reflects the uncertainty of the corresponding Q-estimate and implements pessimism in the face of uncertainty. The entire algorithm, which is a *single-pass* algorithm that only requires reading the offline dataset once, is summarized in Algorithm 1.

4.2 Theoretical guarantees for LCB-Q

The proposed LCB-Q algorithm manages to achieve an appealing sample complexity as formalized by the following theorem.

Theorem 1. *Consider any $\delta \in (0,1)$. Suppose that the behavior policy μ satisfies Assumption 1 with single-policy concentrability coefficient $C^\star \geq 1$. Let $c_{\mathrm{b}} > 0$ be some sufficiently large constant, and take $\iota := \log\big(\frac{SAT}{\delta}\big)$. Assume that $T > SC^\star\iota$, then the policy $\widehat{\pi}$ returned by Algorithm 1 satisfies*

$$V_1^\star(\rho) - V_1^{\widehat{\pi}}(\rho) \leq c_{\mathrm{a}}\sqrt{\frac{H^6 SC^\star\iota^3}{T}} \tag{14}$$

with probability at least $1-\delta$, where $c_{\mathrm{a}} > 0$ is some universal constant.

As asserted by Theorem 1, the LCB-Q algorithm is guaranteed to find an ε-optimal policy with high probability, as long as the total sample size $T = KH$ exceeds

$$\widetilde{O}\left(\frac{H^6 SC^\star}{\varepsilon^2}\right) \tag{15}$$

where $\widetilde{O}(\cdot)$ hides logarithmic dependencies. When the behavior policy is close to the optimal policy, the single-policy concentrability coefficient $C^\star$ is closer to 1; if this is the case, then our bound indicates that the sample complexity does not depend on the size A of the action space, which can be a huge saving when the action space is enormous.

Algorithm 1: LCB-Q for offline RL

Parameters: some constant $c_{\mathrm{b}} > 0$, target success probability $1-\delta \in (0,1)$, and $\iota = \log\left(\frac{SAT}{\delta}\right)$.

Initialize $Q_h(s,a) \leftarrow 0$, $N_h(s,a) \leftarrow 0$, and $V_h(s) \leftarrow 0$ for all $(s,h) \in \mathcal{S} \times [H+1]$; $\widehat{\pi}$ s.t. $\widehat{\pi}_h(s) = 1$ for all $(h,s) \in [H] \times \mathcal{S}$.

for *Episode* $k = 1$ **to** K **do**

  Sample a new trajectory $\{s_h, a_h, r_h\}_{h=1}^{H}$ from $\mathcal{D}_\mu$. `// sampling from batch dataset`

  `// update the policy`

  for *Step* $h = 1$ **to** H **do**

    $N_h(s_h,a_h) \leftarrow N_h(s_h,a_h) + 1$. `// update the counter`

    $n \leftarrow N_h(s_h,a_h)$; $\eta_n \leftarrow \frac{H+1}{H+n}$. `// update the learning rate`

    $b_n \leftarrow c_{\mathrm{b}}\sqrt{\frac{H^3\iota^2}{n}}$. `// update the bonus term`

    `// run the Q-learning update with LCB`

    $Q_h(s_h,a_h) \leftarrow Q_h(s_h,a_h) + \eta_n\Big\{r_h(s_h,a_h) + V_{h+1}(s_{h+1}) - Q_h(s_h,a_h) - b_n\Big\}$

    `// update the value estimates`

    $V_h(s_h) \leftarrow \max\Big\{V_h(s_h),\, \max_a Q_h(s_h,a)\Big\}$

    If $V_h(s_h) = \max_a Q_h(s_h,a)$: update $\widehat{\pi}_h(s) \leftarrow \arg\max_a Q_h(s,a)$.

Output: the policy $\widehat{\pi}$.

Comparison with model-based pessimistic approaches. A model-based approach — called Value Iteration with Lower Confidence Bounds (VI-LCB) — has been recently proposed for offline RL [RZM+21, XJW+21]. In the finite-horizon case, VI-LCB incorporates an additional LCB penalty into the classical value iteration algorithm, and updates *all* the entries in the Q-estimate simultaneously as follows

$$Q_h(s,a) \;\leftarrow\; r_h(s,a) + \widehat{P}_{h,s,a}V_{h+1} - b_h(s,a) \tag{16}$$

with the aim of tuning down the confidence on those state-action pairs that have only been visited infrequently. Here, $\widehat{P}_{h,s,a}$ represents the empirical estimation of the transition kernel $P_{h,s,a}$, and $b_h(s,a) > 0$ is chosen to capture the uncertainty level of $(\widehat{P}_{h,s,a} - P_{h,s,a})V_{h+1}$. Working backward, the algorithm estimates the Q-value Q_h recursively over the time steps $h = H, H-1, \cdots, 1$. In comparison with VI-LCB, our sample complexity bound for LCB-Q matches the bound developed for VI-LCB by [XJW+21], while enjoying enhanced flexibility without the need of specifying the transition kernel of the environment (as model estimation might potentially incur a higher memory burden).

4.3 LCB-Q-Advantage for near-optimal offline RL

The careful reader might notice that the sample complexity (15) derived for LCB-Q remains a factor of H^2 away from the minimax lower bound (see Table 1). To further close the gap and improve the sample complexity, we propose a new variant called LCB-Q-Advantage, which leverages the idea of variance reduction to accelerate convergence [JZ13, SWWY18, Wai19, ZZJ20a, XJW+21, LWC+21, LSC+21].

Algorithm 2: Offline LCB-Q-Advantage RL

Parameters: number of epochs M, universal constant $c_{\mathrm{b}} > 0$, probability of failure $\delta \in (0,1)$, and $\iota = \log\left(\frac{SAT}{\delta}\right)$;

Initialize:

$Q_h(s,a), Q_h^{\mathsf{LCB}}(s,a), \overline{Q}_h(s,a), \overline{\mu}_h(s,a), \overline{\mu}_h^{\mathsf{next}}(s,a), N_h(s,a) \leftarrow 0$ for all $(s,a,h) \in \mathcal{S} \times \mathcal{A} \times [H]$;

$V_h(s), \overline{V}_h(s), \overline{V}_h^{\mathsf{next}}(s) \leftarrow 0$ for all $(s,h) \in \mathcal{S} \times [H+1]$;

$\mu_h^{\mathsf{ref}}(s,a), \sigma_h^{\mathsf{ref}}(s,a), \mu_h^{\mathsf{adv}}(s,a), \sigma_h^{\mathsf{adv}}(s,a), \overline{\delta}_h(s,a), \overline{B}_h(s,a) \leftarrow 0$ for all $(s,a,h) \in \mathcal{S} \times \mathcal{A} \times [H]$.

for *Epoch* $m = 1$ **to** M **do**

  $L_m = 2^m$; // specify the number of episodes in the current epoch

  $\widehat{N}_h(s,a) = 0$ for all $(h,s,a) \in [H] \times \mathcal{S} \times \mathcal{A}$. // reset the epoch-wise counter

  /* Inner-loop: update value-estimates $V_h(s,a)$ and Q-estimates $Q_h(s,a)$ */

  for *In-epoch Episode* $t = 1$ **to** L_m **do**

    Sample a new trajectory $\{s_h, a_h, r_h\}_{h=1}^H$. // sampling from batch dataset

    for *Step* $h = 1$ **to** H **do**

      $N_h(s_h,a_h) \leftarrow N_h(s_h,a_h) + 1$; $n \leftarrow N_h(s_h,a_h)$. // update the overall counter

      $\eta_n \leftarrow \frac{H+1}{H+n}$; // update the learning rate

      // run the Q-learning update rule with LCB

      $Q_h^{\mathsf{LCB}}(s_h,a_h) \leftarrow$ `update-lcb-q()`.

      // update the Q-estimate with LCB and reference-advantage

      $\overline{Q}_h(s_h,a_h) \leftarrow$ `update-lcb-q-ra()`.

      // update the Q-estimate Q_h and value estimate V_h

      $Q_h(s_h,a_h) \leftarrow \max\left\{Q_h^{\mathsf{LCB}}(s_h,a_h), \overline{Q}_h(s_h,a_h), Q_h(s_h,a_h)\right\}$.

