Stabilizing Q-learning with Linear Architectures for Provably Efficient Learning

Andrea Zanette [1] **Martin J. Wainwright** [2]

Abstract

The Q-learning algorithm is a simple and widely-used stochastic approximation scheme for reinforcement learning, but the basic protocol can exhibit instability in conjunction with function approximation. Such instability can be observed even with linear function approximation. In practice, tools such as target networks and experience replay appear to be essential, but the individual contribution of each of these mechanisms is not well understood theoretically. This work proposes an exploration variant of the basic Q-learning protocol with linear function approximation. Our modular analysis illustrates the role played by each algorithmic tool that we adopt: a second order update rule, a set of target networks, and a mechanism akin to experience replay. Together, they enable state of the art regret bounds on linear MDPs while preserving the most prominent feature of the algorithm, namely a space complexity independent of the number of step elapsed. We show that the performance of the algorithm degrades very gracefully under a novel and more permissive notion of approximation error. The algorithm also exhibits a form of instance-dependence, in that its performance depends on the "effective" feature dimension.

1. Introduction

The Q-learning algorithm (Watkins, 1989) is a classical and widely-used method for estimating optimal Q-value functions. As a stochastic approximation procedure for solving the Bellman fixed point equation, it comes with strong convergence guarantees when applied to tabular Markov decision processes (e.g., (Tsitsiklis, 1994; Kearns & Singh, 1999; Even-Dar et al., 2003; Wainwright, 2019a; Li et al., 2021)). When combined with function approximation, however, the basic Q-learning algorithm need not converge, and can exhibit instability. This challenge has motivated various proposals for stabilizing the updates. Among other modifications, experience replay is one ingredient that seems essential to state-of-the-art performance. From a theoretical point of view, however, these mechanisms are not well understood. This state of affairs leaves us with the following open question: is it possible to derive a stable Q-learning procedure with rigorous guarantees for a broad class of problem instances?

On one hand, recent work has unveiled information-theoretic barriers applicable to any algorithm (Weisz et al., 2020; Zanette, 2020; Wang et al., 2020a; 2021; Weisz et al., 2021; Foster et al., 2021). On the other hand, there exist several MDP models for which sample-efficient RL is possible. In particular, a recent line of papers (Krishnamurthy et al., 2016; Jiang et al., 2017; Sun et al., 2018; Zanette et al., 2020b; Jin et al., 2021; Du et al., 2021) provide analyses of RL procedures for certain MDP classes, and provide procedures that have polynomial sample complexity, albeit with non-polynomial computational complexity.

The starting point of this paper is to study Q-learning in some settings in which model-free algorithms[1] admit polynomial-time implementation. Examples include the class of low-rank MDPs (Jin et al., 2020; Zanette et al., 2020a; Agarwal et al., 2020b;a; Zanette et al., 2021a), and various generalizations thereof (Wang et al., 2019; 2020b). Although the underlying algorithms are polynomial-time, they can still require prohibitive amounts of computation and storage in practical settings. For instance, the memory requirement scales linearly with the amount of experience collected, which limits its practical applicability.

The Q-learning algorithm is popular in applications precisely because of its low computational complexity, as well

[1] Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, USA [2] Department of Electrical Engineering and Computer Sciences and Department of Statistics, University of California, Berkeley, USA. Correspondence to: Andrea Zanette <zanette@berkeley.edu>, Martin J. Wainwright <wainwrig@berkeley.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

[1] Sample efficient learning algorithms have also been obtained for other settings, see the papers (Ayoub et al., 2020; Modi et al., 2020; 2021).

as memory requirements that do not scale with the iteration count. Thus, we are led to ask whether it is possible to devise a version of Q-learning that is provably efficient when applied to low-rank MDPs. We address this question in the general exploration setting, so that a number of challenges come into play, including credit assignment, moving targets, and distribution shift.

1.1. Our contributions

The main contribution of this paper is to design and analyze a variant of the Q-learning algorithm that is guaranteed to minimize regret over the class of low-rank MDPs. Three main ingredients are key in our analysis: (1) a second-order update rule for improved statistical efficiency; a set of target networks (Mnih et al., 2015) to stabilize the updates, and most importantly, a replay mechanism called *policy replay*. This mechanism is similar to experience replay used in the deep RL literature (e.g., (Mnih et al., 2013)). While the second-order scheme and the target networks have been used in the optimization and the RL literature before, the policy replay mechanism is one key reinforcement learning contribution made in this paper. It stabilizes the learning process by eliminating the distribution shift problem that naturally arises when converging to an optimal controller.

Taken together, these algorithmic tools yield state-of-the-art regret bounds on H-horizon low-rank MDPs with d-dimensional feature representations. At the same time, they preserve one of the most important features of Q-learning, namely a memory requirement that—thanks to the policy replay mechanism—grows only logarithmically with sample size.

We now provide an informal preview of our main result. We consider an MDP with a finite action space of cardinality $|\mathcal{A}|$, and take (rescaling as needed) the optimal value function to be bounded in $[0, 1]$. Letting K the number of episodes elapsed, we have the following:

Theorem 1 (Informal statement)**.** *There is a Q-learning algorithm that achieves the regret upper bound $\widetilde{\mathcal{O}}(H^2 d^{3/2}\sqrt{K})$ while using $\widetilde{\mathcal{O}}(d^3 H^2)$ storage and per-step computational complexity $\mathcal{O}(d^2|\mathcal{A}|)$.*

To our knowledge, this is the first regret bound for Q-learning with any function approximator, which makes it the first algorithm with bounded memory complexity for the considered setting. The regret bound is competitive with the state-of-the-art results (Jin et al., 2020), in particular sub-optimal by a factor of H in the regret bound.

In this work, we also introduce a new notion of model misspecification, one especially well-suited to the analysis of temporal difference RL algorithms. It is a much weaker requirement than the ℓ_∞-norm bounds on mis-specification adoped in prior analyses; instead, it involves the expected off-policy prediction error. To the best of our knowledge, this leads to the mildest form of approximation error control for regret-minimizing algorithms using temporal differences and function approximation.

Our results are also partially instance-dependent, in the sense that we obtain faster rates for "easier problems". In particular, we show that the dimension d in Theorem 1 can (mostly) replaced by the *effective dimension*, a quantity that can be much smaller. We are not aware of instance-dependent results of this type when the algorithm is *not* provided with side knowledge of the problem structure.

Due to space limitation, the relation with past work is discussed in Appendix A.1.

2. Background and problem formulation

2.1. Finite-horizon Markov decision proceses

In this paper, we focus on finite-horizon Markov decision processes; see the standard references (Puterman, 1994; Bertsekas & Tsitsiklis, 1996) for more background and detail. A finite-horizon MDP is specified by a positive integer H, and events take place over a sequence of stages indexed by the time step $h \in [H] \overset{def}{=} \{1, \ldots, H\}$. The underlying dynamics involve a state space $\mathcal{S}$, and are controlled by actions that take values in some action set $\mathcal{A}$. In the analysis of this paper, the state space is allowed to be arbitrary (discrete or continuous), but we restrict to a finite action space.

For each time step $h \in [H]$, there is a reward function $r_h : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, and for every time step h and state-action pair (s, a), there is a probability transition function $\mathbb{P}_h(\cdot \mid s, a)$. When at horizon h, if the agent takes action a in state s, it receives a random reward drawn from a distribution $R_h(s, a)$ with mean $r_h(s, a)$, and it then transitions randomly to a next state s' drawn from the transition function $\mathbb{P}_h(\cdot \mid s, a)$.

A policy π_h at stage h is a mapping from the state space $\mathcal{S}$ to the action space $\mathcal{A}$. Given a full policy $\pi = (\pi_1, \ldots, \pi_H)$, the state-action value function at time step h is given by

$$Q_h^\pi(s, a) = r_h(s, a) + \mathbb{E}_{S_\ell \sim \pi | (s,a)} \sum_{\ell=h+1}^{H} r_\ell(S_\ell, \pi_\ell(S_\ell)),$$

where the expectation is over the trajectories induced by π upon starting from the pair (s, a). When we omit the starting state-action pair (s, a), the expectation is intended to start from a fixed state denoted by s_1. Any policy is associated with a value function $V_h^\pi(s) = Q_h^\pi(s, \pi_h(s))$, along with a Bellman evaluation operator

$$\mathcal{T}_h^\pi(Q_{h+1})(s, a) = r_h(s, a) \\ + \mathbb{E}_{S' \sim \mathbb{P}_h(s,a)} \mathbb{E}_{A' \sim \pi} Q_{h+1}(S', A').$$

Under some regularity conditions (Puterman, 1994; Shreve

& Bertsekas, 1978), there always exists an optimal policy $\pi^\star$ whose value and action-value functions achieve the suprema

$$V_h^\star(s) = V_h^{\pi^\star}(s) = \sup_\pi V_h^\pi(s), \quad \text{and}$$
$$Q_h^\star(s,a) = Q_h^{\pi^\star}(s,a) = \sup_\pi Q_h^\pi(s,a).$$

uniformly over all states and actions. We use $\mathbb{E}_\pi[\phi_h] \overset{def}{=} \mathbb{E}_{(S_h,A_h)\sim\pi}[\phi_h(S_h,A_h)]$ to denote the expected feature vector at timestep h.

We analyze algorithms that produce sequences of policies $\{\pi^1,\dots,\pi^K\}$, and for any such sequence, we define the regret

$$\textsc{Regret}(K) \overset{def}{=} \sum_{k=1}^K \mathbb{E}_{s_1\sim\rho}\left(V_1^\star - V_1^{\pi^k}\right)(s_1). \tag{1}$$

Whenever we have a sequence $n^1,\dots,n^k$ of values we denote with $n^{1:k} = \sum_{i=1}^k n^i$ their sum.

2.2. Structural conditions

Let now us lay out some assumptions on the MDPs and function approximation schemes.

2.2.1. Linear function approximations

For each $h \in [H]$, let $\phi_h : \mathcal{S}\times\mathcal{A} \mapsto \mathbb{R}^d$ be a given feature map. Throughout this paper, we assume the uniform boundedness condition

$$\sup_{s,a} \|\phi_h(s,a)\|_2 \le 1 \qquad \text{for all } h\in[H]. \tag{2}$$

For a given parameter vector $\theta_h \in \mathbb{R}^d$, define the function $f_{h,\theta}(s,a) \overset{def}{=} \langle \phi_h(s,a),\, \theta_h\rangle$. With a slight abuse of notation, given a partitioned vector $\theta = (\theta_1,\dots,\theta_H) \in (\mathbb{R}^d)^H$, we use the shorthand $f_\theta = (f_{1,\theta_1},\dots,f_{H,\theta_H})$ for the associated collection of functions.

In this paper, we study algorithms that produce linear functions in the class

$$\mathcal{Q}^{(\text{lin})} \overset{def}{=} \Big\{ f_\theta \mid \|\theta_h\|_2 \le 1 \quad \text{for all } h\in[H]\Big\}. \tag{3a}$$

Note that the bounded feature map condition (2), in conjunction with the Cauchy-Schwarz inequality, implies that

$$\|f_{h,\theta_h}\|_\infty = \sup_{s,a} |f_{h,\theta_h}(s,a)| \le 1 \qquad \text{for any } f_\theta \in \mathcal{Q}^{(\text{lin})}.$$

Consequently, the function class is contained with the larger class of action-value functions $(s,a)\mapsto Q_h(s,a)$ that are uniformly bounded in sup-norm—more precisely, the class

$$\mathcal{Q}^{(\text{all})} \overset{def}{=} \{(Q_1,\dots Q_H) \mid \|Q_h\|_\infty \le 1 \quad \text{for all } h\in[H]\}. \tag{3b}$$

The definitions above can be specialized for a specific timestep h in a natural way, in which case we denote the corresponding function spaces by $\mathcal{Q}_h^{(\text{lin})}$ and $\mathcal{Q}_h^{(\text{all})}$.

2.2.2. Bellman conditions

Our work covers both the settings with low inherent Bellman error (e.g., (Munos & Szepesvári, 2008; Zanette et al., 2020b)) as well as low-rank MDPs (e.g., (Yang & Wang, 2020; Jin et al., 2020)), which we introduce next. In both cases we assume that $\mathcal{Q}_{H+1}^{(\text{lin})} = \mathcal{Q}_{H+1}^{(\text{all})} = \{0\}$.

Assumption 1 (Bellman closure). *We say that an MDP and a feature representation ϕ have* zero inherent Bellman error *if for each $h\in[H]$ and any $Q_{h+1}\in\mathcal{Q}_{h+1}^{(lin)}$, there exists $Q_h \in \mathcal{Q}_h^{(lin)}$ such that $Q_h = \mathcal{T}_h Q_{h+1}$.*

Assumption 2 (Low-Rank). *An MDP is low rank with respect to the feature representation ϕ if for each $h\in[H]$, the following holds:*

$$\forall Q_{h+1} \in \mathcal{Q}_{h+1}^{(all)}, \textit{ there exists } Q_h \in \mathcal{Q}_h^{(lin)} \textit{ s.t. } Q_h = \mathcal{T}_h Q_{h+1}.$$

It can be shown that the class of low-rank MDP models is strictly contained within the class of MDPs with zero inherent Bellman error; see the paper (Zanette et al., 2020b) for further details.

Model misspecification: When the representation conditions do not exactly hold, we need to measure model misspecification. With this aim, we introduce two definitions of model misspecification that are appropriate for RL with temporal difference methods. The first one measures the violation of Assumption 1 with respect to a stationary external controller, while the second one measures the violation with respect to Assumption 2 when a single stationary controller is not available.

Before stating the definitions, let us introduce some more notation and terminology along with their motivation. Let π be a policy that generates a dataset used to fit a predictor. Using the data generated by π, we will make predictions about a target policy $\overline{\pi}$ which could be arbitrary. The predictor that we seek should fit $\mathcal{T}Q'$ where $Q'\in\mathcal{Q}^{(\text{lin})}$ or $Q'\in\mathcal{Q}^{(\text{all})}$ depending on whether we seek to quantify the violation of Assumption 1 or Assumption 2, respectively. Accordingly, define the population minimizer $\theta^{\pi,Q',h}$ along π with Q' as next state value function as

$$\theta^{\pi,Q',h} \overset{def}{=} \arg\min_{\theta\in\mathcal{B}} \mathbb{E}_{(S_h,A_h)\sim\pi} \Big\{\langle\phi_h(S_h,A_h),\theta\rangle - (\mathcal{T}_h Q')(S_h,A_h))\Big\}^2. \tag{4}$$

Let us now state a definition of model misspecification that measures the violation with respect to Assumption 1 *(Bellman closure)* whenever there exists an external stationary controller π. This definition involves a non-negative error term $\nu \geq 0$ referred to as *transfer error*.

Definition 1 (Model Misspecification w.r.t. Bellman Closure)**.** *An MDP and a feature map ϕ are ν-misspecified with respect to the Bellman closure condition and the stationary policy π if for any policy $\overline{\pi}$ and action-value function $Q' \in \mathcal{Q}^{(lin)}$ the best on policy fit $Q_h : (s,a) \mapsto \langle \phi_h(s,a), \theta^{\pi,Q',h} \rangle$ along π satisfies the bound*

$$\Big| \sum_{h=1}^{H} \mathbb{E}_{(S_h,A_h)\sim\overline{\pi}} \Big[Q_h(S_h,A_h) - (\mathcal{T}_h Q'_{h+1})(S_h,A_h) \Big] \Big| \leq \nu. \tag{5}$$

In summary, Definition 1 measures the average Bellman error that arises when evaluating the predictor fit on the controller's distribution along other distributions. This is a significantly more generous requirement than ℓ_∞ model misspecification, and is algorithm-independent. Notice that the expectation is inside the absolute value. We conclude by presenting an extension of Definition 1, one that applies to the exploration setting where there is no single stationary controller that generates the dataset.

Definition 2 (Model Misspecification w.r.t. Low Rank)**.** *An MDP and a feature map ϕ are ν-misspecified with respect to the low rank condition if for any two policies $\pi, \overline{\pi}$ and action value function $Q' \in \mathcal{Q}^{(all)}$, the best on policy fit $Q_h : (s,a) \mapsto \langle \phi_h(s,a), \theta^{\pi,Q',h} \rangle$ satisfies the bound* (5)*.*

The primary distinction between Definition 1 and Definition 2 is that the latter needs to hold when $Q' \in \mathcal{Q}^{(\mathrm{all})}$ instead of just $Q' \in \mathcal{Q}^{(\mathrm{lin})}$.

3. Algorithms

This section is devoted to a description of the Q-learning procedures analyzed in this paper. We begin by providing some intuition for our algorithms in Section 3.1. Section 3.2 is devoted to the description of *Stabilized, Second-Order, Streaming Q-learning* algorithm, or S^{3}Q-LEARNING for short. It corresponds to a stabilized and streaming form of Q-learning that estimates the optimal policy based on data drawn from some fixed (stationary) controller policy. We use this algorithm as a building block for the more sophisticated algorithm described in Section 3.3, which allows for the data-generating policy to also change, essential to obtaining an overall scheme with low regret. We refer to this procedure as *Sequentially Stabilized Second-order Streaming Q-learning*, or S^{4}Q-LEARNING for short.

3.1. Some intuition

Let us begin by providing some intuition for the algorithms that are proposed and analyzed in this paper. When the basic form of Q-learning is implemented with linear function approximation, the updates are performed directly on the parameter θ associated with the linear representation. Upon observing the tuple (s_h, a_h, r_h, s'_h), representing the experienced state, action, reward and successor state at level h, the update rule for a user defined learning rate $\alpha \in \mathbb{R}$ takes the familiar form

$$\theta_h \leftarrow \theta_h - \alpha \Delta \phi_h(s_h,a_h) \quad \text{where} \tag{6}$$
$$\Delta = \langle \phi_h(s_h,a_h), \theta_h \rangle - r_h - \max_{a'} \langle \phi_{h+1}(s'_h,a'), \theta_{h+1} \rangle.$$

Although Q-learning is a form of stochastic approximation, as are stochastic gradient methods, the above update is not equivalent to stochastic gradient. However, for the purposes of analysis, it is useful to consider some restrictions under which it can be related to a stochastic gradient method.