      $V_h(s_h) \leftarrow \max_a Q_h(s_h,a)$

      // update the epoch-wise counter and $\overline{\mu}_h^{\mathsf{next}}$ for the next epoch

      $\widehat{N}_h(s_h,a_h) \leftarrow \widehat{N}_h(s_h,a_h) + 1$;

      $\overline{\mu}_h^{\mathsf{next}}(s_h,a_h) \leftarrow \left(1 - \frac{1}{\widehat{N}_h(s_h,a_h)}\right)\overline{\mu}_h^{\mathsf{next}}(s_h,a_h) + \frac{1}{\widehat{N}_h(s_h,a_h)}\overline{V}_{h+1}^{\mathsf{next}}(s_{h+1})$;

  /* Update the reference $(\overline{V}_h, \overline{V}_h^{\mathsf{next}})$ and $(\overline{\mu}_h, \overline{\mu}_h^{\mathsf{next}})$ */

  for $(s,a,h) \in \mathcal{S} \times \mathcal{A} \times [H+1]$ **do**

    $\overline{V}_h(s) \leftarrow \overline{V}_h^{\mathsf{next}}(s)$; $\overline{\mu}_h(s,a) \leftarrow \overline{\mu}_h^{\mathsf{next}}(s,a)$ // set $\overline{V}_h$ and $\overline{\mu}_h$ for the next epoch

    $\overline{V}_h^{\mathsf{next}}(s) \leftarrow V_h(s)$; $\overline{\mu}_h^{\mathsf{next}}(s,a) \leftarrow 0$. // restart $\overline{\mu}_h^{\mathsf{next}}$ and set $\overline{V}_h^{\mathsf{next}}$ for the next epoch

Output: the policy $\widehat{\pi}$ s.t. $\widehat{\pi}_h(s) = \arg\max_a Q_h(s,a)$ for any $(s,h) \in \mathcal{S} \times [H]$.

Inspired by the reference-advantage decomposition adopted in [ZZJ20a, LSC+21] for online Q-learning, LCB-Q-Advantage maintains a collection of reference values $\{\overline{V}_h\}_{h=1}^H$, which serve as running proxy for the optimal values $\{V_h^\star\}_{h=1}^H$ and allow for reduced variability in each iteration. To be more specific, the LCB-Q-Advantage algorithm (cf. Algorithm 2 as well as the subroutines in Algorithm 3 that closely resemble [LSC+21]) proceeds in an epoch-based style (the m-th epoch consists of $L_m = 2^m$ episodes of samples), where the reference values are updated at the end of each epoch to be used in the next epoch, and the Q-estimates are iteratively updated during the remaining time of each epoch. By maintaining two auxiliary sequences of *pessimistic* Q-estimates — that is, Q^{LCB} constructed by the pessimistic Q-learning update, and $\overline{Q}$ constructed by the

pessimistic Q-learning update based on the reference-advantage decomposition — the Q-estimate is updated by taking the maximum over the three candidates (cf. line 16 of Algorithm 2)

$$Q_h(s,a) \leftarrow \max\{Q_h^{\mathsf{LCB}}(s,a),\, \overline{Q}_h(s,a),\, Q_h(s,a)\} \tag{17}$$

when the state-action pair (s,a) is visited at the step h. We now take a moment to discuss the key ingredients of the proposed algorithm in further detail.

Updating the references $\overline{V}_h$ and $\overline{\mu}_h$. At the end of each epoch, the reference values $\{\overline{V}_h\}_{h=1}^H$, as well as the associated running average $\{\overline{\mu}_h\}_{h=1}^H$, are determined using what happens during the current epoch. More specifically, the following update rules for $\overline{V}_h$ and $\overline{\mu}_h$ are carried out at the end of the m-th epoch:

$$\overline{V}_h(s) \leftarrow \overline{V}_h^{\mathsf{next}}(s), \tag{18a}$$

$$\overline{\mu}_h(s,a) \leftarrow \frac{\sum_{t=1}^{L_m} \mathbb{1}(s_h^t = s, a_h^t = a)\overline{V}_{h+1}(s_{h+1}^t)}{\max\Big\{\big\{\sum_{t=1}^{L_m} \mathbb{1}(s_h^t = s, a_h^t = a)\big\}, 1\Big\}} \tag{18b}$$

for all $(h,s,a) \in [H]\times\mathcal{S}\times\mathcal{A}$. Here, $\overline{V}_h(s)$ is assigned by $\overline{V}_h^{\mathsf{next}}(s)$, which is maintained as the value estimate $V_h(s)$ at the end of the $(m-1)$-th epoch, and the update of $\overline{\mu}_h(s,a)$ is implemented in a recursive manner in the current m-th epoch. See also line 21 and line 19 of Algorithm 2.

Learning Q-estimate $\overline{Q}_h$ based on the reference-advantage decomposition. Armed with the references $\overline{V}_h$ and $\overline{\mu}_h$ updated at the end of the previous $(m-1)$-th epoch, LCB-Q-Advantage iteratively updates the Q-estimate $\overline{Q}_h$ in all episodes during the m-th epoch. At each time step h in any episode, whenever (s,a) is visited, LCB-Q-Advantage updates the reference Q-value as follows:

$$\overline{Q}_h(s,a) \leftarrow (1-\eta)\overline{Q}_h(s,a) + \eta\Big\{r_h(s,a) + \underbrace{\widehat{P}_{h,s,a}\big(V_{h+1} - \overline{V}_{h+1}\big)}_{\text{estimate of } P_{h,s,a}(V_{h+1}-\overline{V}_{h+1})} + \underbrace{\overline{\mu}_h}_{\text{estimate of } P_{h,s,a}\overline{V}_{h+1}} - \overline{b}_h(s,a)\Big\} \tag{19}$$

Intuitively, we decompose the target $P_{h,s,a}V_{h+1}$ into a reference part $P_{h,s,a}\overline{V}_{h+1}$ and an advantage part $P_{h,s,a}(V_{h+1} - \overline{V}_{h+1})$, and cope with the two parts separately. In the sequel, let us take a moment to discuss three essential ingredients of the update rule (19), which shed light on the design rationale of our algorithm.

- Akin to LCB-Q, the term $\widehat{P}_{h,s,a}\big(V_{h+1} - \overline{V}_{h+1}\big)$ serves as an unbiased stochastic estimate of $P_{h,s,a}\left(V_{h+1} - \overline{V}_{h+1}\right)$ if a sample transition (s,a,s_{h+1}) at time step h is observed. If V_{h+1} stays close to the reference $\overline{V}_{h+1}$ as the algorithm proceeds, the variance of this stochastic term can be lower than that of the stochastic term $\widehat{P}_{h,s,a}V_{h+1}$ in (13).

- The auxiliary estimate $\overline{\mu}_h$ introduced in (18b) serves as a running estimate of the reference part $P_{h,s,a}\overline{V}_{h+1}$. Based on the update rule (18b), we design $\overline{\mu}_h(s,a)$ to estimate the running mean of the reference part $\left[P_{h,s,a}\overline{V}_{h+1}\right]$ using a number of previous samples. As a result, we expect the variability of this term to be well-controlled, particularly as the number of samples in each epoch grows exponentially (recall that $L_m = 2^m$).

- In each episode, the term $\overline{b}_h(s,a)$ serves as the additional confidence bound on the error between the estimates of the reference/advantage and the ground truth. More specifically, $\mu_h^{\mathsf{ref}}(s,a)$ and $\sigma_h^{\mathsf{ref}}(s,a)$ are respectively the running mean and 2nd moment of the reference part $\left[P_{h,s,a}\overline{V}_{h+1}\right]$ (cf. lines 9-10 of Algorithm 3); $\mu_h^{\mathsf{adv}}(s,a)$ and $\sigma_h^{\mathsf{adv}}(s,a)$ represent respectively the running mean and 2nd moment of the advantage part $\left[P_{h,s,a}(V_{h+1}-\overline{V}_{h+1})\right]$ (cf. lines 11-12 of Algorithm 3); $\overline{B}_h(s,a)$ aggregates the empirical standard deviations of the reference and the advantage parts. The LCB penalty term $\overline{b}_h(s,a)$ is updated using $\overline{B}_h(s,a)$ and $\overline{\delta}_h(s_h,a_h)$ (cf. lines 5-6 of Algorithm 3), taking into account the confidence bounds for both the reference and the advantage.