For a moment, let us additionally assume that (a) the next timestep parameter θ_{h+1} is never updated, and (b) the tuple (s_h, a_h, r_h, s'_h) is drawn from a stationary distribution. When these conditions are met, the update (6) corresponds to a stochastic gradient update as applied to the squared loss

$$\theta_h \mapsto \mathbb{E}_{(S_h,A_h,R_h,S'_h)} \Big[\underbrace{\langle \phi_h(S_h,A_h), \theta_h \rangle}_{\text{predictor}} - \underbrace{\big(R_h + \max_{a'} \langle \phi_{h+1}(S'_h,a'), \theta_{h+1} \rangle\big)}_{\text{fixed target function}} \Big]^2, \tag{7}$$

where the expectation is over the stationary distribution that generates the data. This is an algorithm that we know how to analyze.

With this perspective in place, our high-level idea is to enforce these two conditions—namely, a fixed target and a stationary distribution for drawing samples. However, so as to be able to estimate an optimal policy while incurring low regret, the next state-value function and the sampling distribution cannot be "locked in" forever, but instead need to evolve with time. The core algorithmic contribution of this paper is the design of a device to periodically update the next-state value function and the sampling distribution so as to allow convergence to an optimal controller in a stable way. In addition, we use a second-order update in place of the first-order scheme (6) so as to achieve improved statistical efficiency.

We first describe an algorithm for the controlled setting, in which stream of states, actions, rewards and transitions are generated by a stationary controller. In this case, only the next-state action value function needs to be updated periodically, because the distribution that generates the experience is fixed. Next, we design a meta-algorithm that performs exploration while additionally ensuring that the Q-learning update rule is fed with data from a stationary controller.

3.2. S^{3}Q-LEARNING

In this section, we introduce the *Stabilized, Second-Order, Streaming Q-learning* algorithm, or S^{3}Q-LEARNING for short. The algorithm takes as input a stationary controller policy π that generates a stream of states, actions, rewards and transitions. Target networks are used to stabilize the value function updates, which are performed via a second-order update rule for improved statistical efficiency. This is a streaming algorithm, meaning that each sample is immediately processed and then discarded.

Learning mechanics: The S^{3}Q-LEARNING algorithm is detailed in Algorithm 1. The algorithm proceeds over a sequence of epochs, denoted by e in the algorithm. Each epoch is handled by the outermost "while" loop. Within each epoch, the algorithm sequentially updates the target networks[2] $\widehat{Q}^{tar}_h$ at each level ℓ proceeding backward from $\ell = H$ to $\ell = 1$. This order of updates ensures that the next-timestep ($\ell + 1$) target network is always up to date to compute the bootstrapped Q values needed at level ℓ to compute the temporal difference (TD) error (see Line 12 and Eq. (8)). When the update has completed in every level $\ell \in [H]$, the target networks are stored in the predictor $\widehat{Q}^{\star}$, which is the one that the algorithm considers to be the "best" estimate of $Q^{\star}$. At this point, a new epoch begins.

Let us now describe the update rule in Lines 12 and 13. At each timestep h the algorithm observes a tuple of state, action, reward and transition (s_h, a_h, r_h, s_{h+1}) and uses them to update $\widehat{\theta}_h$. To be clear, $\widehat{\theta}_h$ is associated to a network different from the target network $\widehat{Q}^{tar}_h$ for that timestep. To perform the update, the algorithm first computes the temporal difference error

$$\mathrm{TD}_h \overset{def}{=} r_h + \max_{a'} \widehat{Q}^{tar}_{h+1}(s_{h+1}, a') - \langle \phi_h(s_h, a_h)), \widehat{\theta}_h \rangle \tag{8}$$

in Line 12 and then updates the network parameter $\widehat{\theta}_h$ using a second order update rule (in place of Eq. (6)) together with the empirical covariance Σ_h^{-1}, see Line 13. Such update rule effectively minimizes the least-squares criterion (7), as it coincides with the Sherman-Morrison rank one update.

The stopping condition in line 8 can be any arbitrary stopping time; without it, the algorithm will simply keep running indefinitely.

3.3. S^{4}Q-LEARNING

When the stream of data is generated by a controller that is converging to an optimal one—a necessary condition to obtain low regret—the experience it generates is no longer stationary. We will now introduce a simple device, the *policy replay* mechanism, that allows the controller to evolve with time while ensuring that there is no distribution shift during the Q-learning updates. It leads to an algorithm that converges to an optimal controller under some assumptions. This is achieved by *sequentially* invoking S^{3}Q-LEARNING using stationary controllers that are increasingly more optimal; the resulting algorithm is called *Sequentially Stabilized Second-order Streaming Q-learning*, or S^{4}Q-LEARNING for short, and is detailed in Algorithm 2.

Policy replay for experience replay: The policy replay mechanism generates new experience using past policies. The past policies are stored in the *policy replay memory* $\Pi \overset{def}{=} \{(\pi_i, n_i)\}_{i=1}^{p}$, which contains a set of policies π_i associated to a number of samples n_i. The policy replay mechanism extracts a stationary mixture policy from Π, defined as the controller that plays each policy π_i with probability proportional to n_i for the full episode. Such mixture policy is taken as stationary controller to invoke S^{3}Q-LEARNING, along with a suitable exploration bonus to produce optimistic Q-values that guide the exploration. The stopping condition to be used in Line 8 in Algorithm 1 is the number of trajectories cHm_{tot} for an appropriate constant c, see line Line 6 in Algorithm 2.

The policy replay mechanism is similar in purpose to experience replay where the experience $\{(s_i, a_i, r_i, s'_i)\}$ generated so far is stored and used to retrain the network. However, unlike experience replay, the policy replay memory does not store the full dataset and instead just contains a 'recipe' for generating a statistically similar dataset by re-playing past policies. In this way, the memory complexity does not grow with the number of iterations beyond a mild logarithmic term, making our algorithm truly streaming. And while the policy replay mechanism requires additional samples, the regret remains well controlled because the policies in Π are progressively more and more near-optimal: in the limit, the policies in the policy replay memory generate samples with vanishing regret.

Having described the policy replay mechanism, we can now illustrate how S^{4}Q-LEARNING conducts exploration using such device.

Learning mechanics: The S^{4}Q-LEARNING algorithm proceeds in phases which are indexed by p inside the algorithm. At the beginning of each phase, the algorithm invokes the S^{3}Q-LEARNING subroutine with a controller π_{Control} that is a mixture policy among those in the current policy replay memory Π, as described in the prior paragraph. The S^{3}Q-LEARNING procedure then returns an optimistic action-value function estimate Q from which the greedy (optimistic) policy π is extracted in Line 7 of Algorithm 2.

[2] In this work we refer to the next-timestep linear approximator as to 'target network' for consistency with some of the Q-learning literature. However, notice that our 'networks' are linear.

Algorithm 1 S^{3}Q-LEARNING

```
 1: Input: Controller $\pi$, (optional) stopping condition, (optional) bonus function $b$
 2: $\widehat{Q}^{tar}_{\ell}(\cdot,\cdot) = 0,\ \forall \ell \in [H+1];\ e = 0$        ▷ Initialize target network and epoch counter
 3: while True do
 4:     $e = e + 1$                                                              ▷ New epoch begins
 5:     for level $\ell = H, H-1, \dots, 2, 1$ do
 6:         $\widehat{\theta}_\ell = 0;\ \Sigma_\ell = \lambda_{\text{Reg}} I;$                                   ▷ Initialize network and covariance
 7:         for $n^\ell = 1, \dots, 2^e$ do
 8:             if STOPPING CONDITION then return $\widehat{Q}^\star$
 9:             $s_1 \sim \rho$                                                    ▷ Get start state
10:             for timestep $h = 1, 2, \dots, H$ do
11:                 Play $a_h = \pi_h(s_h)$ and get $(r_h, s_{h+1})$; $\phi_h \overset{def}{=} \phi_h(s_h, a_h)$       ▷ Play and advance
12:                 $\text{TD}_h = r_h + \max_{a'} \widehat{Q}^{tar}_{h+1}(s_{h+1}, a') - \langle \phi_h, \widehat{\theta}_h \rangle,$        ▷ Compute TD error
13:                 $\widehat{\theta}_h \leftarrow \widehat{\theta}_h + \frac{\Sigma_h^{-1}\phi_h \text{TD}_h}{1+\|\phi_h\|^2_{\Sigma_h^{-1}}};\qquad \Sigma_h^{-1} \leftarrow \Sigma_h^{-1} - \frac{\Sigma_h^{-1}\phi_h\phi_h^\top\Sigma_h^{-1}}{1+\|\phi_h\|^2_{\Sigma_h^{-1}}}$   ▷ Update network and covariance
14:             end for
15:         end for
16:         $\widehat{\theta}^{tar}_\ell = \min_{\theta \in \mathcal{B}} \|\theta - \widehat{\theta}_\ell\|^2_{\Sigma_\ell}$                                 ▷ Project parameter
17:         $\widehat{Q}^{tar}_\ell(\cdot,\cdot) \leftarrow \langle \phi_\ell(\cdot,\cdot), \widehat{\theta}^{tar}_\ell \rangle$ or $\widehat{Q}^{tar}_\ell(\cdot,\cdot) \leftarrow \min\{1, \langle \phi_\ell(\cdot,\cdot), \widehat{\theta}^{tar}_\ell \rangle + b_\ell(\cdot,\cdot)\}$   ▷ Update target network
18:     end for
19:     $\widehat{Q}^\star \leftarrow \widehat{Q}^{tar}$                                                        ▷ Save best approximator
20: end while
```

The S^{4}Q-LEARNING algorithm then proceeds to collect trajectories from the greedy policy π until "sufficient progress" is made. In order to measure the progress, the agent maintains the accumulator T and updates it in Line 13. Once T is larger than a certain value (see Line 16), the procedure has made sufficient progress on the current data, so that Q network should be updated with fresh data. The triggering condition in Line 16 is essentially equivalent to checking that the determinant of the cumulative covariance has doubled with respect to that of the prior epoch (cf. a similar condition for linear bandits (Abbasi-Yadkori et al., 2011)). When the determinant doubles, the agent has acquired sufficient information and a new policy may be computed. An important difference here is that the determinant should refer to the expected cumulative covariance, which is unknown, and such determinant ratio must thus be estimated from data; our accumulator T performs such task.

In order to update the Q network, S^{4}Q-LEARNING constructs a new bonus function and adds the current greedy policy π to the policy replay memory (together with the number of trajectories that should be generated from such policy). A new phase can now begin with a call to S^{3}Q-LEARNING using a newly constructed, more optimal controller and smaller bonus function.

4. Main results

We now turn to the statement of our main results, along with discussion of some of their consequences. We begin in Section 4.1 by describing the form of the bonus function used in our algorithms, along with the effective dimension that appears in our bounds. Section 4.2 is devoted to our main result—namely, a performance guarantee for the S^{4}Q-LEARNING algorithm. In Section 4.3, we elaborate upon the guarantees for the S^{3}Q-LEARNING algorithm that underlie our main result.

4.1. Bonus function and effective dimension

Bonus function: The bonus function used in phase p is

$$b_h(\cdot,\cdot) \overset{def}{=} \alpha_h \|\phi_h(\cdot,\cdot)\|_{\Sigma_h^{-1}}, \qquad \text{where}$$

$$\alpha_h \overset{def}{=} c\left\{\sqrt{d \log\left(\frac{dpn^{(1:p)}}{\delta}\right)} + \sqrt{\lambda_{\text{Reg}}}\right\}. \tag{9}$$

Here $c > 0$ is a universal constant; we use $n^{(1:p)} = \sum_{i=1}^{p} n^{(i)}$ to denote the samples used in phases 1 through p; and Σ_h is a cumulative covariance matrix. The empirically estimated cumulative covariance Σ is constructed as follows. Let π_{Control} denote the controller used to call S^{3}Q-LEARNING in phase $p-1$, and let π denote the greedy policy used by S^{4}Q-LEARNING in the same phase. The cumulative covariance takes the form

$$\Sigma_h = \underbrace{\sum_{i=1}^{m_{\text{tot}}} \phi_{ih}\phi_{ih}^\top + \lambda_{\text{Reg}} I}_{\substack{\text{S}^3\text{Q-LEARNING} \\ \text{covariance} \\ \text{returned in phase } p-1}} + \underbrace{\sum_{j=1}^{m_p} \phi_{jh}\phi_{jh}^\top}_{\substack{\text{S}^4\text{Q-LEARNING} \\ \text{covariance} \\ \text{added in phase } p-1}}, \tag{10}$$

Algorithm 2 S^{4}Q-LEARNING

```
1: Input: Bonus function $b$, update trigger $T_{\text{Trig}} = \Theta(\log \frac{pK}{\delta})$
2: $\Pi_h = \emptyset;\ \forall h \in [H]$    ▷ Initialize policy replay memory
3: for phase $p = 1, 2, \dots$ do
4:     $m_{\text{tot}} = \sum_{j=1}^{p-1} m_j$ for $(\pi_j, m_j) \in \Pi$    ▷ Get total # trajectories to simulate
5:     $\pi_{\text{Control}} \overset{def}{=}$ at the start of the episode play $\pi_j$ with probability $m_j/m_{\text{tot}},\ \forall j \in [p-1]$    ▷ Define controller
6:     $(Q, \Sigma^{ref}) \leftarrow$S$^3$Q-LEARNING$(\pi_{\text{Control}}, cHm_{\text{tot}}, b),\ c \in \mathbb{R}$    ▷ Get optimistic $Q$ values
7:     $\pi_h(\cdot) = \arg\max_a Q_h(\cdot, a),\ \forall h \in [H]$    ▷ Extract greedy policy
8:     $\Sigma = \Sigma^{ref},\ m = 0,\ T_h = 0,\ \forall h \in [H]$    ▷ Initialize # trajectories and trigger value
9:     repeat
10:        $s_1 \sim \rho$    ▷ Get start state
11:        for $h = 1, 2, \dots, H$ do
12:            Play $a = \pi_h(s);\ m = m + 1$    ▷ Play and increment counter
13:            $T_h \leftarrow T_h + \|\phi_h(s,a)\|^2_{(\Sigma_h^{ref})^{-1}}; \Sigma_h \leftarrow \Sigma_h + \phi_h(s,a)\phi_h(s,a)^\top$    ▷ Increment accumulator and covariance
14:            Get next state $s^+$; $s \leftarrow s^+$    ▷ Advance
15:        end for
16:    until $\exists h \in [H]$ such that $T_h \geq T_{\text{Trig}}$
17:    $\Pi \leftarrow \Pi \cup \{(\pi, m)\};\ b_h(\cdot,\cdot) = \alpha_h \|\phi(\cdot,\cdot)\|_{\Sigma_h^{-1}}$    ▷ Update policy replay memory and bonus
18: end for
```

where $\phi_{ih} \sim \pi_{\text{Control}},\ \phi_{jh} \sim \pi$. Note that Σ_h is formed by the sum of of the cumulative covariance matrix returned by S^{3}Q-LEARNING along with additional terms computed by S^{4}Q-LEARNING between Line 9 to Line 16 in phase $p-1$.

Notice that the empirical covariance Σ_h is estimated de novo within each phase p, and due to statistical fluctuations, it does not grow monotonically across phases. Nonetheless, the covariance and the bonus are two devices to measure the progress of the algorithm.

Information gain and effective dimension: We now define the *effective dimension* $\widetilde{d}_h$ at time step h. It is a scalar quantity that governs the complexity of the exploration problem, defined as

$$\widetilde{d}_h = \max_\pi \log\Big(\det\big(I + \tfrac{n}{\lambda_{\text{Reg}}}\,\mathbb{E}_{\phi_h \sim \pi}[\phi_h \phi_h^\top]\big)\Big). \quad (11)$$

We note that this notion has been exploited in past work (Srinivas et al., 2009; Yang & Wang, 2020; Agarwal et al., 2020a; Du et al., 2021). The information gain can be much smaller than the dimensionality d of the feature vectors ϕ —that is, we can have $\widetilde{d}_h \ll d_h$. This scaling holds, for instance, when the feature moment matrix $\mathbb{E}_{\phi_h \sim \pi}[\phi_h \phi_h^\top]$ is mostly concentrated along few directions; see Lemma 11 in Appendix D.2 for details.

4.2. Guarantees for S^{4}Q-LEARNING

We are now ready to present our main result, namely a bound on the regret incurred by all the policies that generate rollouts, including those played by the S^{3}Q-LEARNING subroutine when it is called by S^{4}Q-LEARNING. We assume that the bonus function is defined according to Eq. (9) with an appropriately chosen universal constant.

Theorem 1 (Performance Bound of S^{4}Q-LEARNING)**.** *Consider an MDP that is ν-misspecified w.r.t. Bellman closure (cf. Definition 2). Then for any number of episodes, there exists an event $\mathcal{E}_K$ that holds with probability at least $1 - \delta$, and under this event, we have the following guarantees:*

(a) The average regret of S^{4}Q-LEARNING *is upper bounded as* $\text{AVEREG}(K) \leq$

$$cL\left\{\frac{H}{\sqrt{K}}\left[\Big(\sum_{h=1}^{H}\widetilde{d}_h\Big) \times \Big(\sum_{h=1}^{H}(d_h+1)\widetilde{d}_h\Big)\right]^{1/2} + \nu\right\} \quad (12)$$

where $L \overset{def}{=} \log\big(\frac{dp_{tot}K}{\delta}\big)$.

(b) The memory complexity is bounded as $\mathcal{O}(Ld^2H\sum_h \widetilde{d}_h) = \mathcal{O}(Ld^3H^2)$ while the per-step computational complexity is $\mathcal{O}(|\mathcal{A}|d^2)$.

See Appendix C for the proof of this claim.

In absence of model misspecification ($\nu = 0$) and the special case $d_h = \widetilde{d}_h$ for all h, the worst-case regret bound becomes $\widetilde{\mathcal{O}}\big(H^2\sqrt{d^3K}\big)$. Here $\widetilde{\mathcal{O}}$ includes constant and logarithmic factors.

Some comments: When the value function is rescaled to be in $[0, H]$ instead of the unit interval $[0, 1]$, the regret bound of S^{4}Q-LEARNING becomes $\widetilde{\mathcal{O}}(H^3\sqrt{d^3K})$, which is larger by only a factor of H relative to the state-of-the-art $\widetilde{\mathcal{O}}(H^2\sqrt{d^3K})$ bound available for computationally tractable algorithms (Jin et al., 2020).

This slightly sub-optimal sample complexity is counterbalanced by a number of advantages of S^{4}Q-LEARNING. One major benefit is the low memory footprint, which depends only on K via a logarithmic factor. To the best of our knowledge, all previous methods that apply in this setting need to store the full experience, leading to a memory requirement scaling at least linearly in the number of episodes K. For problems with a large number of interactions, this linear scaling can be prohibitive.

Additionally if the horizon is not very large, the S^{4}Q-LEARNING bound might be substantially tighter than the guarantees in Jin et al. (Jin et al., 2020), since it primarily scales with the effective dimension $\widetilde{d}$. To the best of our knowledge, our result gives the first adaptive and tractable algorithm for this setting with regret bounds depending on the effective dimension $\widetilde{d}$. In contrast, existing algorithms with regret bounds in terms of the effective dimension need to know its value to inherit an improved regret bound (Agarwal et al., 2020a; Yang & Wang, 2020); in this sense, they are non-adaptive guarantees.

Finally, our approximation error guarantees are new for value-based exploration algorithms and close some of the gaps with respect to policy gradient methods. The approximation error of S^{4}Q-LEARNING scales with the worst-case off-policy expected prediction error. To the best of our knowledge, temporal difference methods that perform exploration have only been analyzed with ℓ_∞-approximation guarantees. Instead, in policy gradient algorithms (Agarwal et al., 2020a; Zanette et al., 2021a), the approximation error is measured in expectation with respect to an arbitrary comparator. Our approximation error depends on the worst-case (with respect to the policies) approximation error, but the error is still measured in expectation, and furthermore, the expectation is inside the absolute value.

4.3. Guarantees for S^{3}Q-LEARNING

When a stationary controller is available, the S^{3}Q-LEARNING algorithm can be used to maintain a running estimate of the optimal action value function $Q^\star$ with a low memory footprint. This guarantee is of independent interest: the basic protocol illustrated in Algorithm 1 serves as a building block for sophisticated algorithms, with the S^{4}Q-LEARNING procedure analyzed in this paper being one example. Of course, in the controlled setting the quality of the value function estimate will also depend on how exploratory the controller policy is; some notation will be introduced shortly to quantify this.

Let us define the *expected cumulative covariance matrix* after n samples from the (controller) policy π as

$$\overline{\Sigma}_h(n) = n\left\{\mathbb{E}_{\phi_h\sim\pi}\left[\phi_h\phi_h^\top\right] + \lambda_{\text{Reg}}I\right\}, \tag{13}$$

where $\lambda_{\text{Reg}} > 0$ is a fixed positive regularization parameter. Given a stationary controller π and a bonus function b, the S^{3}Q-LEARNING algorithm returns a sequence of estimated Q-functions $\widehat{Q}^\star = (\widehat{Q}_1^\star, \ldots, \widehat{Q}_H^\star) \in \mathcal{Q}^{(\text{lin})}$. In this section, we state some bounds on the value function error $\widehat{Q}^\star - Q^\star$ when the bonus function $b = 0$ and the violation of Assumption 1 *(Bellman closure)* is measured according to Definition 1. (We consider the extension to the setting $b \geq 0$ in Appendix D).

Our analysis involves the Bellman error associated with $\widehat{Q}^\star$, defined for each $h \in [H]$ and (s,a) as the discrepancy of $\widehat{Q}^\star$ in satisfying the Bellman equations with $\widehat{Q}_{h+1}^\star$ as the next state value function:

$$\mathcal{E}_h(s,a) \overset{def}{=} \widehat{Q}_h^\star(s,a) - \big(\mathcal{T}_h\widehat{Q}_{h+1}^\star\big)(s,a). \tag{14}$$

Our analysis shows how this Bellman error can be bounded in terms of an uncertainty function and an approximation error. For a given integer sample size $n > 1$ and user-defined error probability $\delta \in (0,1)$, define the scalar quantity

$$\overline{\alpha}_h(n,\delta) = c\left\{\sqrt{d\log\left(\tfrac{dneH}{\delta}\right)} + \sqrt{\lambda_{\text{Reg}}}\right\}, \tag{15}$$

where $c > 0$ is a universal constant, whose specific value can be determined via the proof.

Suppose that the algorithm terminates after e_{tot} epochs, and let K be the total number of trajectories. For a given tolerance probability $\delta_{\text{master}} \in (0,1)$, we define the uncertainty function

$$\mathcal{U}_h(s,a) \overset{def}{=} \overline{\alpha}_h(n^*,\delta^*)\|\phi_h(s,a)\|_{\left(\overline{\Sigma}_h(n^*)\right)^{-1}} \tag{16a}$$

$$\text{where } n^* \overset{def}{=} \frac{K}{4H}, \text{ and } \delta^* \overset{def}{=} \frac{\delta_{\text{master}}}{2He_{tot}^2d}. \tag{16b}$$

We define the comparator error

$$\Delta_h^\pi(s,a) \overset{def}{=} (\mathcal{T}_h\widehat{Q}_{h+1}^{tar})(s,a) - \left\langle\phi_h(s,a),\, \theta^{\pi,\widehat{Q}_{h+1}^{tar},h}\right\rangle. \tag{16c}$$

In order to state the theorem, let $\mathbb{E}_\pi$ denote the expectation over the trajectories $(S_1, A_1, \ldots, S_H, A_H)$ induced by following π after sampling from the starting distribution ρ.

Theorem 2 (Performance bound for S^{3}Q-LEARNING)**.** *Consider an MDP that is ν-misspecified w.r.t. Bellman closure (cf. Definition 1). If the* S^{3}Q-LEARNING *algorithm is run with the uncertainty function* (16a)*, then for any episode K, with probability at least $1-\delta_{master}$, it returns a solution $\widehat{Q}^\star$ such that:*

(a) Its Bellman error function $\mathcal{E}_h \overset{def}{=} \widehat{Q}_h^\star - \big(\mathcal{T}_h \widehat{Q}_{h+1}^\star\big)$ *satisfies the pointwise bound for each* (s, a)

$$\Big| \mathcal{E}_h(s,a) + \Delta_h^\pi(s,a) \Big| \le \mathcal{U}_h(s,a). \tag{17a}$$

(b) For each $h \in [H]$*, the greedy policy* $\overline{\pi}_h(s) \overset{def}{=} \arg\max_a \widehat{Q}_h^\star(s,a)$ *satisfies the bounds*

$$- \mathbb{E}_{\pi^\star} \sum_{h=1}^{H} \mathcal{U}_h(S_\tau, A_h) - \nu \le \mathbb{E}_{S\sim\rho}(\widehat{V}_1^\star - V_1^\star)(S) \tag{17b}$$

$$\le \mathbb{E}_{\overline{\pi}} \sum_{h=1}^{H} \mathcal{U}_h(S_\tau, A_\tau) + \nu.$$

Furthermore, the memory complexity of the algorithm is bounded as $\mathcal{O}(d^2 H)$ *and the per-step computational complexity is bounded as* $\mathcal{O}(d^2 + |\mathcal{A}|d)$.

See Appendix B for the proof of this claim.

5. Discussion

In this paper, we have introduced several modifications to the basic Q-learning protocol so as to derive an exploratory procedure that operates with linear function approximation, and is equipped with performance guarantees while remaining computation and memory-efficient. It is natural to ask to what extent these modifications—the second-order update rule, the use of target networks and policy replay—are needed in order to obtain such guarantees.

On the second-order rule: Of the three ingredients, the second-order update rule only serves to improve the statistical efficiency. When the target network and sampling distribution are fixed, using first-order update rule in Eq. (6) would be essentially equivalen to a stochastic gradient update; such updates would minimizes the loss function (7) with a rate $1/\sqrt{n}$ instead of the $1/n$ enabled by our second-order updates. It should be observed that the higher cost and memory of the second-order rule are not a problem when conducting exploration, as the main computational bottleneck is the calculation of the bonus function, while the main memory requirement is due to the policy replay.

Nonetheless, if one is interested purely in the optimization setting, where neither the replay memory nor the exploration bonuses are needed, some techniques from variance reduction can be used in conjunction with a first-order update rule (Frostig et al., 2015; Li et al., 2020; Wainwright, 2019b; Mou et al., 2022) to lower the computational and space complexity while retaining high sample efficiency. We leave this as an interesting direction for future work.

On the use of target networks: The target networks considerably simplify the analysis of the algorithm by establishing a connection with linear regression with a fixed target. There is no real downside with adopting target networks, and whether they could be removed is left as future work.

On the policy replay mechanism: A truly critical ingredient in this work is the policy replay mechanism, which ensures that the Q-learning updates are performed on data generated by a stationary controller and with the most recent bonus and value function estimate. Without the replay mechanism, the network weights would be updated with a changing target—recall that the target networks need to be updated periodically—and under a non-stationary distribution. The main issue is that the target networks contain both statistical errors and bias due to the exploration bonus which decay with time. In this case, the high errors present in the target networks in early phases would hurt the algorithm in later phases. Discounting early updates by an appropriate learning rate to favor later updates may seem like a solution, but this can lead to "catastrophic forgetting" of past experience because the learning distribution is non-stationary. When using linear function approximation, a concrete consequence is that the algorithm can forget what it has learned along the directions it played in the early phases.

Exploration algorithms inspired by Least Square Value Iteration (LSVI) avoid these issues, because they update the weights of the network by computing the full least-squares solution using the most recent, and thus most accurate, target function. In this way, the next state value function is as accurate as possible over the full domain, and is perturbed everywhere by the most recent (and smallest) exploration bonus, one that truly reflects the current model uncertainty.

Likewise, experience replay would alleviate these issues by re-training the network using the most recent target network. Unfortunately, doing so seems to require a replay buffer of size proportional to all the experience collected so far, making Q-learning no longer a truly streaming algorithm. The policy replay memory is a simple solution to such issues, one that preserves the streaming nature of Q-learning.

Due to space constraints, future directions are discussed in Appendix A.2

Acknowledgment

This work was partially supported by National Science Foundation FODSI grant 2023505, DOD ONR Office of Naval Research N00014-21-1-2842, National Science Foundation DMS grant 2015454, and National Science Foundation CCF grant 1955450. Part of this work was completed while Andrea Zanette was visiting Learning and Games at the Simons Institute for the Theory of Computing. The authors are very grateful to the reviewers, as well as to the meta-reviewer, for identifying clarity issues present in an earlier draft.

References

Abbasi-Yadkori, Y., Pal, D., and Szepesvari, C. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

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

Agarwal, A., Kakade, S., Krishnamurthy, A., and Sun, W. Flambe: Structural complexity and representation learning of low rank mdps. *arXiv preprint arXiv:2006.10814*, 2020b.

Agarwal, N., Chaudhuri, S., Jain, P., Nagaraj, D., and Netrapalli, P. Online target q-learning with reverse experience replay: Efficiently finding the optimal policy for linear mdps. *arXiv preprint arXiv:2110.08440*, 2021.

Al Marjani, A. and Proutiere, A. Adaptive sampling for best policy identification in markov decision processes. In *International Conference on Machine Learning*, pp. 7459–7468. PMLR, 2021.

Ayoub, A., Jia, Z., Szepesvari, C., Wang, M., and Yang, L. F. Model-based reinforcement learning with value-targeted regression. *arXiv preprint arXiv:2006.01107*, 2020.

Baird, L. Residual algorithms: Reinforcement learning with function approximation. In *International Conference on Machine Learning (ICML)*. 1995.

Bertsekas, D. P. and Tsitsiklis, J. N. *Neuro-dynamic programming*. Athena Scientific, 1996.

Beygelzimer, A., Langford, J., Li, L., Reyzin, L., and Schapire, R. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 19–26. JMLR Workshop and Conference Proceedings, 2011.

Bhandari, J., Russo, D., and Singal, R. A finite time analysis of temporal difference learning with linear function approximation. In *Conference on learning theory*, pp. 1691–1692. PMLR, 2018.

Cai, Q., Yang, Z., Lee, J., and Wang, Z. Neural temporal-difference learning converges to global optima. 2019.

Carvalho, D., Melo, F. S., and Santos, P. A new convergent variant of q-learning with linear function approximation. *Advances in Neural Information Processing Systems*, 33: 19412–19421, 2020.

Du, S. S., Kakade, S. M., Lee, J. D., Lovett, S., Mahajan, G., Sun, W., and Wang, R. Bilinear classes: A structural framework for provable generalization in rl. *arXiv preprint arXiv:2103.10897*, 2021.

Duan, Y., Wainwright, M. J., and Wang, M. Optimal value estimation using kernel-based temporal difference methods. Technical report, Princeton University, September 2021.

Even-Dar, E., Mansour, Y., and Bartlett, P. Learning rates for q-learning. *Journal of machine learning Research*, 5 (1), 2003.

Fan, J., Wang, Z., Xie, Y., and Yang, Z. A theoretical analysis of deep q-learning. In *Learning for Dynamics and Control*, pp. 486–489. PMLR, 2020.

Foster, D. J., Krishnamurthy, A., Simchi-Levi, D., and Xu, Y. Offline reinforcement learning: Fundamental barriers for value function approximation, 2021.

Frostig, R., Ge, R., Kakade, S. M., and Sidford, A. Competing with the empirical risk minimizer in a single pass. In *Conference on learning theory*, pp. 728–763. PMLR, 2015.

Golub, G. H. and Van Loan, C. F. *Matrix Computations*. JHU Press, 2012.

Gordon, G. J. Stable function approximation in dynamic programming. In *International Conference on Machine Learning (ICML)*, pp. 261–268. 1995.

He, J., Zhou, D., and Gu, Q. Logarithmic regret for reinforcement learning with linear function approximation. In *International Conference on Machine Learning*, pp. 4171–4180. PMLR, 2021.

Jaakkola, T., Jordan, M. I., and Singh, S. P. On the convergence of stochastic iterative dynamic programming algorithms. *Neural computation*, 6(6):1185–1201, 1994.

Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., and Schapire, R. E. Contextual decision processes with low Bellman rank are PAC-learnable. In Precup, D. and Teh, Y. W. (eds.), *International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1704–1713, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/jiang17c.html.

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

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, 2020.

Jin, C., Liu, Q., and Miryoosefi, S. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *arXiv preprint arXiv:2102.00815*, 2021.

Kearns, M. and Singh, S. Finite-sample convergence rates for q-learning and indirect algorithms. *Advances in neural information processing systems*, pp. 996–1002, 1999.

Khamaru, K., Xia, E., Wainwright, M. J., and Jordan, M. I. Instance-optimality in optimal value estimation: Adaptivity via variance-reduced Q-learning. Technical report, UC Berkeley, June 2021. Arxiv technical report 2106.14352.

Kong, D., Salakhutdinov, R., Wang, R., and Yang, L. F. Online sub-sampling for reinforcement learning with general function approximation, 2021.

Krishnamurthy, A., Agarwal, A., and Langford, J. Pac reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1840–1848, 2016.

Lakshminarayanan, C. and Szepesvari, C. Linear stochastic approximation: How far does constant step-size and iterate averaging go? In *International Conference on Artificial Intelligence and Statistics*, pp. 1347–1355. PMLR, 2018.

Li, C. J., Mou, W., Wainwright, M. J., and Jordan, M. I. Root-sgd: Sharp nonasymptotics and asymptotic efficiency in a single algorithm. *arXiv preprint arXiv:2008.12690*, 2020.

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

Li, Z., Xu, T., and Yu, Y. A note on target q-learning for solving finite mdps with a generative oracle. *arXiv preprint arXiv:2203.11489*, 2022.

Lin, L.-J. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3-4):293–321, 1992.

Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S., and Petrik, M. Finite-sample analysis of proximal gradient td algorithms. *arXiv preprint arXiv:2006.14364*, 2020.

Liu, S. and Su, H. Provably efficient kernelized q-learning. *arXiv preprint arXiv:2204.10349*, 2022.

Maurer, A. and Pontil, M. Empirical bernstein bounds and sample variance penalization. In *Conference on Learning Theory (COLT)*, 2009.

Mehta, N. Fast rates with high probability in exp-concave statistical learning. In *Artificial Intelligence and Statistics*, pp. 1085–1093. PMLR, 2017.

Mehta, P. and Meyn, S. Q-learning and pontryagin's minimum principle. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 3598–3605. IEEE, 2009.

Melo, F. S., Meyn, S. P., and Ribeiro, M. I. An analysis of reinforcement learning with function approximation. In *International Conference on Machine Learning (ICML)*, 2008.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533, 2015.

Modi, A., Jiang, N., Tewari, A., and Singh, S. Sample complexity of reinforcement learning using linearly combined model ensembles. In *International Conference on Artificial Intelligence and Statistics*, pp. 2010–2020. PMLR, 2020.

Modi, A., Chen, J., Krishnamurthy, A., Jiang, N., and Agarwal, A. Model-free representation learning and exploration in low-rank mdps. *arXiv preprint arXiv:2102.07035*, 2021.

Mou, W., Khamaru, K., Wainwright, M. J., Bartlett, P. L., and Jordan, M. I. Optimal variance-reduced stochastic approximation in Banach spaces. Technical report, UC Berkeley, January 2022.

Munos, R. and Szepesvári, C. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9 (May):815–857, 2008.