In a nutshell, the auxiliary sequences of the reference values are designed to help reduce the variance of the stochastic Q-learning updates, which taken together with the principle of pessimism play a crucial role in the improvement of sample complexity for offline RL.

4.4 Theoretical guarantees for LCB-Q-Advantage

Encouragingly, the proposed LCB-Q-Advantage algorithm provably achieves near-optimal sample complexity for sufficiently small ε, as demonstrated by the following theorem.

Theorem 2. *Consider any $\delta \in (0,1)$, and recall that $\iota = \log\left(\frac{SAT}{\delta}\right)$ and $T = KH$. Suppose that $c_{\mathrm{b}} > 0$ is chosen to be a sufficiently large constant, and that the behavior policy μ satisfies Assumption 1. Then there exists some universal constant $c_{\mathrm{g}} > 0$ such that with probability at least $1-\delta$, the policy $\widehat{\pi}$ output by Algorithm 2 satisfies*

$$V_1^{\star}(\rho) - V_1^{\widehat{\pi}}(\rho) \le c_{\mathrm{g}}\left(\sqrt{\frac{H^4SC^{\star}\iota^5}{T}} + \frac{H^5SC^{\star}\iota^4}{T}\right) \tag{20}$$

As a consequence, Theorem 2 reveals that the LCB-Q-Advantage algorithm is guaranteed to find an ε-optimal policy (i.e., $V_1^{\star}(\rho) - V_1^{\widehat{\pi}}(\rho) \le \varepsilon$) as long as the total sample size T exceeds

$$\widetilde{O}\left(\frac{H^4SC^{\star}}{\varepsilon^2} + \frac{H^5SC^{\star}}{\varepsilon}\right) \tag{21}$$

For sufficiently small accuracy level ε (i.e., $\varepsilon \le 1/H$), this results in a sample complexity of

$$\widetilde{O}\left(\frac{H^4SC^{\star}}{\varepsilon^2}\right) \tag{22}$$

thereby matching the minimax lower bound developed in [XJW+21] up to logarithmic factor. Compared with the minimax lower bound $\Omega\left(\frac{H^4SA}{\varepsilon^2}\right)$ in the online RL setting [DMKV21], this suggests that offline RL can be fairly sample-efficient when the behavior policy closely mimics the optimal policy in terms of the resulting state-action occupancy distribution (a scenario where $C^{\star}$ is potentially much smaller than the size of the action space).

Comparison with offline model-based approaches. In the same offline finite-horizon setting, the state-of-art model-based approach called PEVI-Adv has been proposed by [XJW+21], which also leverage the idea of reference-advantage decomposition. In comparison with PEVI-Adv, LCB-Q-Advantage not only enjoys the flexibility of model-free approaches, but also achieves optimal

sample complexity for a broader range of target accuracy level ε. More precisely, the ε-range for which the algorithm achieves sample optimality can be compared as follows:

$$\underbrace{\varepsilon \le \left(0, H^{-1}\right]}_{\text{(Our LCB-Q-Advantage)}} \quad \text{vs.} \quad \underbrace{\varepsilon \le \left(0, H^{-2.5}\right]}_{\text{(PEVI-Adv)}} \tag{23}$$

offering an improvement by a factor of $H^{1.5}$.

5 Analysis

In this section, we outline the main steps needed to establish the main results in Theorem 1 and Theorem 2. Before proceeding, let us first recall the following rescaled learning rates

$$\eta_n = \frac{H+1}{H+n} \tag{24}$$

for the n-th visit of a given state-action pair at a given time step h, which are adopted in both LCB-Q and LCB-Q-Advantage. For notational convenience, we further introduce two sequences of related quantities defined for any integers $N \ge 0$ and $n \ge 1$:

$$\eta_0^N := \begin{cases} \prod_{i=1}^N (1-\eta_i) = 0, & \text{if } N > 0, \\ 1, & \text{if } N = 0, \end{cases} \quad \text{and} \quad \eta_n^N := \begin{cases} \eta_n \prod_{i=n+1}^N (1-\eta_i), & \text{if } N > n, \\ \eta_n, & \text{if } N = n \\ 0, & \text{if } N < n \end{cases} \tag{25}$$

The following identity can be easily verified:

$$\sum_{n=0}^N \eta_n^N = 1 \tag{26}$$

5.1 Analysis of LCB-Q

To begin with, we intend to derive a recursive formula concerning the update rule of Q_h^k — the estimate of the Q-function at step h at the beginning of the k-th episode. Note that we have omitted the dependency of all quantities on the episode index k in Algorithm 1. For notational convenience and clearness, we rewrite Algorithm 1 as Algorithm 4 by specifying the dependency on the episode index k and shall often use the following set of short-hand notation when it is clear from context.

- $N_h^k(s,a)$, or the shorthand N_h^k: the number of episodes that has visited (s,a) at step h before the beginning of the k-th episode.
- $k_h^n(s,a)$, or the shorthand k^n: the index of the episode in which the state-action pair (s,a) is visited at step h for the n-th times. We also adopt the convention that $k^0 = 0$.
- $P_h^k \in \{0,1\}^{1\times S}$: a row vector corresponding to the empirical transition at step h of the k-th episode, namely,

$$P_h^k(s) = \mathbb{1}\left(s = s_{h+1}^k\right) \qquad \text{for all } s \in \mathcal{S} \tag{27}$$

Algorithm 3: Auxiliary functions

Function `update-lcb-q()`:

$Q_h^{\mathsf{LCB}}(s_h,a_h) \leftarrow (1-\eta_n)Q_h^{\mathsf{LCB}}(s_h,a_h) + \eta_n\big(r(s_h,a_h) + V_{h+1}(s_{h+1}) - c_{\mathrm{b}}\sqrt{\frac{H^3\iota^2}{n}}\big)$

Function `update-lcb-q-ra()`:

`/* update the moment statistics of the interested terms */`

$[\mu_h^{\mathsf{ref}}, \sigma_h^{\mathsf{ref}}, \mu_h^{\mathsf{adv}}, \sigma_h^{\mathsf{adv}}](s_h,a_h) \leftarrow$ `update-moments()`;

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

$[\overline{\delta}_h, \overline{B}_h](s_h,a_h) \leftarrow$ `update-bonus()`;

$\overline{b}_h(s_h,a_h) \leftarrow \overline{B}_h(s_h,a_h) + (1-\eta_n)\frac{\overline{\delta}_h(s_h,a_h)}{\eta_n} + c_{\mathrm{b}}\frac{H^{7/4}\iota}{n^{3/4}} + c_{\mathrm{b}}\frac{H^2\iota}{n}$;

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

$\overline{Q}_h(s_h,a_h) \leftarrow$
$(1-\eta_n)\overline{Q}_h(s_h,a_h) + \eta_n\big(r_h(s_h,a_h) + V_{h+1}(s_{h+1}) - \overline{V}_{h+1}(s_{h+1}) + \overline{\mu}_h(s_h,a_h) - \overline{b}_h\big)$;

Function `update-moments()`:

$\mu_h^{\mathsf{ref}}(s_h,a_h) \leftarrow (1-\frac{1}{n})\mu_h^{\mathsf{ref}}(s_h,a_h) + \frac{1}{n}\overline{V}_{h+1}^{\mathsf{next}}(s_{h+1})$; `// mean of the reference`

$\sigma_h^{\mathsf{ref}}(s_h,a_h) \leftarrow (1-\frac{1}{n})\sigma_h^{\mathsf{ref}}(s_h,a_h) + \frac{1}{n}\big(\overline{V}_{h+1}^{\mathsf{next}}(s_{h+1})\big)^2$; `// 2nd moment of the reference`

$\mu_h^{\mathsf{adv}}(s_h,a_h) \leftarrow (1-\eta_n)\mu_h^{\mathsf{adv}}(s_h,a_h) + \eta_n\big(V_{h+1}(s_{h+1}) - \overline{V}_{h+1}(s_{h+1})\big)$; `// mean of the advantage`

$\sigma_h^{\mathsf{adv}}(s_h,a_h) \leftarrow (1-\eta_n)\sigma_h^{\mathsf{adv}}(s_h,a_h) + \eta_n\big(V_{h+1}(s_{h+1}) - \overline{V}_{h+1}(s_{h+1})\big)^2$. `// 2nd moment of the advantage`

Function `update-bonus()`:

$B_h^{\mathsf{next}}(s_h,a_h) \leftarrow$
$c_{\mathrm{b}}\sqrt{\frac{\iota}{n}}\Big(\sqrt{\sigma_h^{\mathsf{ref}}(s_h,a_h) - \big(\mu_h^{\mathsf{ref}}(s_h,a_h)\big)^2} + \sqrt{H}\sqrt{\sigma_h^{\mathsf{adv}}(s_h,a_h) - \big(\mu_h^{\mathsf{adv}}(s_h,a_h)\big)^2}\Big)$;

$\overline{\delta}_h(s_h,a_h) \leftarrow B_h^{\mathsf{next}}(s_h,a_h) - \overline{B}_h(s_h,a_h)$;

$\overline{B}_h(s_h,a_h) \leftarrow B_h^{\mathsf{next}}(s_h,a_h)$.

- $\pi^k = \{\pi_h^k\}_{h=1}^H$ with $\pi_h^k(s) := \arg\max_a Q_h^k(s,a), \forall (h,s) \in [H]\times\mathcal{S}$: the deterministic greedy policy at the beginning of the k-th episode.
- $\widehat{\pi}$: the final output $\widehat{\pi}$ of Algorithms 1 corresponds to π^{K+1} defined above; for notational simplicity, we shall treat $\widehat{\pi}$ as π^K in our analysis, which does not affect our result at all.

Consider any state-action pair (s,a). According to the update rule in line 11 of Algorithm 4, we can express (with the assistance of the above notation)

$$Q_h^k(s,a) = Q_h^{k^{N_h^k}+1}(s,a) = \big(1-\eta_{N_h^k}\big)Q_h^{k^{N_h^k}}(s,a) + \eta_{N_h^k}\Big\{r_h(s,a) + V_{h+1}^{k^{N_h^k}}\big(s_{h+1}^{k^{N_h^k}}\big) - b_{N_h^k}\Big\} \tag{28}$$

where the first identity holds since $k^{N_h^k}$ denotes the latest episode prior to k that visits (s,a) at step h, and the learning rate is defined in (24). Note that it always holds that $k > k^{N_h^k}$. Applying the above relation (28) recursively and using the notation (25) lead to

$$Q_h^k(s,a) = \eta_0^{N_h^k}Q_h^1(s,a) + \sum_{n=1}^{N_h^k}\eta_n^{N_h^k}\Big(r_h(s,a) + V_{h+1}^{k^n}\big(s_{h+1}^{k^n}\big) - b_n\Big) \tag{29}$$

Algorithm 4: LCB-Q for offline RL (a rewrite of Algorithm 1 to specify dependency on k)

1 **Parameters:** some constant $c_{\mathrm{b}} > 0$, target success probability $1-\delta \in (0,1)$, and $\iota = \log\left(\frac{SAT}{\delta}\right)$.

2 **Initialize** $Q_h^1(s,a) \leftarrow 0$; $N_h^1(s,a) \leftarrow 0$ for all $(s,a,h) \in \mathcal{S} \times \mathcal{A} \times [H]$; $V_h^1(s) \leftarrow 0$ for all $(s,h) \in \mathcal{S} \times [H+1]$; π^1 s.t. $\pi_h^1(s) = 1$ for all $(s,h) \in \mathcal{S} \times [H]$.

3 **for** *Episode* $k = 1$ **to** K **do**

4 Sample the k-th trajectory $\{s_h^k, a_h^k, r_h^k\}_{h=1}^H$ from $\mathcal{D}_\mu$. `// sampling from batch dataset`

5 **for** *Step* $h = 1$ **to** H **do**

6 **for** $(s,a) \in \mathcal{S} \times \mathcal{A}$ **do**

`// carry over the estimates and policy`

7 $N_h^{k+1}(s,a) \leftarrow N_h^k(s,a)$; $Q_h^{k+1}(s,a) \leftarrow Q_h^k(s,a)$; $V_h^{k+1}(s) \leftarrow V_h^k(s)$; $\pi_h^{k+1}(s) \leftarrow \pi_h^k(s)$.

8 $N_h^{k+1}(s_h^k, a_h^k) \leftarrow N_h^k(s_h^k, a_h^k) + 1$. `// update the counter`

9 $n \leftarrow N_h^{k+1}(s_h^k, a_h^k)$; $\eta_n \leftarrow \frac{H+1}{H+n}$. `// update the learning rate`

10 $b_n \leftarrow c_{\mathrm{b}}\sqrt{\frac{H^3\iota^2}{n}}$. `// update the bonus term`

`// update the Q-estimates with LCB`

11 $Q_h^{k+1}(s_h^k, a_h^k) \leftarrow Q_h^k(s_h^k, a_h^k) + \eta_n\Big\{r_h(s_h^k, a_h^k) + V_{h+1}^k(s_{h+1}^k) - Q_h^k(s_h^k, a_h^k) - b_n\Big\}$

`// update the value estimates`

12 $V_h^{k+1}(s_h^k) \leftarrow \max\Big\{V_h^k(s_h^k),\, \max_a Q_h^{k+1}(s_h^k, a)\Big\}$

`// update the policy`

13 If $V_h^{k+1}(s_h^k) = \max_a Q_h^{k+1}(s_h^k, a)$: update $\pi_h^{k+1}(s_h^k) = \arg\max_a Q_h^{k+1}(s_h^k, a)$

As another important fact, the value estimate V_h^k is monotonically non-decreasing in k, i.e.,

$$V_h^{k+1}(s) \geq V_h^k(s) \qquad \text{for all } (s,k,h) \in \mathcal{S} \times [K] \times [H] \tag{30}$$

which is an immediate consequence of the update rule in line 12 of Algorithm 4. Crucially, we observe that the iterate V_h^k forms a "pessimistic view" of $V_h^{\pi^k}$ — and in turn $V_h^\star$ — resulting from suitable design of the penalty term. This observation is formally stated in the following lemma, with the proof postponed to Section B.1.

Lemma 1. *Consider any $\delta \in (0,1)$, and suppose that $c_{\mathrm{b}} > 0$ is some sufficiently large constant. Then with probability at least $1-\delta$,*

$$\left|\sum_{n=1}^{N_h^k(s,a)} \eta_n^{N_h^k(s,a)} \Big(P_{h,s,a} - P_h^{k^n(s,a)}\Big) V_{h+1}^{k^n(s,a)}\right| \leq \sum_{n=1}^{N_h^k(s,a)} \eta_n^{N_h^k(s,a)} b_n \tag{31}$$

holds simultaneously for all $(k,h,s,a) \in [K] \times [H] \times \mathcal{S} \times \mathcal{A}$, and

$$V_h^k(s) \leq V_h^{\pi^k}(s) \leq V_h^\star(s) \tag{32}$$

holds simultaneously for all $(k,h,s) \in [K] \times [H] \times \mathcal{S}$.

In a nutshell, the result (32) in Lemma 1 reveals that V_h^k is a pointwise lower bound on $V_h^{\pi^k}$ and $V_h^\star$, thereby forming a pessimistic estimate of the optimal value function. In addition, the property (31) in Lemma 1 essentially tells us that the weighted sum of the penalty terms dominates the weighted sum of the uncertainty terms, which plays a crucial role in ensuring the aforementioned pessimism property. As we shall see momentarily, Lemma 1 forms the basis of the subsequent proof.

We are now ready to embark on the analysis for LCB-Q, which is divided into multiple steps as follows.

Step 1: decomposing estimation errors. With the aid of Lemma 1, we can develop an upper bound on the performance difference of interest in (20) as follows

$$\begin{aligned} V_1^\star(\rho) - V_1^{\widehat{\pi}}(\rho) &= \mathop{\mathbb{E}}_{s_1\sim\rho}\left[V_1^\star(s_1)\right] - \mathop{\mathbb{E}}_{s_1\sim\rho}\left[V_1^{\pi^K}(s_1)\right] \\ &\overset{\text{(i)}}{\le} \mathop{\mathbb{E}}_{s_1\sim\rho}\left[V_1^\star(s_1)\right] - \mathop{\mathbb{E}}_{s_1\sim\rho}\left[V_1^{K}(s_1)\right] \\ &\overset{\text{(ii)}}{\le} \frac{1}{K}\sum_{k=1}^{K}\left(\mathop{\mathbb{E}}_{s_1\sim\rho}\left[V_1^\star(s_1)\right] - \mathop{\mathbb{E}}_{s_1\sim\rho}\left[V_1^{k}(s_1)\right]\right) \\ &= \frac{1}{K}\sum_{k=1}^{K}\sum_{s\in\mathcal{S}} d_1^{\pi^\star}(s)\left(V_1^\star(s) - V_1^k(s)\right) \end{aligned} \tag{33}$$

where (i) results from Lemma 1 (i.e., $V_1^{\pi^K}(s) \ge V_1^K(s)$ for all $s\in\mathcal{S}$), (ii) follows from the monotonicity property in (30), and the last equality holds since $d_1^{\pi^\star}(s) = \rho(s)$ (cf. (4)).