Perkins, T. J. and Pendrith, M. D. On the existence of fixed points for q-learning and sarsa in partially observable domains. In *ICML*, pp. 490–497, 2002.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1994. ISBN 0471619779.

Riedmiller, M. Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. In *European conference on machine learning*, pp. 317–328. Springer, 2005.

Santos, P. P., Melo, F. S., Sardinha, A., and Carvalho, D. S. Understanding the impact of data distribution on q-learning with function approximation. *arXiv preprint arXiv:2111.11758*, 2021.

Shi, L., Li, G., Wei, Y., Chen, Y., and Chi, Y. Pessimistic q-learning for offline reinforcement learning: Towards optimal sample complexity. *arXiv preprint arXiv:2202.13890*, 2022.

Shreve, S. E. and Bertsekas, D. P. Alternative theoretical frameworks for finite horizon discrete-time stochastic optimal control. *SIAM Journal on control and optimization*, 16(6):953–978, 1978.

Simchowitz, M. and Jamieson, K. Non-asymptotic gap-dependent regret bounds for tabular mdps. *arXiv preprint arXiv:1905.03814*, 2019.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Sun, W., Jiang, N., Krishnamurthy, A., Agarwal, A., and Langford, J. Model-based reinforcement learning in contextual decision processes. *arXiv preprint arXiv:1811.08540*, 2018.

Szepesvári, C. et al. The asymptotic convergence-rate of q-learning. *Advances in neural information processing systems*, pp. 1064–1070, 1998.

Tirinzoni, A., Pirotta, M., and Lazaric, A. A fully problem-dependent regret lower bound for finite-horizon mdps. *arXiv preprint arXiv:2106.13013*, 2021.

Tropp, J. A. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015. ISSN 1935-8237. doi: 10.1561/2200000048. URL http://dx.doi.org/10.1561/2200000048.

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

Tsitsiklis, J. N. and Van Roy, B. Analysis of temporal-diffference learning with function approximation. In *Advances in Neural Information Processing Systems (NIPS)*, 1997.

Wagenmaker, A., Chen, Y., Simchowitz, M., Du, S. S., and Jamieson, K. First-order regret in reinforcement learning with linear function approximation: A robust estimation approach, 2021a.

Wagenmaker, A., Simchowitz, M., and Jamieson, K. Beyond no regret: Instance-dependent pac reinforcement learning. *arXiv preprint arXiv:2108.02717*, 2021b.

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

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

Wang, R., Foster, D. P., and Kakade, S. M. What are the statistical limits of offline rl with linear function approximation? *arXiv preprint arXiv:2010.11895*, 2020a.

Wang, R., Salakhutdinov, R., and Yang, L. F. Provably efficient reinforcement learning with general value function approximation, 2020b.

Wang, Y., Wang, R., Du, S. S., and Krishnamurthy, A. Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136*, 2019.

Wang, Y., Wang, R., and Kakade, S. M. An exponential lower bound for linearly-realizable mdps with constant suboptimality gap. *arXiv preprint arXiv:2103.12690*, 2021.

Watkins, C. J. Learning from delayed rewards. 1989.

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

Weisz, G., Amortila, P., and Szepesvári, C. Exponential lower bounds for planning in mdps with linearly-realizable optimal action-value functions. *arXiv preprint arXiv:2010.01374*, 2020.

Weisz, G., Szepesvári, C., and György, A. Tensorplan and the few actions lower bound for planning in mdps under linear realizability of optimal value functions. *arXiv preprint arXiv:2110.02195*, 2021.

Xia, E., Khamaru, K., Wainwright, M. J., and Jordan, M. I. Instance-dependent confidence and early stopping in reinforcement learning. Technical report, UC Berkeley, January 2022.

Xie, T., Cheng, C.-A., Jiang, N., Mineiro, P., and Agarwal, A. Bellman-consistent pessimism for offline reinforcement learning. *arXiv preprint arXiv:2106.06926*, 2021.

Xu, H., Ma, T., and Du, S. S. Fine-grained gap-dependent bounds for tabular mdps via adaptive multi-step bootstrap. *arXiv preprint arXiv:2102.04692*, 2021a.

Xu, H., Zhan, X., and Zhu, X. Constraints penalized q-learning for safe offline reinforcement learning. *arXiv preprint arXiv:2107.09003*, 2021b.

Yan, Y., Li, G., Chen, Y., and Fan, J. The efficacy of pessimism in asynchronous q-learning. *arXiv preprint arXiv:2203.07368*, 2022.

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

Yang, L. F. and Wang, M. Reinforcement leaning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning (ICML)*, 2020.

Yin, M. and Wang, Y.-X. Towards instance-optimal offline reinforcement learning with pessimism. *arXiv preprint arXiv:2110.08695*, 2021.

Zanette, A. Exponential lower bounds for batch reinforcement learning: Batch rl can be exponentially harder than online rl. *arXiv preprint arXiv:2012.08005*, 2020.

Zanette, A. and Brunskill, E. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In *International Conference on Machine Learning (ICML)*, 2019. URL http://proceedings.mlr.press/v97/zanette19a.html.

Zanette, A., Brunskill, E., and J. Kochenderfer, M. Almost horizon-free structure-aware best policy identification with a generative model. In *Advances in Neural Information Processing Systems*, 2019.

Zanette, A., Brandfonbrener, D., Pirotta, M., and Lazaric, A. Frequentist regret bounds for randomized least-squares value iteration. In *AISTATS*, 2020a.

Zanette, A., Lazaric, A., Kochenderfer, M., and Brunskill, E. Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning (ICML)*, 2020b.

Zanette, A., Cheng, C.-A., and Agarwal, A. Cautiously optimistic policy optimization and exploration with linear function approximation. *arXiv preprint arXiv:2103.12923*, 2021a.

Zanette, A., Wainwright, M. J., and Brunskill, E. Provable benefits of actor-critic methods for offline reinforcement learning. *arXiv preprint arXiv:2108.08812*, 2021b.

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

A. Additional Material

A.1. Relation to past work

There is a long line of past work on Q-learning for tabular problems, with results in both the asymptotic settings (e.g., (Watkins & Dayan, 1992; Tsitsiklis, 1994; Jaakkola et al., 1994; Szepesvári et al., 1998)), as well as the non-asymptotic setting (e.g., (Kearns & Singh, 1999; Even-Dar et al., 2003; Wainwright, 2019a; Li et al., 2021)). Other work on Q-learning in tabular problems has derived regret bounds that are also near-optimal (Jin et al., 2018; Zhang et al., 2020).

It is well known that once function approximation is introduced, then the Q-learning algorithm may diverge (Baird, 1995). Such divergence does not occur in certain special cases, including when the dynamics are restricted to induce similar directions in feature space (Melo et al., 2008), or the function approximators are ℓ_∞-contractive in an appropriate sense (e.g. (Gordon, 1995)). Related results are presented in the papers (Tsitsiklis & Van Roy, 1997; Perkins & Pendrith, 2002; Mehta & Meyn, 2009; Liu et al., 2020; Bhandari et al., 2018; Lakshminarayanan & Szepesvari, 2018). In contrast, our analysis does not impose such conditions. We also note that there is some recent analysis of Q-learning with deep neural networks (Fan et al., 2020) that leverages connectionms to neural fitted Q-iteration (Riedmiller, 2005; Munos & Szepesvári, 2008); see also the papers (Cai et al., 2019; Carvalho et al., 2020).

Some of the algorithmic techniques used in this work—specifically, the use of target networks and experience replay—are believed to be essential to recent empirical successes in reinforcement learning. Experience replay was introduced by Lin (Lin, 1992), and popularized more widely by the influential paper (Mnih et al., 2013). To be clear, our replay mechanism differs in that it does not store past rewards and transitions; this fact is essential to maintaining low memory complexity. Our replay mechanism is related to the policy cover mechanism (Agarwal et al., 2020a; Zanette et al., 2021a), but differs in that it needs to store high performance policies, and it is not used as starting distribution for policies roll-outs. As for target networks, they have also been a core component of past empirical successes (Mnih et al., 2015).

We note that recent work by Agarwal et al. (Agarwal et al., 2021) also shows the importance of forms of experience replay, in establishing a result related to our Theorem 2. Our work shows that experience replay is not needed when the controller is stationary. Indeed, our primary contribution is in the exploration setting (cf. Theorem 1), whose literature we discuss next. A related and concurrent work in the exploration setting is (Liu & Su, 2022).

To the best of our knowledge, this paper constitutes the first analysis of an exploratory form of Q-learning combined with function approximation. It can be compared with the work of Jin et al. (Jin et al., 2020), who proved guarantees for exploration based on a form of least-squares value iteration (LSVI) with optimism for the class of low-rank MDPs. However, their algorithm has a space complexity that grows linearly with time, and the approximation error requirements are expressed via sup-norm (ℓ_∞) bounds. Better approximation error requirements with respect to a fixed comparator are given by policy gradient methods (Agarwal et al., 2020a; Zanette et al., 2021a), whose memory complexity still grows with the required accuracy[3]. Our work shows that attractive approximation error guarantees are not unique to policy gradient algorithms: temporal difference methods also inherit favorable—albeit different—guarantees. While this has recently been noted in the offline setting, such guarantees were enabled by a dataset generated from a stationary distribution (Xie et al., 2021), as opposed to a reactive controller (Zanette et al., 2021b), which is the standard case in the exploration setting.

Finally, to our knowledge none of algorithms discussed so far inherit instance-dependent regret bounds while being agnostic to the setting. The bulk of past instance-dependent results correspond to tabular problems (e.g., (Zanette & Brunskill, 2019; Zanette et al., 2019; Simchowitz & Jamieson, 2019; Yin & Wang, 2021; Tirinzoni et al., 2021; Al Marjani & Proutiere, 2021; Xu et al., 2021a; Wagenmaker et al., 2021b; Yang et al., 2021; Khamaru et al., 2021; Xia et al., 2022); a few exceptions include the logarithmic regret bounds given in the paper (He et al., 2021) and the recent paper (Wagenmaker et al., 2021a), as well as some partially instance-dependent results on kernel LSTD (Duan et al., 2021). Other studies related to Q-learning include the papers (Li et al., 2022; Yan et al., 2022; Shi et al., 2022; Santos et al., 2021; Xu et al., 2021b).

A.2. Future directions

Our work focused exclusively on linear approximations of Q-functions, but some of the underlying ideas are more generally applicable. One interesting direction is to extend our analysis to models with low Eluder dimension (Wang et al., 2020b;

[3] In the paper (Agarwal et al., 2020a), the policy cover grows linearly with the iteration count while the method (Zanette et al., 2021a) needs to store past trajectories to perform data reuse.

Kong et al., 2021), and to see whether the regret bound can be improved. Second, our definition of approximation error is very permissive, in that it only measures the expected prediction error. It would be interesting to understand whether or not there exist exploration algorithms based on least-square value iteration (without "policy replay") that inherit similar guarantees. Finally, this work establishes a partial form of instance-dependence, in that the results depend on the effective dimension. In the simpler tabular setting, the instance dependence of Q-learning has been studied through the lens of local minimax theory (Khamaru et al., 2021; Xia et al., 2022) to obtain completely sharp instance-dependent guarantees. It would be interesting to develop similarly sharp analyses in this more general setting with function approximation.

B. Proofs for the S^{3}Q-LEARNING algorithm

This section is devoted to proving the bounds on S^{3}Q-LEARNING stated in Theorem 2. At the same time, we also establish a related result, to be stated momentarily as Theorem 3. The proofs of both results share a very similar structure, following the same argument except for the way in which the violation of Assumption 1 or Assumption 2 is measured. More precisely, Theorem 2 measures misspecification according to ν, which is zero when Assumption 1 holds and the bonus is zero. On the other hand, Theorem 3 measures misspecification using ν defined according to Definition 2. This quantity is zero under the low-rank assumption (Assumption 2). Thus, both theorems can be proved within a common framework, with the only difference being the way in which ν is defined.

Let us now state the second result to be proved in this section.

Theorem 3. *Consider an MDP that is ν-misspecified w.r.t. the low rank assumption according to Definition 2; assume $b \geq 0$ pointwise. With the uncertainty function (16a), for any episode K, the* S^{3}Q-LEARNING *algorithm returns a solution $\widehat{Q}^\star$ with Bellman error $\mathcal{E}$ such that with probability at least $1 - \delta_{master}$:*

(a) The Bellman error function satisfies the pointwise bound

$$\min\{0, -\mathcal{U}_h + b_h\}(s,a) \leq \big(\mathcal{E}_h + \Delta_h^\pi\big)(s,a) \leq \big(\mathcal{U}_h + b_h\big)(s,a). \tag{18}$$

(b) For each $h \in [H]$, the greedy policy $\overline{\pi}_h(s) \overset{def}{=} \arg\max_a \widehat{Q}_h^\star(s,a)$ satisfies the bound

$$\mathbb{E}_{S\sim\rho}(\widehat{V}_1^\star - V_1^\star)(S) \leq \mathbb{E}_{S\sim\rho}(\widehat{V}_1^\star - V_1^{\overline{\pi}})(S) \leq \mathbb{E}_{\overline{\pi}} \sum_{h=1}^{H} (\mathcal{U}_h + b_h)(S_\tau, A_\tau) + \nu. \tag{19}$$

Proof : For each epoch, the argument can be broken into four steps.

- First, we show that for any level $\ell \in [H]$, the second-order update rule (13) produces the same iterates as least-squares regression would.
- Second, we show for any level $\ell \in [H]$, learned predictor $\widehat{Q}^\star$ uses at least $\sim \frac{1}{H}$ of the total data.
- Third, we bound the least-square prediction error under either Assumption 1 or Assumption 2. This analysis controls the error made by the algorithm at each level $\ell \in [H]$ during the epoch under consideration.
- The final step is to compute how the least-square errors propagate and accumulate through timesteps, thereby leading to the final performance bound in terms of the learned action value function $\widehat{Q}^\star$.

Notation: Let us summarize here some notation for convenient reference. We say that the algorithm *has completed learning* at level ℓ in epoch e if $n^\ell = 2^e$, i.e., when the loop over n^ℓ has terminated. We indicate with $n_e = 2^e$ the number of samples allocated in the epoch e.

Let $\{(s_i, a_i, r_i, s_i')\}_{i=1}^{n_e}$ be the samples acquired at level ℓ in epoch e. For a parameter vector θ and a next-state action-value function Q', define the (ℓ, e)-empirical loss as

$$\widehat{\mathcal{L}}_{\ell e}(\theta \mid\mid Q') \overset{def}{=} \sum_{i=1}^{n_e} \Big[\big\langle \phi_\ell(s_i, a_i),\, \theta\big\rangle - r_i - \max_{a'} Q'(s_i', a')\Big]^2 + \lambda_{\text{Reg}} \|\theta\|_2^2, \tag{20}$$

where $\lambda_{\text{Reg}} > 0$ is a given regularization parameter. With a minor overload of notation, Recalling the class of linear functions $\mathcal{Q}^{(\text{lin})}$ from equation (3a), we define

$$Q_{\min} = \underset{Q \in \mathcal{Q}^{(\text{lin})}}{\arg\min}\, \widehat{\mathcal{L}}_{\ell e}(Q \mid\mid Q') \tag{21}$$

if $Q_{\min}$ can be written as $Q_{\min} : (s,a) \mapsto \big\langle \phi(s,a),\, \theta_{\min}\big\rangle$, where $\theta_{\min} = \arg\min_{\|\theta\|_2 \leq 1} \widehat{\mathcal{L}}_{\ell e}(\theta \mid\mid Q')$.

B.1. Main argument

We now proceed to the core of the argument. When the algorithm terminates at the evaluation episode K, it returns the predictor $\widehat{Q}^\star$. For the rest of the proof, we let e_{tot} be the epoch in which $\widehat{Q}^\star$ was last updated upon termination of the algorithm. Moreover, our proof makes use of three auxiliary lemmas, which we begin by stating.

Lemma 1 (Equivalence with Least-Squares)**.** *Upon completion of level ℓ within epoch e, the* S^{3}Q-LEARNING *algorithm returns a parameter vector $\widehat{\theta}^{tar}_{\ell e}$ such that*

$$\widehat{\theta}^{tar}_{\ell e} = \arg\min_{\|\theta\|_2 \le 1} \left\{ \widehat{\mathcal{L}}_{\ell e}(\theta \mid\mid \widehat{Q}^{tar}_{\ell+1,e}) \right\}. \tag{22}$$

See Appendix B.2 for the proof of this claim.

We emphasize that the target network $\widehat{Q}^{tar}_{\ell+1,e}$ remains fixed throughout a given epoch. The lemma establishes that the second-order update rule produces the same solution as a batch least-square regression would.

Our next step is to lower bound the number of samples used to solve the regression problem:

Lemma 2 (Number of Samples)**.** *Upon termination in episode K, the algorithm returns a parameter sequence $\widehat{\theta}^\star = \{\widehat{\theta}^\star_\ell\}_{\ell=1}^H$ such that*

$$\widehat{\theta}^\star_\ell = \arg\min_{\|\theta\|_2 \le 1} \widehat{\mathcal{L}}_{\ell, e_{tot}}(\theta \mid\mid \widehat{Q}^\star_{\ell+1}) \qquad \textit{for each } \ell \in [H] \tag{23}$$

and moreover, the level ℓ regression problem uses at least $n^\ell \ge \frac{K}{4H}$ of the form $\{(s_i, a_i, r_i, s_i')\}_{i=1}^{n^\ell}$.

The above lemma states that nearly all the data collected are used. The proof can be found in Appendix B.3.