We then attempt to bound the quantity on the right-hand side of (33). Given that $\pi^\star$ is assumed to be a deterministic policy, we have $d_h^{\pi^\star}(s) = d_h^{\pi^\star}(s,\pi^\star(s))$. Taking this together with the relations $V_h^k(s) \ge \max_a Q_h^k(s,a) \ge Q_h^k(s,\pi_h^\star(s))$ (see line 12 of Algorithm 4) and $V_h^\star(s) = Q_h^\star(s,\pi_h^\star(s))$, we obtain

$$\begin{aligned} \sum_{k=1}^{K}\sum_{s\in\mathcal{S}} d_h^{\pi^\star}(s)\left(V_h^\star(s) - V_h^k(s)\right) &= \sum_{k=1}^{K}\sum_{s\in\mathcal{S}} d_h^{\pi^\star}(s,\pi_h^\star(s))\left(V_h^\star(s) - V_h^k(s)\right) \\ &\le \sum_{k=1}^{K}\sum_{s\in\mathcal{S}} d_h^{\pi^\star}(s,\pi_h^\star(s))\Big(Q_h^\star\big(s,\pi_h^\star(s)\big) - Q_h^k\big(s,\pi_h^\star(s)\big)\Big) \\ &= \sum_{k=1}^{K}\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_h^{\pi^\star}(s,a)\left(Q_h^\star(s,a) - Q_h^k(s,a)\right) \end{aligned} \tag{34}$$

for any $h\in[H]$, where the last identity holds since $\pi^\star$ is deterministic and hence

$$d_h^{\pi^\star}(s,a) = 0 \qquad \text{for any } a \ne \pi_h^\star(s) \tag{35}$$

In view of (34), we need to properly control $Q_h^\star(s,a) - Q_h^k(s,a)$. By virtue of (26), we can rewrite $Q_h^\star(s,a)$ as follows

$$Q_h^\star(s,a) = \sum_{n=0}^{N_h^k}\eta_n^{N_h^k} Q_h^\star(s,a) = \eta_0^{N_h^k} Q_h^\star(s,a) + \sum_{n=1}^{N_h^k}\eta_n^{N_h^k} Q_h^\star(s,a)$$

$$= \eta_0^{N_h^k} Q_h^\star(s,a) + \sum_{n=1}^{N_h^k} \eta_n^{N_h^k} \left(r_h(s,a) + P_{h,s,a} V_{h+1}^\star\right) \tag{36}$$

where the second line follows from Bellman's optimality equation (10). Combining (29) and (36) leads to

$$\begin{aligned}
&Q_h^\star(s,a) - Q_h^k(s,a) \\
&= \eta_0^{N_h^k} \left(Q_h^\star(s,a) - Q_h^1(s,a)\right) + \sum_{n=1}^{N_h^k} \eta_n^{N_h^k} \left(P_{h,s,a} V_{h+1}^\star - V_{h+1}^{k^n}(s_{h+1}^{k^n}) + b_n\right) \\
&= \eta_0^{N_h^k} \left(Q_h^\star(s,a) - Q_h^1(s,a)\right) + \sum_{n=1}^{N_h^k} \eta_n^{N_h^k} b_n + \sum_{n=1}^{N_h^k} \eta_n^{N_h^k} P_{h,s,a}\left(V_{h+1}^\star - V_{h+1}^{k^n}\right) + \sum_{n=1}^{N_h^k} \eta_n^{N_h^k} \left(P_{h,s,a} - P_h^{k^n}\right) V_{h+1}^{k^n} \qquad (37) \\
&\le \eta_0^{N_h^k} H + 2\sum_{n=1}^{N_h^k} \eta_n^{N_h^k} b_n + \sum_{n=1}^{N_h^k} \eta_n^{N_h^k} P_{h,s,a}\left(V_{h+1}^\star - V_{h+1}^{k^n}\right) \qquad (38)
\end{aligned}$$

where we have made use of the definition in (27) by recognizing $P_h^{k^n} V_{h+1}^{k^n} = V_{h+1}^{k^n}(s_{h+1}^{k^n})$ in (37), and the last inequality follows from the fact $Q_h^\star(s,a) - Q_h^1(s,a) = Q_h^\star(s,a) - 0 \le H$ and the bound (31) in Lemma 1. Substituting the above bound into (34), we arrive at

$$\begin{aligned}
\sum_{k=1}^{K} \sum_{s\in\mathcal{S}} d_h^{\pi^\star}(s) \left(V_h^\star(s) - V_h^k(s)\right) \le \underbrace{\sum_{k=1}^{K} \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_h^{\pi^\star}(s,a) \eta_0^{N_h^k(s,a)} H + 2\sum_{k=1}^{K} \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_h^{\pi^\star}(s,a) \sum_{n=1}^{N_h^k(s,a)} \eta_n^{N_h^k(s,a)} b_n}_{=: I_h} \\
+ \sum_{k=1}^{K} \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_h^{\pi^\star}(s,a) P_{h,s,a} \sum_{n=1}^{N_h^k(s,a)} \eta_n^{N_h^k(s,a)} \left(V_{h+1}^\star - V_{h+1}^{k_h^n(s,a)}\right)
\end{aligned} \tag{39}$$

Step 2: establishing a crucial recursion. As it turns out, the last term on the right-hand side of (39) can be used to derive a recursive relation that connects step h with step $h+1$, as summarized in the next lemma.

Lemma 2. *With probability at least $1-\delta$, the following recursion holds:*

$$\begin{aligned}
&\sum_{k=1}^{K} \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_h^{\pi^\star}(s,a) P_{h,s,a} \sum_{n=1}^{N_h^k(s,a)} \eta_n^{N_h^k(s,a)} \left(V_{h+1}^\star - V_{h+1}^{k_h^n(s,a)}\right) \\
&\le \left(1 + \frac{1}{H}\right) \sum_{k=1}^{K} \sum_{s\in\mathcal{S}} d_{h+1}^{\pi^\star}(s) \left(V_{h+1}^\star(s) - V_{h+1}^k(s)\right) + 24\sqrt{H^2 C^\star K \log\frac{2H}{\delta}} + 12HC^\star \log\frac{2H}{\delta}
\end{aligned} \tag{40}$$

Lemma 2 taken together with (39) implies that

$$\sum_{k=1}^{K} \sum_{s\in\mathcal{S}} d_h^{\pi^\star}(s) \left(V_h^\star(s) - V_h^k(s)\right) \le \left(1 + \frac{1}{H}\right) \sum_{k=1}^{K} \sum_{s\in\mathcal{S}} d_{h+1}^{\pi^\star}(s) \left(V_{h+1}^\star(s) - V_{h+1}^k(s)\right)$$

$$+ I_h + 24\sqrt{H^2C^\star K \log\frac{2H}{\delta}} + 12HC^\star \log\frac{2H}{\delta} \tag{41}$$

Invoking (41) recursively over the time steps $h = H, H-1, \cdots, 1$ with the terminal condition $V_{H+1}^k = V_{H+1}^\star = 0$, we reach

$$\begin{aligned}\sum_{k=1}^{K}\sum_{s\in\mathcal{S}} d_1^{\pi^\star}(s)\left(V_1^\star(s) - V_1^k(s)\right) &\le \max_{h\in[H]}\sum_{k=1}^{K}\sum_{s\in\mathcal{S}} d_h^{\pi^\star}(s)\left(V_h^\star(s) - V_h^k(s)\right)\\ &\le \sum_{h=1}^{H}\left(1+\frac{1}{H}\right)^{h-1}\left(I_h + 24\sqrt{H^2C^\star K\log\frac{2H}{\delta}} + 12HC^\star\log\frac{2H}{\delta}\right)\end{aligned} \tag{42}$$

which captures the estimation error resulting from the use of pessimism principle.

Step 3: controlling the right-hand side of (42). The right-hand side of (42) can be bounded through the following lemma, which will be proved in Appendix B.3.

Lemma 3. *Consider any $\delta \in (0,1)$. With probability at least $1-\delta$, we have*

$$\sum_{h=1}^{H}\left(1+\frac{1}{H}\right)^{h-1}\left(I_h + 24\sqrt{H^2C^\star K\log\frac{2H}{\delta}} + 12HC^\star\log\frac{2H}{\delta}\right) \lesssim H^2SC^\star\iota + \sqrt{H^5SC^\star K\iota^3} \tag{43}$$

where we recall that $\iota \coloneqq \log\left(\frac{SAT}{\delta}\right)$.

Combining Lemma 3 with (42) and (33) yields

$$\begin{aligned}V_1^\star(\rho) - V_1^{\widehat{\pi}}(\rho) &\le \frac{1}{K}\sum_{k=1}^{K}\sum_{s\in\mathcal{S}} d_1^{\pi^\star}(s)\left(V_1^\star(s) - V_1^k(s)\right)\\ &\le \frac{1}{K}\max_{h\in[H]}\sum_{k=1}^{K}\sum_{s\in\mathcal{S}} d_h^{\pi^\star}(s)\left(V_h^\star(s) - V_h^k(s)\right)\\ &\le \frac{c_{\mathrm{a}}}{2}\sqrt{\frac{H^5SC^\star\iota^3}{K}} + \frac{c_{\mathrm{a}}}{2}\frac{H^2SC^\star\iota}{K} = \frac{c_{\mathrm{a}}}{2}\sqrt{\frac{H^6SC^\star\iota^3}{T}} + \frac{c_{\mathrm{a}}}{2}\frac{H^3SC^\star\iota}{T}\\ &\le c_{\mathrm{a}}\sqrt{\frac{H^6SC^\star\iota^3}{T}}\end{aligned} \tag{44}$$

for some sufficiently large constant $c_{\mathrm{a}} > 0$, where the last inequality is valid as long as $T > SC^\star\iota$. This concludes the proof of Theorem 1.

5.2 Analysis of LCB-Q-Advantage

We now turn to the analysis of LCB-Q-Advantage. Thus far, we have omitted the dependency of all quantities on the epoch number m and the in-epoch episode number t in Algorithms 2 and 3. While it allows for a more concise description of our algorithm, it might hamper the clarity of our proofs. In the following, we introduce the notation k to denote the current episode as follows:

$$k \coloneqq \sum_{i=1}^{m-1} L_i + t \tag{45}$$

which corresponds to the t-th in-epoch episode in the m-th epoch; here, $L_m = 2^m$ stands for the total number of in-epoch episodes in the m-th epoch. With this notation in place, we can rewrite Algorithm 2 as Algorithm 5 in order to make clear the dependency on the episode index k, epoch number m, and in-epoch episode index t.

Before embarking on our main proof, we make two crucial observations which play important roles in our subsequent analysis. First, similar to the property (30) for LCB-Q, the update rule (cf. lines 16-17 of Algorithm 5) ensures the monotonic non-decreasing property of $V_h(s)$ such that for all $k \in [K]$,

$$V_h^{k+1}(s) \geq V_h^k(s), \qquad \text{for all } (k,s,h) \in [K] \times \mathcal{S} \times [H] \tag{46}$$

Secondly, V_h^k forms a "pessimistic view" of $V_h^\star$, which is formalized in the lemma below; the proof is deferred to Appendix C.1.

Lemma 4. *Let $\delta \in (0,1)$. Suppose that $c_{\mathrm{b}} > 0$ is some sufficiently large constant. Then with probability at least $1-\delta$, the value estimates produced by Algorithm 2 satisfy*

$$V_h^k(s) \leq V_h^{\pi^k}(s) \leq V^\star(s) \tag{47}$$

for all $(k,h,s) \in [K] \times [H+1] \times \mathcal{S}$.

With these two observations in place, we can proceed to present the analysis for LCB-Q-Advantage. To begin with, the performance difference of interest can be controlled similar to (33) as follows:

$$\begin{aligned} V_1^\star(\rho) - V_1^{\widehat{\pi}}(\rho) &= \mathop{\mathbb{E}}_{s_1\sim\rho}\left[V_1^\star(s_1)\right] - \mathop{\mathbb{E}}_{s_1\sim\rho}\left[V_1^{\pi^K}(s_1)\right] \\ &\overset{\text{(i)}}{\leq} \mathop{\mathbb{E}}_{s_1\sim\rho}\left[V_1^\star(s_1)\right] - \mathop{\mathbb{E}}_{s_1\sim\rho}\left[V_1^{K}(s_1)\right] \\ &\overset{\text{(ii)}}{\leq} \frac{1}{K}\sum_{k=1}^{K}\left(\mathop{\mathbb{E}}_{s_1\sim\rho}\left[V_1^\star(s_1)\right] - \mathop{\mathbb{E}}_{s_1\sim\rho}\left[V_1^{k}(s_1)\right]\right) \\ &= \frac{1}{K}\sum_{k=1}^{K}\sum_{s\in\mathcal{S}} d_1^{\pi^\star}(s)\left(V_1^\star(s) - V_1^k(s)\right) \end{aligned} \tag{48}$$

where (i) follows from Lemma 4 (i.e., $V_1^{\pi^K}(s) \geq V_1^K(s)$ for all $s \in \mathcal{S}$), (ii) holds due to the monotonicity in (46) and the last equality holds since $d_1^{\pi^\star}(s) = \rho(s)$ (cf. (4)). It then boils down to controlling the right-hand side of (48). Towards this end, it turns out that one can control a more general counterpart, i.e.,

$$\sum_{k=1}^{K}\sum_{s\in\mathcal{S}} d_h^{\pi^\star}(s)\left(V_h^\star(s) - V_h^k(s)\right) \tag{49}$$

for any $h \in [H]$. This is accomplished via the following lemma, whose proof is postponed to Appendix C.2.

Lemma 5. *Let $\delta \in (0,1)$, and recall that $\iota := \log\left(\frac{SAT}{\delta}\right)$. Suppose that $c_{\mathrm{a}}, c_{\mathrm{b}} > 0$ are some sufficiently large constants. Then with probability at least $1-\delta$, one has*

$$\sum_{k=1}^{K}\sum_{s\in\mathcal{S}} d_h^{\pi^\star}(s)\left(V_h^\star(s) - V_h^k(s)\right) \leq J_h^1 + J_h^2 + J_h^3 \tag{50}$$

where

$$
\begin{aligned}
J_h^1 &:= \sum_{k=1}^{K} \sum_{s,a\in\mathcal{S}\times\mathcal{A}} d_h^{\pi^\star}(s,a) \left[\eta_0^{N_h^k(s,a)} H + \frac{4c_{\mathrm{b}} H^{7/4}\iota}{\left(N_h^k(s,a)\vee 1\right)^{3/4}} + \frac{4c_{\mathrm{b}} H^2\iota}{N_h^k(s,a)\vee 1} \right] \\
J_h^2 &:= 2\sum_{k=1}^{K} \sum_{s,a\in\mathcal{S}\times\mathcal{A}} d_h^{\pi^\star}(s,a) \overline{B}_h^k(s,a) \\
J_h^3 &:= \left(1+\frac{1}{H}\right) \sum_{k=1}^{K} \sum_{s\in\mathcal{S}} d_{h+1}^{\pi^\star}(s) \left(V_{h+1}^\star(s) - V_{h+1}^k(s)\right) + 48\sqrt{HC^\star K \log\frac{2H}{\delta}} + 28c_{\mathrm{a}} H^3 C^\star \sqrt{S}\iota^2
\end{aligned}
\tag{51}
$$

As a direct consequence of Lemma 5, one arrives at a recursive relationship between time steps h and $h+1$ as follows:

$$
\begin{aligned}
&\sum_{k=1}^{K} \sum_{s\in\mathcal{S}} d_h^{\pi^\star}(s) \left(V_h^\star(s) - V_h^k(s)\right) \\
&\le \left(1+\frac{1}{H}\right) \sum_{k=1}^{K} \sum_{s\in\mathcal{S}} d_{h+1}^{\pi^\star}(s) \left(V_{h+1}^\star(s) - V_{h+1}^k(s)\right) + 48\sqrt{HC^\star K \log\frac{2H}{\delta}} + 28c_{\mathrm{a}} H^3 C^\star \sqrt{S}\iota^2 + J_h^1 + J_h^2
\end{aligned}
\tag{52}
$$