Given this equivalence to least-squares regression and the lower bound on the sample size, we can now leverage standard analysis of linear regression so as to bound the prediction error. Recall our definition of the error function

$$\mathcal{E}_{\ell e}(s,a) \overset{def}{=} \widehat{Q}^{tar}_{\ell e}(s,a) - \left(\mathcal{T}_\ell \widehat{Q}^{tar}_{\ell+1,e}\right)(s,a). \tag{24}$$

Lemma 3 (Least Square Error Bounds)**.** *The* S^{3}Q-LEARNING *procedure returns a predictor $\widehat{Q}^\star_\ell$ whose error $\mathcal{E}_\ell$ is sandwiched as*

$$\min\Big\{0, (-\mathcal{U}_\ell(s,a) + b_\ell(s,a))\Big\} - \Delta^\pi_\ell(s,a) \overset{(a)}{\le} \mathcal{E}_\ell(s,a) \overset{(b)}{\le} \mathcal{U}_\ell(s,a) + b_\ell(s,a) - \Delta^\pi_\ell(s,a) \tag{25}$$

with probability at least $1-\delta$.

See Appendix B.4 for the proof of this claim.

Lemma 3 allows us to quantify the empirical Bellman backup error and uses Assumption 1 *(Bellman closure)* or Assumption 2 *(Low-Rank)* depending on the setting (i.e., optimization vs exploration). In the zero-bonus setting ($b = 0$), the error term can be bounded symmetrically by the uncertainty function (see Eq. (25)), which is always positive. The $\min$ function on the left hand side arises due to "clipping" the $\widehat{Q}^{tar}_\ell$ values, so that adding a bigger bonus b_ℓ does not necessarily make $\widehat{Q}^{tar}_\ell$ (and its error function) more positive.

Our next step to establishing the bounds (17b) is to analyze how errors propagate. We begin with the rightmost inequality in equation (17b). Since $Q^\star$ is the optimal Q-function, we have the pointwise inequality

$$(\widehat{Q}^\star - Q^\star)(s,a) \le (\widehat{Q}^\star - Q^{\overline{\pi}})(s,a) \quad \text{for all } (s,a) \text{ pairs},$$

valid for any policy $\overline{\pi}$; in particular, this bound holds for the greedy policy $\overline{\pi}$ with respect to $\widehat{Q}^\star$. Moreover, for this greedy policy, we have

$$\widehat{Q}^\star_h = \mathcal{E}_h + \left(\mathcal{T}_h \widehat{Q}^\star_{h+1}\right) = \mathcal{E}_h + \left(\mathcal{T}^{\overline{\pi}}_h \widehat{Q}^\star_{h+1}\right), \qquad \text{for all } h \in [H].$$

Since $Q^{\overline{\pi}}$ satisfies the Bellman evaluation equations $Q^{\overline{\pi}}_h = \left(\mathcal{T}^{\overline{\pi}}_h Q^{\overline{\pi}}_{h+1}\right)$ for each $h \in [H]$, the claim now follows, as $\widehat{Q}^\star_h$ can be thought of as the action value function of $\overline{\pi}$ on an MDP with dynamics specified by $\mathcal{T}^{\overline{\pi}}$, and reward function consisting of a portion from $\mathcal{T}^{\overline{\pi}}$, along with an additional reward equal to $\mathcal{E}$.

The proof of the left inequality in equation (17b) is similar. In particular, we observe that

$$\widehat{Q}^\star_h = \mathcal{E}_h + \left(\mathcal{T}_h \widehat{Q}^\star_{h+1}\right) \geq \mathcal{E}_h + \left(\mathcal{T}^{\pi^\star}_h \widehat{Q}^\star_{h+1}\right), \qquad \text{for all } h \in [H].$$

Expanding the definition of error function along the trajectories identified by $\pi^\star$ concludes the proof of the claim.

B.2. Proof of Lemma 1 *(Equivalence with Least-Squares)*

Fix an epoch e and let the current level be ℓ. By construction, the target network for the next state value function $\widehat{Q}^{tar}_{\ell+1,e}$ has already been updated in that epoch. We observe that S^{3}Q-LEARNING is updating $\widehat{\theta}_h$ in a way equivalent to Algorithm 3 with $a_k = \phi_h$ and $b_k = r_h + \max_{a'} \widehat{Q}^{tar}_{h+1}(s_{h+1}, a')$ and $\mathcal{B} = \{x \mid \|x\|_2 \leq 1\}$.

Algorithm 3 STREAMING LEAST SQUARES

```
$\overline{\overline{\Sigma}}_1 = \lambda_{\text{Reg}} I_{d\times d}$
$x''_0 = 0$
for $k = 1, 2, \ldots$ do
    Receive $(a_k, b_k)$
    $x''_k = x''_{k-1} + \frac{\Sigma_k^{-1} a_k \left(b_k - a_k^\top x''_{k-1}\right)}{1 + a_k^\top \Sigma_k^{-1} a_k}$
    $\overline{\overline{\Sigma}}_{k+1}^{-1} = \overline{\overline{\Sigma}}_k^{-1} - \frac{\overline{\overline{\Sigma}}_k^{-1} a_k a_k^\top \overline{\overline{\Sigma}}_k^{-1}}{1 + a_k^\top \overline{\overline{\Sigma}}_k^{-1} a_k}$
end for
return $\arg\min_{\|x\|_2 \leq 1} \|x - x''_K\|^2_{\overline{\overline{\Sigma}}_{K+1}}$
```

Thus, in order to prove Lemma 1, it suffices to show that Algorithm 3 finds the empirical risk minimizer. In particular, we claim that given any sequence of tuples $(a_1, b_1), \ldots, (a_k, b_k)$, then upon termination, Algorithm 3 returns the constrained minimizer

$$\arg\min_{\|x\|_2 \leq 1} \left\{ \sum_{k=1}^{K} \left(x^\top a_k - b_k\right)^2 + \lambda_{\text{Reg}} \|x\|_2^2 \right\}. \tag{26}$$

In order to prove this claim, we introduce some helpful notation. With the initialization $A_0 \overset{def}{=} \sqrt{\lambda_{\text{Reg}}} I_{d\times d}$ and $B_0 \overset{def}{=} 0$, define the recursions

$$A_k \overset{def}{=} \begin{bmatrix} A_{k-1} \\ a_k^\top \end{bmatrix}, \qquad B_k \overset{def}{=} \begin{bmatrix} B_{k-1} \\ b_k \end{bmatrix} \tag{27}$$

The associated solution to the normal equations is given by $x'_k \overset{def}{=} (A_k^\top A_k)^{-1} A_k^\top B_k$. With these definition, we then have the equivalences

$$x'_k \overset{\text{(i)}}{=} \arg\min_x \|A_k x - B_k\|_2^2 \overset{\text{(ii)}}{=} \sum_{i=1}^{k} \left(x^\top a_i - b_i\right)^2 + \lambda_{\text{Reg}} \|x\|_2^2, \tag{28}$$

where step (i) follows by definition; and step (ii) follows from how the dataset (A_0, b_0) was constructed, in particular including the pair $\{(\sqrt{\lambda_{\text{Reg}}} e_1, 0), \ldots, (\sqrt{\lambda_{\text{Reg}}} e_d, 0)\}$ prior to the k samples $\{(a_i, b_i)\}_{i=1}^k$.

Now we proceed by induction on the index k. For the base case $k = 0$, observe that $x'_0 = x''_0$ and $A_0^\top A_0 = \lambda_{\text{Reg}} I$ by the given initialization. Turning to the induction step, let us suppose that the equalities

$$x'_{k-1} = x''_{k-1}, \quad \text{and} \quad A_{k-1}^\top A_{k-1} = \overline{\overline{\Sigma}}_k \tag{29}$$

hold for a certain $k \geq 1$. We can write the next iterate x'_k as

$$x'_k = (A_k^\top A_k)^{-1} A_k^\top B_k = \left(A_{k-1}^\top A_{k-1} + a_k a_k^\top\right)^{-1} \begin{bmatrix} A_{k-1}^\top & a_k \end{bmatrix} \begin{bmatrix} B_{k-1} \\ b_k \end{bmatrix}$$

$$\overset{\text{(iii)}}{=} \left[\left(A_{k-1}^\top A_{k-1}\right)^{-1} - \frac{\left(A_{k-1}^\top A_{k-1}\right)^{-1} a_k a_k^\top \left(A_{k-1}^\top A_{k-1}\right)^{-1}}{1 + a_k^\top \left(A_{k-1}^\top A_{k-1}\right)^{-1} a_k} \right] \left(A_{k-1}^\top B_{k-1} + a_k b_k\right),$$

where step (iii) follows from the Sherman Morrison rank-one matrix inversion formula (e.g., (Golub & Van Loan, 2012)). Recalling that

$$x'_{k-1} = \left(A_{k-1}^\top A_{k-1}\right)^{-1} A_{k-1}^\top B_{k-1},$$

we find that

$$\begin{aligned} x'_k &= x'_{k-1} + \left(A_{k-1}^\top A_{k-1}\right)^{-1} a_k b_k - \frac{\left(A_{k-1}^\top A_{k-1}\right)^{-1} a_k a_k^\top \left(A_{k-1}^\top A_{k-1}\right)^{-1}}{1 + a_k^\top \left(A_{k-1}^\top A_{k-1}\right)^{-1} a_k} \left(A_{k-1}^\top B_{k-1} + a_k b_k\right) \\ &= x'_{k-1} + \frac{\left(A_{k-1}^\top A_{k-1}\right)^{-1} a_k b_k - \left(A_{k-1}^\top A_{k-1}\right)^{-1} a_k a_k^\top \overbrace{\left(A_{k-1}^\top A_{k-1}\right)^{-1} A_{k-1}^\top B_{k-1}}^{x'_{k-1}}}{1 + a_k^\top \left(A_{k-1}^\top A_{k-1}\right)^{-1} a_k} \\ &= x'_{k-1} + \frac{\left(A_{k-1}^\top A_{k-1}\right)^{-1} a_k \left(b_k - a_k^\top x'_{k-1}\right)}{1 + a_k^\top \left(A_{k-1}^\top A_{k-1}\right)^{-1} a_k}. \end{aligned}$$

Since $x'_{k-1} = x''_{k-1}$ by the induction hypothesis, it follows that $x'_k = x''_k$ as the above display matches Line 5 of Algorithm 3. Applying the Sherman-Morrison rank one update, we find that

$$(\overline{\overline{\Sigma}}_{k+1})^{-1} = (A_k^\top A_k)^{-1}. \tag{30}$$

Thus, we have established that the equalities (29) hold for every k. The proof is concluded upon noticing that x''_k is the unconstrained solution to the loss in (26), and the projection step in the final line of the algorithm is thus equivalent to solving (26) with the constraint $x \in \mathcal{B}$.

B.3. Proof of Lemma 2 *(Number of Samples)*

In every epoch the algorithm updates the target networks in the order $\widehat{Q}^{tar}_{H,e}, \dots, \widehat{Q}^{tar}_{1,e}$. Since the algorithm returns $\widehat{Q}^\star_\ell = \widehat{Q}^{tar}_{\ell,e_{tot}}$ for each $\ell \in [H]$, the statement (23) follows by construction of the algorithm and Lemma 1.

It remains to lower bound the number of samples involved in the computation of $\widehat{Q}^\star_\ell$. By construction, every epoch e uses exactly $2^e H$ trajectories. Let m denote the number of trajectories in the current (unfinished) epoch $e_{tot} + 1$ when we evaluate the algorithm (at the stopping time in the episode K). We must have

$$K = H\left(2^1 + 2^2 + \dots + 2^{e_{tot}} + m\right) \leq H\left(2^{e_{tot}+1} + 2^{e_{tot}+1}\right) = 4H2^{e_{tot}}. \tag{31}$$

Since the algorithm in epoch e_{tot} has sampled $H2^{e_{tot}}$ trajectories, using the above relation we deduce that it must have used $H2^{e_{tot}} \geq \frac{K}{4}$ total episodes, meaning at least $n^\ell \geq \frac{K}{4H}$ in every level to solve the regression problem (23), as stated.

B.4. Proof of Lemma 3 *(Least Square Error Bounds)*

Let $\{(s_i, a_i, r_i, s'_i)\}_{i=1}^{n_e}$ be the sequence of n_e states, actions, rewards, successor states acquired while learning level ℓ in epoch e. We drop the epoch index $e = e_{tot}$ as this is fixed through the proof.

Within an epoch, S^{3}Q-LEARNING updates the target networks in the order $\widehat{Q}^{tar}_H, \widehat{Q}^{tar}_{H-1}, \dots, \widehat{Q}^{tar}_2, \widehat{Q}^{tar}_1$. Thus, when the algorithm updates $\widehat{Q}^{tar}_\ell$, it must use $\widehat{Q}^{tar}_{\ell+1}$, which has already been updated in that epoch, to compute the backup in Line 12. Notice that the next-timestep target $\widehat{Q}^{tar}_{\ell+1}$ stays fixed while learning at level ℓ. Observe that regardless of the choice of bonus (so in either the optimization or exploration setting), we have $\|\widehat{Q}^{tar}_{\ell+1}\|_\infty \leq 1$ by construction.

We introduce the shorthand $\Delta_i^\pi \overset{def}{=} \left(\mathcal{T}_\ell \widehat{Q}_{\ell+1}^{tar}\right)(s_i, a_i) - \langle \phi_\ell(s_i, a_i), \theta^{\pi, \widehat{Q}_{\ell+1}^{tar}, \ell} \rangle$ for the comparator error evaluated at (s_i, a_i). With this shorthand, we can write

$$r_i + \max_{a'} \widehat{Q}_{\ell+1}^{tar}(s_i', a') = \left(\mathcal{T}_\ell \widehat{Q}_{\ell+1}^{tar}\right)(s_i, a_i) + \eta_i = \langle \phi_\ell(s_i, a_i), \theta^{\pi, \widehat{Q}_{\ell+1}^{tar}, \ell} \rangle + \Delta_i^\pi + \eta_i, \tag{32}$$

where $\eta_i \overset{def}{=} r_i + \max_{a'} \widehat{Q}_{\ell+1}^{tar}(s_i', a') - \left(\mathcal{T}_\ell \widehat{Q}_{\ell+1}^{tar}\right)(s_i, a_i)$ is the Bellman noise. Note that conditioned on (s_i, a_i), the random variable on the left hand side is bounded in $[-1, +1]$.

Now, to conclude we use the following high-probability error bound on a perturbed least-squares estimator. Given a joint distribution μ over pairs (X, Y), define the constrained least-squares estimate

$$\theta^\star \overset{def}{=} \min_{\|\theta\|_2 \le 1} \mathbb{E}_{(X,Y)} \left(\langle X,\, \theta \rangle - Y \right)^2. \tag{33a}$$

Given n i.i.d. samples $(x_i, y_i) \sim \mu$, we define the empirical version of this estimator

$$\widehat{\theta} \overset{def}{=} \min_{\|\theta\|_2 \le 1} \frac{1}{n} \sum_{i=1}^n \left(\langle x_i,\, \theta \rangle - y_i \right)^2. \tag{33b}$$

The following result bounds the difference between the empirical and population estimates:

Lemma 4 (Convergence to Population Minimizer)**.** *The empirical estimate* (33b) *satisfies the bound*

$$\|\widehat{\theta} - \theta^\star\|_{\left(n \mathbb{E}_\mu x x^\top + \lambda I\right)} \le c \left\{ \sqrt{d \log \frac{dn}{\delta}} + \sqrt{\lambda} \right\} \tag{34}$$

with probability at least $1 - \delta$*.*

See Appendix B.5 for the proof of this claim.

After redefining δ and collecting probabilities, Cauchy-Schwartz now ensures with probability $1 - \delta$ that

$$\left| \langle \phi_\ell(s, a),\, \widehat{\theta}_\ell^{tar} - \theta^{\pi, \widehat{\theta}_{\ell+1}^{tar}, \ell} \rangle \right| \le \|\phi_\ell(s, a)\|_{\left(\overline{\Sigma}_\ell(n^\star)\right)^{-1}} \|\widehat{\theta}_\ell^{tar} - \theta^{\pi, \widehat{\theta}_{\ell+1}^{tar}, \ell}\|_{\overline{\Sigma}_\ell(n^\star)} \le \underbrace{\alpha(n^\star, \delta^\star) \|\phi_\ell(s, a)\|_{\left(\overline{\Sigma}_\ell(n^\star)\right)^{-1}}}_{\equiv \mathcal{U}_\ell(s,a)}, \tag{35}$$

where we have used the definition (16a) for the uncertainty parameter with the number of samples given by Lemma 2 *(Number of Samples)* together with a union bound over the horizon and the random epoch at evaluation time.

We now use the bound (35) to prove the two inequalities in equation (25).

Proof of inequality (25)**(b):** We begin with the upper bound. Due to the clipping step, we are guaranteed to have the upper bound $\widehat{Q}_\ell^{tar}(s, a) \le \langle \phi_\ell(s, a),\, \widehat{\theta}_\ell^{tar} \rangle + b_\ell(s, a)$, and hence

$$\widehat{Q}_\ell^{tar}(s, a) - \langle \phi_\ell(s, a),\, \theta^{\pi, \widehat{\theta}_{\ell+1}^{tar}, \ell} \rangle = \langle \phi_\ell(s, a),\, \widehat{\theta}_\ell^{tar} - \theta^{\pi, \widehat{\theta}_{\ell+1}^{tar}, \ell} \rangle + b_\ell(s, a) \overset{(i)}{\le} \mathcal{U}_\ell(s, a) + b_\ell(s, a),$$

where step (i) follows from our earlier inequality (35). Thus, we have established the bound (25)(b) once we recall the definition of transfer error Δ_ℓ^π.