Recursing over time steps $h = H, H-1, \cdots, 1$ with the terminal condition $V_{H+1}^k = V_{H+1}^\star = 0$, we can upper bound the performance difference at $h=1$ as follows

$$
\begin{aligned}
\sum_{k=1}^{K} \sum_{s\in\mathcal{S}} d_1^{\pi^\star}(s) \left(V_1^\star(s) - V_1^k(s)\right) &\le \max_{h\in[H]} \sum_{k=1}^{K} \sum_{s\in\mathcal{S}} d_h^{\pi^\star}(s) \left(V_h^\star(s) - V_h^k(s)\right) \\
&\le \sum_{h=1}^{H} \left(1+\frac{1}{H}\right)^{h-1} \left(48\sqrt{HC^\star K \log\frac{2H}{\delta}} + 28c_{\mathrm{a}} H^3 C^\star \sqrt{S}\iota^2 + J_h^1 + J_h^2\right)
\end{aligned}
\tag{53}
$$

To finish up, it suffices to upper bound each term in (53) separately. We summarize their respective upper bounds as follows; the proof is provided in Appendix C.3.

Lemma 6. *Fix $\delta \in (0,1)$, and recall that $\iota := \log\left(\frac{SAT}{\delta}\right)$. With probability at least $1-\delta$, we have*

$$
\sum_{h=1}^{H} \left(1+\frac{1}{H}\right)^{h-1} J_h^1 \lesssim H^{2.75} (SC^\star)^{\frac{3}{4}} K^{\frac{1}{4}} \iota^2 + H^3 SC^\star \iota^3 \tag{54a}
$$

$$
\sum_{h=1}^{H} \left(1+\frac{1}{H}\right)^{h-1} J_h^2 \lesssim \sqrt{H^4 SC^\star \iota^3 \max_{h\in[H]} \sum_{k=1}^{K} \sum_{s\in\mathcal{S}} d_h^{\pi^\star}(s) \left(V_h^\star(s) - V_h^k(s)\right)} + \sqrt{H^3 SC^\star K\iota^5} + H^4 SC^\star \iota^4 \tag{54b}
$$

$$
\sum_{h=1}^{H} \left(1+\frac{1}{H}\right)^{h-1} \left(48\sqrt{HC^\star K \log\frac{2H}{\delta}} + 28c_{\mathrm{a}} H^3 C^\star \sqrt{S}\iota^2\right) \lesssim \sqrt{H^3 C^\star K \log\frac{2H}{\delta}} + H^4 C^\star \sqrt{S}\iota^2 \tag{54c}
$$

Substituting the above upper bounds into (48) and (53) and recalling that $T = HK$, we arrive at

$$
\begin{aligned}
V_1^\star(\rho) - V_1^{\widehat{\pi}}(\rho) &\lesssim \frac{1}{K} \max_{h\in[H]} \sum_{k=1}^{K} \sum_{s\in\mathcal{S}} d_h^{\pi^\star}(s) \left(V_h^\star(s) - V_h^k(s)\right) \\
&\lesssim \frac{1}{K} \left(\sqrt{H^4 SC^\star \iota^3 \max_{h\in[H]} \sum_{k=1}^{K} \sum_{s\in\mathcal{S}} d_h^{\pi^\star}(s) \left(V_h^\star(s) - V_h^k(s)\right)} + \left(\sqrt{H^3 SC^\star K\iota^5} + H^4 SC^\star \iota^4 + H^{2.75} (SC^\star)^{\frac{3}{4}} K^{\frac{1}{4}} \iota^2 \right) \right) \\
&\overset{\text{(i)}}{\asymp} \frac{1}{K} \left(\sqrt{H^4 SC^\star \iota^3 \max_{h\in[H]} \sum_{k=1}^{K} \sum_{s\in\mathcal{S}} d_h^{\pi^\star}(s) \left(V_h^\star(s) - V_h^k(s)\right)} + \sqrt{H^3 SC^\star K\iota^5} + H^4 SC^\star \iota^4 \right) \\
&\overset{\text{(ii)}}{\lesssim} \frac{1}{K} \left(\sqrt{H^3 SC^\star K\iota^5} + H^4 SC^\star \iota^4 \right) \\
&\asymp \sqrt{\frac{H^4 SC^\star \iota^5}{T}} + \frac{H^5 SC^\star \iota^4}{T}
\end{aligned}
$$

where (i) has made use of the AM-GM inequality:

$$
2H^{2.75} (SC^\star)^{\frac{3}{4}} K^{\frac{1}{4}} \le \left(H^{0.75} (SC^\star)^{\frac{1}{4}} K^{\frac{1}{4}} \right)^2 + \left(H^2 (SC^\star)^{\frac{1}{2}} \right)^2 = \sqrt{H^3 SC^\star K} + H^4 SC^\star
$$

and (ii) holds by letting $x \coloneqq \max_{h\in[H]} \sum_{k=1}^{K} \sum_{s\in\mathcal{S}} d_h^{\pi^\star}(s) \left(V_h^\star(s) - V_h^k(s)\right)$ and solving the inequality $x \lesssim \sqrt{H^4 SC^\star \iota^3 x} + \sqrt{H^3 SC^\star K\iota^5} + H^4 SC^\star \iota^4$. This concludes the proof.

6 Discussions

This paper focuses on model-free paradigms for offline reinforcement learning (RL), addressing the challenges of insufficient data coverage and sample scarcity in historical datasets. We introduce and analyze pessimistic variants of Q-learning for finite-horizon Markov decision processes, employing lower confidence bounds (LCB) and variance reduction techniques to improve sample efficiency. Our results demonstrate that model-free algorithms, when combined with pessimism, can achieve near-optimal sample complexities under the single-policy concentrability assumption, thereby relaxing the need for full state-action coverage.

Furthermore, this work opens several exciting directions for future research. Pessimistic Q-learning algorithms could be integrated with optimistic counterparts when additional online data becomes available, enabling further policy refinement. Additionally, while the proposed algorithms are sample-optimal for a limited range of accuracy ($\varepsilon \in (0, 1/H]$), extending this range without compromising sample efficiency remains a critical question. Finally, adapting these methods to scenarios involving low-complexity function approximation offers significant potential for broader practical applications.

References

[AKY20] Alekh Agarwal, Sham Kakade, and Lin F Yang. Model-based reinforcement learning with a generative model is minimax optimal. *Conference on Learning Theory*, pages 67–83, 2020.

[AOM17] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, pages 263–272. JMLR. org, 2017.

[Ber17] Dimitri P Bertsekas. *Dynamic programming and optimal control (4th edition)*. Athena Scientific, 2017.

[BGB20] Jacob Buckman, Carles Gelada, and Marc G Bellemare. The importance of pessimism in fixed-dataset policy optimization. In *International Conference on Learning Representations*, 2020.

[BS12] Carolyn L Beck and Rayadurgam Srikant. Error bounds for constant step-size Q-learning. *Systems & control letters*, 61(12):1203–1208, 2012.

[BXJW19] Yu Bai, Tengyang Xie, Nan Jiang, and Yu-Xiang Wang. Provably efficient Q-learning with low switching cost. In *Advances in Neural Information Processing Systems*, pages 8002–8011, 2019.

[CJ19] Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pages 1042–1051. PMLR, 2019.

[CMSS20] Zaiwei Chen, Siva Theja Maguluri, Sanjay Shakkottai, and Karthikeyan Shanmugam. Finite-sample analysis of stochastic approximation using smooth convex envelopes. *arXiv preprint arXiv:2002.00874*, 2020.

[DCL+17] Simon S Du, Jianshu Chen, Lihong Li, Lin Xiao, and Dengyong Zhou. Stochastic variance reduction methods for policy evaluation. In *International Conference on Machine Learning*, pages 1049–1058. PMLR, 2017.

[DMKV21] Omar Darwiche Domingues, Pierre Ménard, Emilie Kaufmann, and Michal Valko. Episodic reinforcement learning in finite MDPs: Minimax lower bounds revisited. In *Algorithmic Learning Theory*, pages 578–598. PMLR, 2021.

[DRV+21] Robert Dadashi, Shideh Rezaeifar, Nino Vieillard, Léonard Hussenot, Olivier Pietquin, and Matthieu Geist. Offline reinforcement learning with pseudometric learning. *arXiv preprint arXiv:2103.01948*, 2021.

[DWCW19] Kefan Dong, Yuanhao Wang, Xiaoyu Chen, and Liwei Wang. Q-learning with UCB exploration is sample efficient for infinite-horizon MDP. *arXiv preprint arXiv:1901.09311*, 2019.

[EDM03] Eyal Even-Dar and Yishay Mansour. Learning rates for Q-learning. *Journal of Machine Learning Research*, 5:1–25, 2003.

[FMP19] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062. PMLR, 2019.

[Fre75] David A Freedman. On tail probabilities for martingales. *the Annals of Probability*, pages 100–118, 1975.

[JAZBJ18] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pages 4863–4873, 2018.

[JJS94] Tommi Jaakkola, Michael I Jordan, and Satinder P Singh. Convergence of stochastic iterative dynamic programming algorithms. In *Advances in neural information processing systems*, pages 703–710, 1994.

[JJWJL20] Mehdi Jafarnia-Jahromi, Chen-Yu Wei, Rahul Jain, and Haipeng Luo. A model-free learning algorithm for infinite-horizon average-reward MDPs with near-optimal regret. *arXiv preprint arXiv:2006.04354*, 2020.

[JYW21] Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline RL? In *International Conference on Machine Learning*, pages 5084–5096, 2021.

[JZ13] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

[KPR+20] Koulik Khamaru, Ashwin Pananjady, Feng Ruan, Martin J Wainwright, and Michael I Jordan. Is temporal difference learning optimal? an instance-dependent analysis. *arXiv preprint arXiv:2003.07337*, 2020.

[KRNJ20] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *arXiv preprint arXiv:2005.05951*, 2020.

[KZTL20] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191. Curran Associates, Inc., 2020.

[LCC+21] Gen Li, Changxiao Cai, Yuxin Chen, Yuantao Gu, Yuting Wei, and Yuejie Chi. Is Q-learning minimax optimal? a tight sample complexity analysis. *arXiv preprint arXiv:2102.06548*, 2021.

[LGR12] Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning*, pages 45–73. Springer, 2012.

[LKTF20] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[LSAB20] Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Provably good batch reinforcement learning without great exploration. *arXiv preprint arXiv:2007.08202*, 2020.

[LSC+21] Gen Li, Laixi Shi, Yuxin Chen, Yuantao Gu, and Yuejie Chi. Breaking the sample complexity barrier to regret-optimal model-free reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[LWC+20] Gen Li, Yuting Wei, Yuejie Chi, Yuantao Gu, and Yuxin Chen. Breaking the sample size barrier in model-based reinforcement learning with a generative model. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[LWC+21] Gen Li, Yuting Wei, Yuejie Chi, Yuantao Gu, and Yuxin Chen. Sample complexity of asynchronous Q-learning: Sharper analysis and variance reduction. *IEEE Transactions on Information Theory*, 68(1):448–473, 2021.

[MKS+15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, and Georg Ostrovski. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[Put14] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[QW20] Guannan Qu and Adam Wierman. Finite-time analysis of asynchronous stochastic approximation and Q-learning. In *Conference on Learning Theory*, pages 3185–3205. PMLR, 2020.

[RDV+21] Shideh Rezaeifar, Robert Dadashi, Nino Vieillard, Léonard Hussenot, Olivier Bachem, Olivier Pietquin, and Matthieu Geist. Offline reinforcement learning as anti-exploration. *arXiv preprint arXiv:2106.06431*, 2021.

[RYJR20] Nived Rajaraman, Lin Yang, Jiantao Jiao, and Kannan Ramchandran. Toward the fundamental limits of imitation learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[RZM+21] Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Neural Information Processing Systems (NeurIPS)*, 2021.

[SSS+17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[SWW+18] Aaron Sidford, Mengdi Wang, Xian Wu, Lin Yang, and Yinyu Ye. 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, 2018.

[SWWY18] Aaron Sidford, Mengdi Wang, Xian Wu, and Yinyu Ye. 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, 2018.

[US21] Masatoshi Uehara and Wen Sun. Pessimistic model-based offline reinforcement learning under partial coverage. *arXiv preprint arXiv:2107.06226*, 2021.

[UZS21] Masatoshi Uehara, Xuezhou Zhang, and Wen Sun. Representation learning for online and offline RL in Low-rank MDPs. *arXiv preprint arXiv:2110.04652*, 2021.

[Ver18] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[Wai19] Martin J Wainwright. Variance-reduced Q-learning is minimax optimal. *arXiv preprint arXiv:1906.04697*, 2019.

[Wat89] C. J. C. H. Watkins. Learning from delayed rewards. *PhD thesis, King's College, University of Cambridge*, 1989.

[WD92] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[WHY+19] Hoi-To Wai, Mingyi Hong, Zhuoran Yang, Zhaoran Wang, and Kexin Tang. Variance reduced policy evaluation with smooth function approximation. *Advances in Neural Information Processing Systems*, 32:5784–5795, 2019.

[XCJ+21] Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. *arXiv preprint arXiv:2106.06926*, 2021.

[XJW+21] Tengyang Xie, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. *arXiv preprint arXiv:2106.04895*, 2021.

[XWZL19] Tengyu Xu, Zhe Wang, Yi Zhou, and Yingbin Liang. Reanalysis of variance reduced temporal difference learning. In *International Conference on Learning Representations*, 2019.

[YBW21a] Ming Yin, Yu Bai, and Yu-Xiang Wang. Near-optimal offline reinforcement learning via double variance reduction. *Advances in neural information processing systems*, 34, 2021.

[YBW21b] Ming Yin, Yu Bai, and Yu-Xiang Wang. Near-optimal provable uniform convergence in offline policy evaluation for reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1567–1575. PMLR, 2021.

[YDWW22] Ming Yin, Yaqi Duan, Mengdi Wang, and Yu-Xiang Wang. Near-optimal offline reinforcement learning with linear representation: Leveraging variance information with pessimism. *arXiv preprint arXiv:2203.05804*, 2022.

[YKC+21] Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Sergey Levine, and Chelsea Finn. Conservative data sharing for multi-task offline reinforcement learning. *arXiv preprint arXiv:2109.08128*, 2021.

[YKR+21] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *arXiv preprint arXiv:2102.08363*, 2021.

[YLCF22a] Yuling Yan, Gen Li, Yuxin Chen, and Jianqing Fan. The efficacy of pessimism in asynchronous Q-learning. *arXiv preprint arXiv:2203.07368*, 2022.

[YLCF22b] Yuling Yan, Gen Li, Yuxin Chen, and Jianqing Fan. Model-based reinforcement learning is minimax optimal for offline zero-sum Markov games. *arXiv preprint arXiv:2206.04044*, 2022.

[YTY+20] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. *arXiv preprint arXiv:2005.13239*, 2020.

[YW21] Ming Yin and Yu-Xiang Wang. Towards instance-optimal offline reinforcement learning with pessimism. *Advances in neural information processing systems*, 34, 2021.

[YYD21] Kunhe Yang, Lin Yang, and Simon Du. Q-learning with logarithmic regret. In *International Conference on Artificial Intelligence and Statistics*, pages 1576–1584. PMLR, 2021.

[ZJD20] Zihan Zhang, Xiangyang Ji, and Simon S Du. Is reinforcement learning more difficult than bandits? a near-optimal algorithm escaping the curse of horizon. *arXiv preprint arXiv:2009.13503*, 2020.

[ZWB21] Andrea Zanette, Martin J Wainwright, and Emma Brunskill. Provable benefits of actor-critic methods for offline reinforcement learning. *arXiv preprint arXiv:2108.08812*, 2021.

[ZZJ20a] Zihan Zhang, Yuan Zhou, and Xiangyang Ji. Almost optimal model-free reinforcement learning via reference-advantage decomposition. *Advances in Neural Information Processing Systems*, 33, 2020.

[ZZJ20b] Zihan Zhang, Yuan Zhou, and Xiangyang Ji. Model-free reinforcement learning: from clipped pseudo-regret to sample complexity. *arXiv preprint arXiv:2006.03864*, 2020.