Proof of the lower bound (25)**(a):** We now turn to the lower bound. By construction, we have $\|\widehat{Q}_\ell^{tar}\|_\infty \le 1$, so that we can write

$$\widehat{Q}_\ell^{tar}(s, a) = \min \left\{ 1, \langle \phi_\ell(s, a),\, \widehat{\theta}_\ell^{tar} \rangle + b_\ell(s, a) \right\}.$$

Consequently, by adding and subtracting the term $\left\langle \phi_\ell(s,a),\, \theta^{\pi,\widehat{\theta}^{tar}_{\ell+1},\ell} \right\rangle$, we have

$$\begin{aligned}\widehat{Q}^{tar}_\ell(s,a) &= \min\left\{1, \left\langle \phi_\ell(s,a),\, \widehat{\theta}^{tar}_\ell - \theta^{\pi,\widehat{\theta}^{tar}_{\ell+1},\ell} \right\rangle + b_\ell(s,a) + \left\langle \phi_\ell(s,a),\, \theta^{\pi,\widehat{\theta}^{tar}_{\ell+1},\ell} \right\rangle \right\} \\ &\overset{(i)}{\geq} \min\left\{1, -\mathcal{U}_\ell(s,a) + b_\ell(s,a) + \left\langle \phi_\ell(s,a),\, \theta^{\pi,\widehat{\theta}^{tar}_{\ell+1},\ell} \right\rangle \right\} \\ &\overset{(ii)}{\geq} \min\left\{0, -\mathcal{U}_\ell(s,a) + b_\ell(s,a) \right\} + \left\langle \phi_\ell(s,a),\, \theta^{\pi,\widehat{\theta}^{tar}_{\ell+1},\ell} \right\rangle.\end{aligned}$$

where step (i) uses our earlier bound (35), and step (ii) follows since $\left\langle \phi_\ell(s,a),\, \theta^{\pi,\widehat{\theta}^{tar}_{\ell+1},\ell} \right\rangle \leq 1$. In this way, we have shown that

$$\widehat{Q}^{tar}_\ell(s,a) - (\mathcal{T}_\ell \widehat{Q}^{tar}_{\ell+1})(s,a) \geq \min\left\{0, -\mathcal{U}_\ell(s,a) + b_\ell(s,a)\right\} - \Delta^\pi_\ell(s,a),$$

as claimed in equation (25)(a).

B.5. Proof of Lemma 4

Our proof makes use of known bounds on the excess risk in a linear regression problem. In particular, consider a regression problem based on covariate vectors $\phi \in \mathbb{R}^d$ and responses $y \in \mathbb{R}$ that satisfy the bounds $\|\phi\|_2 \leq 1$ and $|y| \leq y_{max}$.

With the shorthand $z = (\phi, y)$, define the least-squares loss $\mathcal{L}_w(z) = \frac{1}{2}(y - \langle \phi,\, w \rangle)^2$. For some distribution $\mathbb{P}$ over $\mathbb{R}^d \times \mathbb{R}$, define the constrained population and empirical minimizers

$$w^\star \overset{def}{=} \arg \min_{\|w\|_2 \leq B} \mathbb{E}_{Z \sim \mathbb{P}} \mathcal{L}_w(Z), \quad \text{and} \quad \widehat{w} \overset{def}{=} \arg \min_{\|w\|_2 \leq B} \frac{1}{n} \sum_{i=1}^n \mathcal{L}_w(Z_i)$$

where $\{Z_i\}_{i=1}^n$ are drawn i.i.d. according to $\mathbb{P}$.

We claim that the excess risk associated with the constrained least-squares estimate $\widehat{w}$ can be bounded as

$$\mathbb{E}_{Z \sim \mathbb{P}}[\mathcal{L}_{\widehat{w}}(Z) - \mathcal{L}_{w^\star}(Z)] \leq \frac{1}{n}\left\{32(B + y_{max})^2 \times \left[d \log\left(32Bn(B + y_{max})\right) + \log\left(\tfrac{1}{\delta'}\right)\right] + 1\right\}, \tag{36}$$

with probability at least $1 - \delta'$. This bound follows as a consequence of a result due to Mehta (Mehta, 2017). In particular, the maximum value the loss can take is $L^2_{max} = (B + y_{max})^2$. Applying Theorem 1 in the paper (Mehta, 2017) to the least-squares objective, which is $1/(4L^2_{max})$-exp-concave, yields the claim.

To conclude, the proof of Lemma 4 follows by combining the bound with Lemma 16 *(Excess Risk with Regularization)*.

C. Proof of Theorem 1

This section is devoted to the proof of the performance bound on S^{4}Q-LEARNING stated in Theorem 1. At a high level, the proof consists of three steps. First, we decompose the regret into a sum of partial regrets incurred in each phase. Second, we show that the exploration bonus correctly quantify the uncertainty; this allows the algorithm to ensure optimism. Finally, we use an elliptic potential argument and a bound on the number of phases to conclude the proof.

Notation: Letting p be the current phase, we use $Q^{(p)}$ and $V^{(p)}$ (respectively) to denote the Q-action-value and value functions returned by S^{3}Q-LEARNING, (cf. Line 6) and $\theta^{(p)}$ to denote the associated parameter of its linear representation. We let $\pi^{(p)}$ be the policy extracted in Line 7 of S^{4}Q-LEARNING after termination of S^{3}Q-LEARNING. Let $n^{(p)}$ be the (random) number of times that the policy is executed in phase p between Line 9 and Line 16 of S^{4}Q-LEARNING (Algorithm 2). Denote with p_{tot} the total number of phases, including the one that is still in progress at the evaluation episode K. We let $b^{(p)}$ denote the bonus (17) created at the end of phase $p-1$ in Line 17 of Algorithm 2; the bonus will be actively used in phase p. We use the shorthand $\Delta_h^{(p)}$ with the same meaning as Eq. (16c) where the policy π is the one used in phase p within S^{3}Q-LEARNING.

C.1. Main argument

We begin by decomposing the regret of S^{4}Q-LEARNING into the sums of partial regrets generated in each phase $p = 1, \dots, p_{tot}$. The partial regret REGRET$^{(p)}$ in phase p corresponds to the regret incurred by playing all policies in that phase. Policy rollouts that generate regret are performed in one of two places: (i) in the call to S^{3}Q-LEARNING (see Line 6 of S^{4}Q-LEARNING), or; (ii) by S^{4}Q-LEARNING between Line 9 and Line 16. We can write

$$\text{REGRET}^{(p)} \overset{def}{=} \sum_{\substack{\pi \text{ played in} \\ \text{the call to} \\ \text{S}^3\text{Q-LEARNING} \\ \text{in phase } p}} \mathbb{E}_{S_1 \sim \rho} \left(V_1^\star - V_1^\pi\right)(S_1) + \sum_{\substack{\pi \text{ played in} \\ \text{the main loop of} \\ \text{S}^4\text{Q-LEARNING} \\ \text{in phase } p}} \mathbb{E}_{S_1 \sim \rho} \left(V_1^\star - V_1^\pi\right)(S_1). \tag{37}$$

The following lemma leverages the mechanics of the algorithm to upper bounds the total regret up to episode K by expressing it as a sum of the regrets generated only by the main loop of S^{4}Q-LEARNING (so excluding the call to S^{3}Q-LEARNING). What makes this possible is that S^{3}Q-LEARNING only plays the policies from the policy replay memory $\Pi^{(p)}$, which have already been played by the controller (S^{4}Q-LEARNING). Summing over the phases and accounting for possible statistical deviations (due to sampling from the policy mixture) yields the following claim:

Lemma 5 (Phased Regret)**.** *With probability at least* $1 - \delta$*, uniformly over all* K*, we have the upper bound* REGRET$(K) \overset{def}{=} \sum_{p=1}^{p_{tot}}$ REGRET$^{(p)} \leq T_1 + T_2$*, where*

$$T_1 \overset{def}{=} H \Big\{ \sum_{p=1}^{p_{tot}} \sum_{j=1}^{p-1} n^{(j)} \times \mathbb{E}_{S \sim \rho} \big[V_1^\star(S) - V_1^{\pi^{(j)}}(S) \big] \Big\} + H \log \left(\tfrac{p_{tot}}{\delta} \right) \quad \textit{and} \tag{38a}$$

$$T_2 \overset{def}{=} \sum_{p=1}^{p_{tot}} n^{(p)} \times \mathbb{E}_{S \sim \rho} \big[V_1^\star(S) - V_1^{\pi^{(p)}}(S) \big]. \tag{38b}$$

See Appendix C.2 for the proof of this claim.

Note that T_1 corresponds to the regret associated with S^{3}Q-LEARNING, whereas T_2 is associated with S^{4}Q-LEARNING. To be clear, the regret to S^{4}Q-LEARNING excludes the regret of the policies that are rolled out within the S^{3}Q-LEARNING subroutine. Now notice that for some constant $c \in \mathbb{R}$ we can write the total number of episodes as

$$K = \underbrace{cH \sum_{p=1}^{p_{tot}} \sum_{j=1}^{p-1} n^{(j)}}_{\substack{\text{Total trajectories by} \\ \text{S}^3\text{Q-LEARNING}}} + \underbrace{\sum_{p=1}^{p_{tot}} n^{(p)}}_{\substack{\text{Total trajectories by} \\ \text{S}^4\text{Q-LEARNING}}} .$$

Applying the Cauchy–Schwarz inequality yields the bound REGRET$(K) \leq T_3 \times \sqrt{T_4}$, where

$$T_3 \overset{def}{=} \sqrt{cH \sum_{p=1}^{p_{tot}} \sum_{j=1}^{p-1} n^{(j)}, \sum_{p=1}^{p_{tot}} n^{(p)},} \quad \text{and}$$

$$T_4 \overset{def}{=} cH \sum_{p=1}^{p_{tot}} \sum_{j=1}^{p-1} n^{(j)} \Big[\mathbb{E}_{S \sim \rho} \Big(V_1^\star - V_1^{\pi^{(j)}} \Big)(S) \Big]^2 + \sum_{p=1}^{p_{tot}} n^{(p)} \Big[\mathbb{E}_{S \sim \rho} \Big(V_1^\star - V_1^{\pi^{(p)}} \Big)(S) \Big]^2 .$$

Thus, a standard $\sqrt{K}$ regret bound can be obtained as soon as the term T_4 is bounded. The next step is then to transform T_4 into an estimation problem through optimism.

Recall from equation (9). that we introduce optimism via an exploration bonus of the form

$$b_h^{(p)}(s, a) \overset{def}{=} \alpha_h^{(p)} \|\phi_h(s, a)\|_{(\Sigma_h^{(p)})^{-1}} \tag{39}$$

for all state-action pairs. This bonus is created in Line 17, and passed to S^{3}Q-LEARNING. The uncertainty parameter $\alpha^{(p)}$ to be used in phase p is defined in Eq. (9). To be clear, the covariance matrix used to construct the bonus is the one in the current phase $\Sigma^{(p)}$. In order to proceed, we must relate the bonus to the uncertainty function. Let us define the reference covariance

$$\overline{\Sigma}_h^{(p)} \overset{def}{=} \sum_{j=1}^{p-1} n^{(j)} \mathbb{E}_{\phi_h \sim \pi^{(j)}}[\phi_h \phi_h^\top] + \lambda_{\text{Reg}} I.$$

We also recall our earlier definition (15) of the uncertainty parameter

$$\alpha_h^{(p)} = c\Big\{\sqrt{d \log\big(\tfrac{dpn^{(1:p)}}{\delta}\big)} + \sqrt{\lambda_{\text{Reg}}}\Big\}. \tag{40}$$

On our way to prove optimism, the next proposition highlights the relation between the bonus and the uncertainty function.

Lemma 6 (Bonus Bound)**.** *Set $\lambda_{Reg} = \Theta(\log \frac{dpn^{(1:p)}}{\delta})$. There exists a large enough $c \in \mathbb{R}$ in Eq. (40) such that with probability at least $1-\delta$ jointly for all episodes K we have the pointwise bound*

$$\mathcal{U}_h^{(p)} \le b_h^{(p)} \le c_o \mathcal{U}_h^{(p)} \tag{41}$$

for some constant $c_o \in \mathbb{R}$.

The lemma is proved in Appendix C.3, and it allows us to claim that the algorithm is optimistic (using the left inequality above) but without using a bonus that is too large (right inequality above) which would create too much regret. We verify such optimistic claim in Appendix C.4. Recall the definition of transfer error in Eq. (5) and the comparator error in Eq. (16c) where $\widehat{Q}_{h+1}^{tar} = Q_{h+1}^{(p)}$ is the network returned by S^{3}Q-LEARNING.

Lemma 7 (Near-Optimism)**.** *Suppose that the event in Lemma 6 holds jointly for all episodes K. Then optimism holds in the following sense:*

$$Q_h^{(p)}(s,a) \ge Q_h^\star(s,a) - \sum_{\tau=h}^{H} \mathbb{E}_{(S'_\tau, A'_\tau)\sim\pi^\star|(s,a)} \Delta_\tau^{(p)}(S'_\tau, A'_\tau), \qquad \forall (s,a) \tag{42a}$$

As a consequence, under the same event, we have

$$\mathbb{E}_{S\sim\rho} V_1^{(p)}(S) \ge \mathbb{E}_{S\sim\rho} V^\star(S) - \nu. \tag{42b}$$

From the optimism in the procedure, at any phase j, we have the bound

$$0 \le \mathbb{E}_{S\sim\rho}\left(V_1^\star - V_1^{\pi^{(j)}}\right)(S) \le \mathbb{E}_{S\sim\rho}\left(V_1^{(j)} - V_1^{\pi^{(j)}}\right)(S) + \nu,$$

and hence

$$T_4 \lesssim T_{4a} + T_{4b} + \underbrace{\left(cH \sum_{p=1}^{p_{tot}} \sum_{j=1}^{p-1} n^{(j)} + \sum_{p=1}^{p_{tot}} n^{(p)}\right)}_{=K} \nu^2$$

where

$$T_{4a} \overset{def}{=} cH \sum_{p=1}^{p_{tot}} \sum_{j=1}^{p-1} n^{(j)} \Big[\mathbb{E}_{S\sim\rho}\big(V_1^{(j)} - V_1^{\pi^{(j)}}\big)(S)\Big]^2 m \quad \text{and} \quad T_{4b} \overset{def}{=} \sum_{p=1}^{p_{tot}} n^{(p)} \Big[\mathbb{E}_{S\sim\rho}\big(V_1^{(p)} - V_1^{\pi^{(p)}}\big)(S)\Big]^2.$$

In turn bounding T_{4a} and T_{4b}, we find that

$$T_4 \lesssim H p_{tot} \sum_{j=1}^{p} n^{(j)} \Big[\mathbb{E}_{S\sim\rho}\left(V_1^{(j)} - V_1^{\pi^{(j)}}\right)(s)\Big]^2 + K\nu^2.$$

The remainder of the proof is devoted to deriving a high probability bound on the first term of the above display. Using the error bounds on S^{3}Q-LEARNING from Theorem 3, we can write

$$\begin{aligned} T_4 &\lesssim Hp_{tot}\sum_{j=1}^{p} n^{(j)}\Big[\sum_{h=1}^{H}\mathbb{E}_{(S_h,A_h)\sim\pi^{(j)}}(\mathcal{U}_h^{(j)}+b_h^{(j)}-\Delta_h^{(j)})(s_h,a_h)\Big]^2+K\nu^2 \\ &\lesssim Hp_{tot}\sum_{j=1}^{p} n^{(j)}\Big[\sum_{h=1}^{H}\mathbb{E}_{(S_h,A_h)\sim\pi^{(j)}}(\mathcal{U}_h^{(j)})(s_h,a_h)\Big]^2+K\nu^2. \end{aligned}$$

The second step follows by bringing ν outside the square and by bounding the bonus by using Lemma 6 *(Bonus Bound)*.

Putting together the pieces, we have

$$\begin{aligned} T_4 &\overset{(i)}{\lesssim} H^2p_{tot}\sum_{j=1}^{p} n^{(j)}\sum_{h=1}^{H}\Big[\mathbb{E}_{(S_h,A_h)\sim\pi^{(j)}}\mathcal{U}_h^{(j)}(s_h,a_h)\Big]^2+K\nu^2 \\ &\overset{(ii)}{\leq} H^2p_{tot}\sum_{h=1}^{H}\sum_{j=1}^{p} n^{(j)}\mathbb{E}_{(S_h,A_h)\sim\pi^{(j)}}\Big[\mathcal{U}_h^{(j)}(s_h,a_h)\Big]^2+K\nu^2 \\ &= H^2p_{tot}\sum_{h=1}^{H}\sum_{j=1}^{p} n^{(j)}\mathbb{E}_{(S_h,A_h)\sim\pi^{(j)}}\Big[\alpha_h^{(j)}\|\phi_h(s_h,a_h)\|_{(\overline{\Sigma}_h^{(j)})^{-1}}\Big]^2+K\nu^2, \end{aligned}$$

where step (i) follows from the Cauchy–Schwarz inequality, and step (ii) follows from Jensen's inequality. Notice that that we have the ordering $\alpha_h^{(1)}\leq\cdots\leq\alpha_h^{(p)}\leq\cdots\leq\alpha_h^{(p_{tot})}$, i.e., the sequence of reference parameter uncertainty must be non-decreasing. Consequently, we find that

$$T_4 \lesssim H^2p_{tot}\sum_{h=1}^{H}\left(\alpha_h^{(p_{tot})}\right)^2\sum_{j=1}^{p} n^{(j)}\mathbb{E}_{(S_h,A_h)\sim\pi^{(j)}}\|\phi_h(s_h,a_h)\|^2_{(\overline{\Sigma}_h^{(j)})^{-1}}+K\nu^2. \tag{43}$$

We now proceed to bound the sum of the quadratic terms, a quantity that arises in linear bandit analysis. In order to do so, we need to define the triggering value used in Line 16 of Algorithm 2.

$$L_{\text{trig}} = \Theta\left(\log\frac{np}{\delta}\right) \tag{44}$$

where $n\leq K$ is the number of times the condition has been checked in phase p. We obtain the following lemma which is proved in Appendix C.5.

Lemma 8 (Elliptic Potential)**.** *Assume* $\lambda_{Reg}=\Theta(\log\frac{dpn^{(1:p)}}{\delta})\geq 1$. *There exists a setting* L_{trig} *as defined in Eq.* (44) *such that with probability at least* $1-\delta$ *we have*

$$\sum_{j=1}^{p} n^{(j)}\mathbb{E}_{(S_h,A_h)\sim\pi^{(j)}}\|\phi_h(s_h,a_h)\|^2_{(\overline{\Sigma}_h^{(j)})^{-1}}\leq 8\left(L_{trig}+1\right)\widetilde{d}_h. \tag{45}$$

Continuing the bound (43), recalling the definition (40) and applying Cauchy-Schwartz gives us

$$\begin{aligned} T_4 &\lesssim H^2p_{tot}\left(\sum_{h=1}^{H}\left(\alpha_h^{(p_{tot})}\right)^2\widetilde{d}_h\right)\times 8\left(L_{\text{trig}}+1\right)+K\nu^2 \\ &\lesssim H^2p_{tot}\sum_{h=1}^{H}\left(d_h+\log\frac{dp_{tot}K}{\delta}\right)\widetilde{d}_h\times 8\left(L_{\text{trig}}+1\right)+K\nu^2. \end{aligned}$$

To conclude, it remains to bound the total number p_{tot} of phases; we state the bound here and prove it in Appendix C.6.

Lemma 9 (Number of Phases). *Under the conditions of Lemma 8, the total number of phases up to episode K is upper bounded as*

$$p_{tot} \leq \sum_{h=1}^{H} \frac{\widetilde{d}_h}{\log\left(1 + \frac{1}{8} L_{trig}\right)} \tag{46}$$

with probability at least $1 - \delta$.

C.2. Proof of Lemma 5

The total regret up to episode K can be expressed as a sum of regret incurred in different phases

$$\textsc{Regret}(K) = \sum_{p=1}^{p_{tot}} \textsc{Regret}^{(p)}. \tag{47}$$

Notice that in every phase p, the S^{3}Q-LEARNING procedure is invoked with the policy replay memory $\Pi^{(p)}$ which consists of the mixture policy $\sum_{j=1}^{p-1} n^{(j)} \pi^{(j)}$ and in addition S^{4}Q-LEARNING plays the policy $\pi^{(p)}$ between Line 9 and Line 16 (in Algorithm 2) for exactly $n^{(p)}$ trajectories. Notice that in this proof the sequence $\{n^{(j)}\}_1^p$ is assumed to be fixed, i.e., non-random.

Outside of the call to S^{3}Q-LEARNING, S^{4}Q-LEARNING induces a regret exactly equal to

$$\text{S}^4\text{Q-LEARNING's } \textsc{Regret}^{(p)} = n^{(p)} \times \mathbb{E}_{S \sim \rho} \left(V_1^\star - V_1^{\pi^{(p)}} \right)(s). \tag{48}$$

In the same phase p, the regret due to the call to S^{3}Q-LEARNING in Line 6 of S^{4}Q-LEARNING is

$$\text{S}^3\text{Q-LEARNING's } \textsc{Regret}^{(p)} \propto H \sum_{j}^{p-1} N_Q^{(j)} \times \mathbb{E}_{S \sim \rho} \left(V_1^\star - V_1^{\pi^{(j)}} \right)(s) \tag{49}$$

where $N_Q^{(j)}$ is the random number of times that policy $\pi^{(j)}$ is actually played within S^{3}Q-LEARNING in phase p. Intuitively, $N_Q^{(j)} \approx n^{(p)}$ since $\mathbb{E} N_Q^{(j)} = n^{(p)}$. We make this precise by applying a Bernstein inequality for martingales (cf. Thm. 1 from the paper (Beygelzimer et al., 2011)).

Let $0 \leq X_j \overset{def}{=} \mathbb{E}_{S' \sim \rho} (V_1^\star - V_1^{\pi^{(j)}})(s') \leq 1$ be the random regret in step j of S^{3}Q-LEARNING; here the randomness comes from the random index j of the policy mixture. Let $n_{\text{tot}} = \sum_{j=1}^{p-1} n^{(j)}$; we can write

$$\sum_{t=1}^{n_{\text{tot}}} X_t = \sum_{j=1}^{p-1} N_Q^{(j)} \times \mathbb{E}_{S' \sim \rho} \left(V_1^\star - V_1^{\pi^{(j)}} \right)(s').$$

By construction, the X_t's are i.i.d., and $\mathbb{E} \sum_{t=1}^{n_{\text{tot}}} X_t = \sum_{j=1}^{p-1} n^{(j)} \mathbb{E}_{S' \sim \rho} (V_1^\star - V_1^{\pi^{(j)}})(S')$. Therefore, upon invoking Theorem 1 from the paper (Beygelzimer et al., 2011) and recalling that $X_t^2 \leq X_t$, we find that

$$\sum_{t=1}^{n_{\text{tot}}} X_t \leq \sum_{t=1}^{n_{\text{tot}}} \mathbb{E} X_t + 2 \sqrt{\left(\sum_{t=1}^{n_{\text{tot}}} \mathbb{E}_t X_t \right) \log(\tfrac{1}{\delta}) + 2 \log(\tfrac{1}{\delta})}$$

with probability at least $1 - \delta$. Completing the square on the right hand side and applying Cauchy–Schwarz inequality yields the upper bound $2 \sum_{t=1}^{n_{\text{tot}}} \mathbb{E} X_t + 3 \log\left(\frac{1}{\delta}\right)$.

Thus, the regret contributed by S^{3}Q-LEARNING in phase p can be upper bounded by the cumulative regret by S^{4}Q-LEARNING (excluding its call to S^{3}Q-LEARNING)—viz.

$$\text{S}^3\text{Q-LEARNING's } \textsc{Regret}^{(p)} \lesssim H \sum_{j}^{p-1} n^{(j)} \times \mathbb{E}_{S \sim \rho} \left(V_1^\star - V_1^{\pi^{(j)}} \right)(s) + 3H \log \frac{1}{\delta}.$$

Summing together these contributions over all phases, and applying the union bound over all possible phases yields the claim.

C.3. Proof of Lemma 6 *(Bonus Bound)*

The main part of the proof is to show that the empirical covariance matrices are accurate enough. S^{4}Q-LEARNING constructs the cumulative covariance used to construct the bonus to be used in phase p as the sum of two terms: 1) the covariance estimate returned by S^{3}Q-LEARNING in Line 6 of Algorithm 2 and 2) the increment obtained by S^{4}Q-LEARNING between Line 9 and Line 16 of Algorithm 2 . We can write:

$$\Sigma_h^{(p)} = \underbrace{\lambda_{\text{Reg}} I + \sum_{i=1}^{n^{(1:p-1)}} \phi_{ih}\phi_{ih}^\top}_{\text{S}^3\text{Q-LEARNING's Covariance Estimate}} + \underbrace{\sum_{j=1}^{n^{(p)}} \phi_{jh}\phi_{jh}^\top}_{\text{S}^4\text{Q-LEARNING's increment}} . \tag{50}$$

For simplicity we have denoted with $n^{(1:p-1)} = \sum_{i=1}^{p-1} n^{(i)}$; the first summation is over the feature vectors $\{\phi_i\}$ sampled by S^{3}Q-LEARNING and the second is over the feature vectors $\{\phi_j\}$ examined by S^{4}Q-LEARNING. Notice that there exists a setting $\lambda_{\text{Reg}} = \Theta(\log \frac{d}{\delta})$ that allows us to use Proposition 1 *(Concentration of Regularized Covariance)* twice and claim with probability at least $1-\delta/2$

$$\begin{aligned}
\Sigma_h^{(p)} &= \frac{\lambda_{\text{Reg}}}{2} I + \sum_{i=1}^{n^{(1:p-1)}} \phi_{ih}\phi_{ih}^\top + \frac{\lambda_{\text{Reg}}}{2} I + \sum_{j=1}^{n^{(p)}} \phi_{jh}\phi_{jh}^\top \\
&\preceq 2\lambda_{\text{Reg}} I + 2n^{(1:p-1)} \mathbb{E}_{\phi\sim\Pi^{(p-1)}} \phi_h\phi_h^\top + 2n^{(p)} \mathbb{E}_{\phi\sim\pi^{(p)}} \phi_h\phi_h^\top \\
&= 2\lambda_{\text{Reg}} I + 2n^{(1:p)} \mathbb{E}_{\phi\sim\Pi^{(p)}} \phi_h\phi_h^\top \\
&= 2\overline{\Sigma}_h^{(p)}.
\end{aligned}$$

(We have indicated with $\phi \sim \pi^{(p)}$ the random feature sampled according to the policy mixture using the policy replay memory). We conclude that under such event we must have

$$b_h^{(p)}(s,a) = \alpha_h^{(p)} \|\phi_h(s,a)\|_{(\Sigma_h^{(p)})^{-1}} \geq \alpha_h^{(p)} \|\phi_h(s,a)\|_{(\overline{\Sigma}_h^{(p)})^{-1}} = \mathcal{U}_h^{(p)}(s,a).$$

and under the same event

$$b_h^{(p)}(\cdot,\cdot) = \alpha_h^{(p)} \|\phi_h(\cdot,\cdot)\|_{(\Sigma_h^{(p)})^{-1}} \leq 2\alpha_h^{(p)} \|\phi_h(\cdot,\cdot)\|_{(\overline{\Sigma}_h^{(p)})^{-1}} = c_o \mathcal{U}_h^{(p)}(\cdot,\cdot). \tag{51}$$

A union bound over all possible phases and rescaling δ concludes.

C.4. Proof of Lemma 7

When S^{3}Q-LEARNING terminates, Theorem 3 ensures it returns a state-action value function $\widehat{Q}^\star$ such that

$$\begin{aligned}
\widehat{Q}_h^\star(s,a) &= \left(\mathcal{T}_\ell \widehat{Q}_{h+1}^\star\right)(s,a) + \mathcal{E}_h(s,a), \qquad \text{and} \\
\min\{0, \left(-\mathcal{U}_h^{(p)} + b_h^{(p)}\right)(s,a)\} &- \Delta_h^{(p)}(s,a) \leq \mathcal{E}_h(s,a),
\end{aligned}$$

where both relations hold uniformly over all state-action pairs (s,a). Conditioned on the event from Lemma 6, we have

$$-\Delta_h^{(p)}(s,a) = \min\{0, -\mathcal{U}_h^{(p)}(s,a) + b_h^{(p)}(s,a)\} - \Delta_h^{(p)}(s,a) \leq \mathcal{E}_h(s,a).$$

which implies that $\mathcal{E}_h(s,a) \geq -\Delta_h^{(p)}(s,a)$ for all state-action pairs and time steps h.

Using this bound, we now perform backwards induction over the timestep h in order to prove that

$$\widehat{Q}_{h+1}^\star(s_{h+1},a_{h+1}) \geq Q_{h+1}^\star(s_{h+1},a_{h+1}) - \sum_{\tau=h+1}^{H} \mathbb{E}_{(s'_\tau,a'_\tau)\sim\pi^\star|(s_{h+1},a_{h+1})} \Delta_h^{(p)}(s'_\tau,a'_\tau), \qquad \forall (s_{h+1},a_{h+1}) \tag{52}$$

For the base case $h = H$, all action-value functions are zero, so that the bound (52) certainly holds.

Now assume that the bound (52) holds at timestep $h+1$, for some $h \in \{1, \dots, H-1\}$; we need to show that it also holds at timestep h. Fix an arbitrary state-action pair (s,a). From our earlier lower bound, we have

$$\begin{aligned}
\widehat{Q}_h^\star(s,a) &= \left(\mathcal{T}_h \widehat{Q}_{h+1}^\star\right)(s,a) + \mathcal{E}_h(s,a) \\
&\geq \left(\mathcal{T}_h^{\pi^\star} \widehat{Q}_{h+1}^\star\right)(s,a) + \mathcal{E}_h(s,a) \\
&\overset{\text{(i)}}{\geq} \left(\mathcal{T}_h^{\pi^\star} Q_{h+1}^\star\right)(s,a) + \mathcal{E}_h(s,a) - \sum_{\tau=h+1}^{H} \mathbb{E}_{(s'_\tau, a'_\tau)\sim\pi^\star|(s,a)} \Delta_h^{(p)}(s'_\tau, a'_\tau) \\
&\geq \left(\mathcal{T}_h Q_{h+1}^\star\right)(s,a) - \sum_{\tau=h}^{H} \mathbb{E}_{(s'_\tau, a'_\tau)\sim\pi^\star|(s,a)} \Delta_h^{(p)}(s'_\tau, a'_\tau).
\end{aligned}$$

Here step (i) follows from the induction hypothesis. Thus, we have shown that the bound (52) holds at timestep h, which completes our proof via induction.

C.5. Proof of Lemma 8

We now prove the elliptic potential bound stated in Lemma 8. Let $\phi_{ih}^{(j)}$ be the i experienced feature vector at level h that S^{4}Q-LEARNING uses to check the triggering condition in Line 16 during phase j. Since $\lambda_{\text{Reg}} \geq 1$ we have $\|\phi\|_{(\Sigma_h^{(j)})^{-1}} \leq 1, \ \forall \|\phi\|_2 \leq 1$. Thus, when the triggering condition holds, the condition itself is not violated by much:

$$L_{\text{trig}} \leq \sum_{i=1}^{n^{(j)}} \|\phi_{ih}^{(j)}\|^2_{(\Sigma_h^{(j)})^{-1}} \leq \underbrace{\sum_{i=1}^{n^{(j)}-1} \|\phi_{ih}^{(j)}\|^2_{(\Sigma_h^{(j)})^{-1}}}_{\leq L_{\text{trig}}} + \underbrace{\|\phi_{n^{(j)}h}^{(j)}\|^2_{(\Sigma_h^{(j)})^{-1}}}_{\leq 1} \leq L_{\text{trig}} + 1. \tag{53}$$

Using Lemma 12 *(Proportional Estimates)* and Proposition 1 *(Concentration of Regularized Covariance)*, we find that with probability at least $1-\delta$

$$\begin{aligned}
n^{(j)} \mathbb{E}_{(S_h,A_h)\sim\pi^{(j)}} \|\phi_h(s_h,a_h)\|^2_{(\overline{\Sigma}_h^{(j)})^{-1}} &\leq 4 n^{(j)} \mathbb{E}_{(S_h,A_h)\sim\pi^{(j)}} \|\phi_h(s_h,a_h)\|^2_{(\Sigma_h^{(j)})^{-1}} \\
&\leq 8 \sum_{i=1}^{n^{(j)}} \|\phi_{ih}^{(j)}\|^2_{(\Sigma_h^{(j)})^{-1}} \leq 8\left(L_{\text{trig}} + 1\right).
\end{aligned}$$

Now recall that

$$\overline{\Sigma}_h^{(j+1)} = \overline{\Sigma}_h^{(j)} + n^{(j)} \mathbb{E}_{(S_h,A_h)\sim\pi^{(j)}} \phi_h(s_h,a_h)\phi_h(s_h,a_h)^\top.$$

Since $(L_{\text{trig}}+1) \geq e-1$, we can invoke Lemma 11 *(Information gain bounds)* so as to ensure that

$$n^{(j)} \mathbb{E}_{(S_h,A_h)\sim\pi^{(p)}} \|\phi_h(s_h,a_h)\|^2_{(\overline{\Sigma}^{(j)})^{-1}} \leq 8\left(L_{\text{trig}} + 1\right) \log \frac{\det\left(\overline{\Sigma}_h^{(j+1)}\right)}{\det\left(\overline{\Sigma}_h^{(j)}\right)}.$$

Summing over the phases and cancelling terms in the telescopic sum yields

$$\begin{aligned}
\sum_{j=1}^{p} n^{(j)} \mathbb{E}_{(S_h,A_h)\sim\pi^{(j)}} \|\phi_h(s_h,a_h)\|^2_{(\overline{\Sigma}_h^{(j)})^{-1}} &\leq 8\left(L_{\text{trig}} + 1\right) \log \frac{\det\left(\overline{\Sigma}_h^{(p+1)}\right)}{\det\left(\overline{\Sigma}_h^{(1)}\right)} \\
&\leq 8\left(L_{\text{trig}} + 1\right) \widetilde{d}_h.
\end{aligned}$$

A union bound over all possible phases concludes.

C.6. Proof of Lemma 9 (*Number of Phases*)

For invertible matrices A and B, note that $A \preceq B$ implies that $A^{-1} \succeq B^{-1}$, and moreover, we have the equivalence $A \preceq B \iff x^\top Ax \leq x^\top Bx$ for all $x \in \mathbb{R}^d$. We now combine Lemma 12 with Proposition 1 so as to argue that Algorithm 2 makes sufficient progress. In particular, consider each time that Line 16 of Algorithm 2 triggers a new phase at level h. Then with probability at least $1 - \delta$, we must have

$$\begin{aligned} n^{(p)}\mathbb{E}_{(S_h,A_h)\sim\pi^{(p)}} \|\phi_h(s_h,a_h)\|^2_{(\overline{\Sigma}_h^{(p)})^{-1}} &\geq \frac{1}{4} n^{(p)}\mathbb{E}_{(S_h,A_h)\sim\pi^{(p)}} \|\phi_h(s_h,a_h)\|^2_{(\Sigma_h^{(p)})^{-1}} \\ &\geq \frac{1}{8}\sum_{i=1}^{n^{(p)}} \|\phi_{ih}^{(p)}\|^2_{(\Sigma_h^{(p)})^{-1}} \\ &\geq \frac{1}{8} L_{\text{trig}}. \end{aligned}$$

When this bound holds and the the triggering condition is satisfied at level h in phase p, then the information ratio must increase by a constant fraction: more precisely, Lemma 11 guarantees that

$$\begin{aligned} \frac{\det(\overline{\Sigma}_h^{(p+1)})}{\det \overline{\Sigma}_h^{(p)}} &= \frac{\det\left(\overline{\Sigma}_h^{(p)} + n^{(p)}\mathbb{E}_{(S_h,A_h)\sim\pi^{(p)}}\phi_h(s_h,a_h)\phi_h(s_h,a_h)^\top\right)}{\det \overline{\Sigma}_h^{(p)}} \\ &\geq 1 + n^{(p)}\mathbb{E}_{(S_h,A_h)\sim\pi^{(p)}} \|\phi_h(s_h,a_h)\|^2_{(\overline{\Sigma}_h^{(p)})^{-1}} \\ &\geq 1 + \frac{1}{8} L_{\text{Trigger}}(\delta_{\text{phase}}). \end{aligned}$$

By induction, in phase p we must have (notice that we are ignoring the additional contribution that arises when level h is not the one that triggers a new phase)

$$\frac{\det \overline{\Sigma}_h^{(p+1)}}{\det \overline{\Sigma}_h^{(1)}} \geq \left(1 + \frac{1}{8} L_{\text{trig}}\right)^{s_h^{(p)}} \tag{54}$$

where $s_h^{(p)}$ is the number of switches up to phase p that were triggered at level h. Taking $\log$ gives

$$s_h^{(p)} \leq \frac{\log \frac{\det \overline{\Sigma}_h^{(p+1)}}{\det \overline{\Sigma}_h^{(1)}}}{\log\left(1 + \frac{1}{3} L_{\text{trig}}\right)}. \tag{55}$$

Recalling the total number of switches across levels equals the total number of phases, i.e., $\sum_{h=1}^H s_h^{(p)} = p_{tot}$, together with the relevant union bound over phases concludes.

D. Auxiliary results

In this appendix, we collect together various auxiliary results that we use in our main argument, along with their proofs.

D.1. Information Gain

Lemma 10 (Upper Bound on Information Gain)**.**

$$\log\left(\frac{\det \overline{\Sigma}_h^{(p_{tot}+1)}}{\det \overline{\Sigma}_h^{(1)}}\right) \leq \widetilde{d}_h \leq d \log \frac{K}{d\lambda_{Reg}}. \tag{56}$$

We must now bound the determinant of $\overline{\Sigma}_h^{(p_{tot})}$ to compute the maximum number of phases; we proceed in a way similar to Lemma 10 of (Abbasi-Yadkori et al., 2011), the only difference being that the increments are not rank one. Let $\alpha_1, \ldots, \alpha_d$

be the eigenvalues of $\overline{\Sigma}_h^{(p_{tot})}$. We must have

$$\det\left(\overline{\Sigma}_h^{(p_{tot})}\right) = \prod_i \alpha_i \leq \left(\frac{\sum_i \alpha_i}{d}\right)^d = \left(\frac{\operatorname{Tr}\overline{\Sigma}_h^{(p_{tot})}}{d}\right)^d. \tag{57}$$

We can upper bound the trace as follows:

$$\begin{aligned}\operatorname{Tr}\left(\overline{\Sigma}_h^{(p_{tot})}\right) = \sum_{p=1}^{p_{tot}} \operatorname{Tr}\left(n^{(p)}\overline{M}_h^{(p)}\right) &= \sum_{p=1}^{p_{tot}} n^{(p)} \operatorname{Tr}\left(\mathbb{E}_{(S_h,A_h)\sim\pi^{(p)}}\phi_h(s_h,a_h)\phi_h(s_h,a_h)^\top\right)\\ &= \sum_{p=1}^{p_{tot}} n^{(p)}\mathbb{E}_{(S_h,A_h)\sim\pi^{(p)}} \operatorname{Tr}\left(\phi_h(s_h,a_h)\phi_h(s_h,a_h)^\top\right)\\ &\leq \sum_{p=1}^{p_{tot}} n^{(p)}\\ &\leq K.\end{aligned}$$

Combining with the prior displays and recalling that $\overline{\Sigma}_h^{(0)} = \lambda_{\text{Reg}} I$ yields the claim.

D.2. Bounds on the information gain

Lemma 11 (Information gain bounds). *For any random vector $\phi \in \mathbb{R}^d$, scalar $\alpha > 0$ and positive definite matrix Σ, we have the upper bound*

$$\log\frac{\det(\Sigma + \alpha\mathbb{E}[\phi\phi^\top])}{\det\Sigma} \leq \alpha\mathbb{E}\|\phi\|^2_{\Sigma^{-1}}. \tag{58a}$$

Moreover, we have the lower bounds

$$\log\frac{\det(\Sigma + \alpha\mathbb{E}[\phi\phi^\top])}{\det(\Sigma)} \geq \log\left(1 + \alpha\mathbb{E}\|\phi\|^2_{\Sigma^{-1}}\right) \overset{(i)}{\geq} \frac{\alpha}{L}\mathbb{E}\|\phi\|^2_{\Sigma^{-1}}, \tag{58b}$$

where (i) holds whenever $\alpha\mathbb{E}\|\phi\|^2_{\Sigma^{-1}} \leq L$ for some $L \geq e - 1$,

Proof. We first begin with equivalent expression for the determinant ratio. Letting $\lambda_j(M)$ denote the j^{th}-ordered eigenvalue of a matrix M, we have

$$\begin{aligned}\frac{\det(\Sigma + \alpha\mathbb{E}\phi\phi^\top)}{\det\Sigma} = \det(I + \alpha\Sigma^{-\frac{1}{2}}\mathbb{E}[\phi\phi^\top]\Sigma^{-\frac{1}{2}}) &= \prod_{j=1}^d \lambda_j\left(I + \alpha\Sigma^{-\frac{1}{2}}\mathbb{E}\phi\phi^\top\Sigma^{-\frac{1}{2}}\right)\\ &= \prod_{j=1}^d\left(1 + \alpha\lambda_j\left(\Sigma^{-\frac{1}{2}}\mathbb{E}\phi\phi^\top\Sigma^{-\frac{1}{2}}\right)\right).\end{aligned} \tag{59}$$

Proof of the upper bound (58a)**:** Taking logarithms in equation (59) and using the elementary bound $\log(1+t) \leq t$, valid for $t \geq 0$, we have

$$\begin{aligned}\log\left(\frac{\det(\Sigma + \alpha\mathbb{E}\phi\phi^\top)}{\det\Sigma}\right) = \sum_{j=1}^d \log\left(1 + \alpha\lambda_j\left(\Sigma^{-\frac{1}{2}}\mathbb{E}\phi\phi^\top\Sigma^{-\frac{1}{2}}\right)\right) &\leq \sum_{j=1}^d\left\{\alpha\lambda_j\left(\Sigma^{-\frac{1}{2}}\mathbb{E}\phi\phi^\top\Sigma^{-\frac{1}{2}}\right)\right\}\\ &\overset{(i)}{=} \alpha\operatorname{Tr}\left(\Sigma^{-\frac{1}{2}}\mathbb{E}\phi\phi^\top\Sigma^{-\frac{1}{2}}\right)\\ &\overset{(ii)}{=} \alpha\mathbb{E}\|\phi\|^2_{\Sigma^{-1}},\end{aligned}$$

where step (i) follows since the trace is equal to the sum of the eigenvalues, and step (ii) follows from cyclic properties of the trace operator, and some algebra.

Proof of the lower bound (58b)**:** In order to prove the lower bound, we again begin with equation (59). Notice that the eigenvalues are all non-negative since the matrix $\Sigma^{-\frac{1}{2}}\mathbb{E}[\phi\phi^\top]\Sigma^{-\frac{1}{2}}$ is positive semidefinite. Thus, we can ignore the higher-order terms in the product so as to obtain the lower bound

$$\frac{\det(\Sigma + \alpha \mathbb{E}\phi\phi^\top)}{\det \Sigma} \geq 1 + \alpha \sum_{j=1}^{d} \lambda_j \left(\Sigma^{-\frac{1}{2}} \mathbb{E}\phi\phi^\top \Sigma^{-\frac{1}{2}} \right) \; = \; 1 + \alpha \mathbb{E}\|\phi\|^2_{\Sigma^{-1}}, \tag{60a}$$

where the final equality follows by the same sequence of calculations as in step (ii) above.

In order to complete the proof, observe that $f(x) = \log(1+x)$ is a concave function. Thus, for any $a \geq e-1$ and $x \in [0, a]$, we can set $\lambda = \frac{x}{a} \in [0,1]$, and write

$$\log(1+x) \; = \; f(x) = f\big(\lambda a + (1-\lambda)0\big) \overset{(iii)}{\geq} \lambda f(a) + (1-\lambda) f(0) \; = \; \lambda f(a) \overset{(iv)}{\geq} \lambda = \frac{x}{a}, \tag{60b}$$

where step (iii) follows from Jensen's inequality; and step (iv) is valid for any $a \geq e-1$.

Finally, taking logarithms in inequality (60a) and applying the lower bound (60b) yields

$$\log \frac{\det(\Sigma + \alpha \mathbb{E}\phi\phi^\top)}{\det \Sigma} \geq \log\left(1 + \alpha \mathbb{E}\|\phi\|^2_{\Sigma^{-1}}\right) \geq \frac{\alpha}{L} \mathbb{E}\|\phi\|^2_{\Sigma^{-1}},$$

as claimed. □

D.3. Proportional estimates under the triggering condition

Suppose that the triggering condition holds, so that we have the lower bound

$$\sum_{i=1}^{n} Z_i \geq 32 \varphi_{\sqrt{n}} \left(\frac{\delta}{2n^2} \right) + 8 \varphi_n \left(\frac{\delta}{2n^2} \right) \overset{def}{=} L_{\text{Trigger}}(\delta). \tag{61}$$

The following lemma shows that under this condition, the sample average $\widehat{S}_n \overset{def}{=} \frac{1}{n}\sum_{i=1}^{n} Z_i$ is close to the expectation $\mathbb{E}[Z]$.

Lemma 12 (Proportional Estimates)**.** *Under the triggering condition* (61)*, for any $\delta \in (0,1)$, we have the sandwich result*

$$\tfrac{1}{2}\widehat{S}_n \leq \mathbb{E}[Z] \leq \tfrac{3}{2}\widehat{S}_n \qquad \textit{with prob. at least } 1-\delta. \tag{62}$$

In order to prove this claim, we first show that for any fixed n at which the triggering conditioning holds, the sandwich bound (62) holds with probability at least $1-\delta'$, where $\delta' = \frac{\delta}{2n^2}$. We can then take a union bound over all $n = 1, 2, \ldots$ to conclude that for any n, sandwich bound (62) holds with probability at least $1 - \sum_{n=1}^{\infty} \frac{\delta}{2n^2} \geq 1 - \delta$, as required.

Thus, for the remainder of the proof, we study the problem for a fixed sample size n. Our proof is based two auxiliary results. First, for i.i.d. random variables $\{Z_i\}_{i=1}^n$ taking values in $[0,1]$, the sample average $\widehat{S}_n$ satisfies the following empirical Bernstein bound: for any $\delta \in (0,1)$,

$$\mathbb{P}\left[\big|\widehat{S}_n - \mathbb{E}[Z]\big| \leq \sqrt{\frac{\varphi_{\sqrt{n}}(\delta)\widehat{\text{Var}}Z}{n}} + \frac{\varphi_n(\delta)}{n-1} \right] \geq 1 - \delta \tag{63}$$

where $\varphi_{\sqrt{n}}(\delta) = 2\log(\frac{4}{\delta})$; $\varphi_n(\delta) = \frac{7}{3}\log(\frac{4}{\delta})$, and

$$\widehat{\text{Var}}Z = \frac{1}{n(n-1)} \sum_{1 \leq i < j \leq n} (Z_i - Z_j)^2$$

is the empirical variance. This claim is a consequence of two applications of the empirical Bernstein bound from the paper (Maurer & Pontil, 2009), as applied to the random variables Z and then $1-Z$, followed by a union bound to obtain the two-sided claim give here.

Next, we observe that the sample variance can be upper bounded as

$$\widehat{\mathrm{Var}} Z \overset{def}{=} \frac{1}{n-1}\left(\sum_{i=1}^{n} Z_i^2 - \sum_{i=1}^{n} Z_i\right) \leq \frac{1}{n-1}\sum_{i=1}^{n} Z_i^2 \leq \frac{1}{n-1}\sum_{i=1}^{n} Z_i \leq \tfrac{n}{n-1} \ \leq \ 2, \tag{64}$$

using the fact that each $Z_i \in [0,1]$.

Now combining the empirical Bernstein bound (63) with the variance bound (64), we find that

$$\left|\widehat{S}_n - \mathbb{E}[Z]\right| \leq \tfrac{1}{n}\Big\{\sqrt{2\widehat{S}_n \varphi_{\sqrt{n}}(\delta')} + 2\varphi_n(\delta')\Big\} \qquad \text{with prob. at least } 1-\delta'. \tag{65}$$

The triggering condition (61) ensures that $n\widehat{S}_n \geq 32\varphi_{\sqrt{n}}(\delta') + 8\varphi_n(\delta')$, whence $\frac{1}{n}\varphi_{\sqrt{n}}(\delta) \leq \frac{\widehat{S}_n}{32}$ and $\frac{1}{n}\varphi_n(\delta) \leq \frac{\widehat{S}_n}{8}$. Putting together the pieces, we find that

$$|\widehat{S}_n - \mathbb{E}[Z]| \leq \tfrac{1}{n}\{\sqrt{2\widehat{S}_n\varphi_{\sqrt{n}}(\delta)} + 2\varphi_n(\delta)\} \leq \sqrt{\frac{(\widehat{S}_n)^2}{16}} + \frac{\widehat{S}_n}{4} \ = \ \frac{\widehat{S}_n}{2}$$

with probability at least $1-\delta'$. Re-arranging shows that we have the sandwich relation $\frac{1}{2}\widehat{S}_n \leq \mathbb{E}[Z] \leq \frac{3}{2}\widehat{S}_n$ with probability $1-\delta'$, as claimed.

D.4. Non-Isotropic Proportional Estimates of the Empirical Covariance

An important step in the analysis is ensuring that the empirical covariance matrices computed by the algorithm are sufficiently close to their (conditional) expectations. In this section, we discuss how to use matrix Chernoff techniques to establish the requisite bounds.

Let $\{Z_k\}_{k=1}^n$ be a sequence of independent, symmetric and positive definite random matrices of dimension $d.\ d$. Suppose that

$$0 \leq \lambda_{\min}(Z_k), \qquad \lambda_{\max}(Z_k) \leq L_Z, \qquad \text{for all } k = 1, \ldots, K. \tag{66}$$

The following result provides bounds on the sum $W = \sum_{k=1}^K Z_k$.

Proposition 1 (Concentration of Regularized Covariance)**.** *Under the above conditions, for any* $\delta \in (0,1)$ *and* $\lambda \geq 2L_Z\frac{\log\frac{2d}{\delta}}{\log\frac{36}{35}}$*, we have*

$$\frac{1}{2}(W + \lambda I) \preceq \mathbb{E}W + \lambda I \preceq \frac{3}{2}(W+\lambda I). \tag{67}$$

with probability at least $1-\delta$*.*

Proof. In fact, we establish a somewhat more general claim: namely, for any $\epsilon > 0$ and $\lambda > 0$, we have

$$\mathbb{P}\Big(\frac{1}{1+\epsilon}(W+\lambda I) \preceq \mathbb{E}W + \lambda I \preceq \frac{1}{1-\epsilon}(W+\lambda I)\Big) \geq 1 - 2d\left(1 - \frac{\epsilon^2}{4}\right)^{\frac{\lambda}{L_Z + \frac{\lambda}{K}}}. \tag{68}$$

In order to recover the stated claim (67) we fix $\epsilon = \frac{1}{3}$. On one hand, if $L_Z \leq \frac{1}{2}\frac{\lambda}{K}$, then we have the (deterministic) sandwich relations

$$0 \preceq W \preceq \tfrac{1}{2}\lambda I, \quad \text{and} \quad 0 \preceq \mathbb{E}W \preceq \tfrac{1}{2}\lambda I$$

so that the bound (67) holds deterministically. On the other hand, if $L_Z \geq \frac{1}{2}\frac{\lambda}{K}$, then the claim follows by choosing $\lambda \geq 2L_Z\frac{\log(\frac{2d}{\delta})}{\log(\frac{36}{35})}$.

In order to prove the bound (68), we make use of the following matrix Chernoff inequality:

Lemma 13 (Matrix Chernoff). *Consider the sum $Y = \sum_{k=1}^K X_k$ of a sequence $\{X_k\}_{k \ge 1}$ of independent, symmetric PSD matrices whose eigenvalues all lie in the interval $[0, L]$, and suppose that $\mathbb{E}[Y] = I$. Then we have*

$$(1-\epsilon)I \preceq Y \preceq (1+\epsilon)I \tag{69}$$

with probability at least $1 - 2d\left(1 - \frac{\epsilon^2}{4}\right)^{\frac{1}{L}}$ for all $\epsilon \in (0,1)$.

This claim follows by applying Theorem 5.1.1 from Tropp (Tropp, 2015) twice, for the upper and lower tail respectively, combined with the inequalities

$$\frac{e^{-\epsilon}}{(1-\epsilon)^{1-\epsilon}} \le 1 - \frac{\epsilon^2}{4}, \quad \text{and} \quad \frac{e^{\epsilon}}{(1+\epsilon)^{1+\epsilon}} \le 1 - \frac{\epsilon^2}{4}, \quad \text{valid for any } \epsilon \in [0,1),$$

along with the fact that $a \le b$ implies that $a^x \le b^x$ for all strictly positive scalars a, b, x.

Using Lemma 13, we can now prove the bound (68). We define "regularized" versions of Z_k and W via $X'_k \overset{def}{=} Z_k + \frac{\lambda}{K} I$ and $Y' \overset{def}{=} \sum_{k=1}^K X'_k$. By definition, we have

$$Y' = \sum_{k=1}^K \left(Z_k + \frac{\lambda}{K} I \right) = W + \lambda I \quad \text{and} \quad \mathbb{E}Y' = \mathbb{E}W + \lambda I. \tag{70}$$

Thus, in order to prove the claim, it suffices to establish a high probability bound on the event

$$\mathcal{E} \overset{def}{=} \left\{ (1-\epsilon)Y' \preceq \mathbb{E}Y' \preceq (1+\epsilon)Y' \right\}. \tag{71}$$

Since $\lambda_{\min}(\mathbb{E}Y') \ge \lambda$, the matrix $\mathbb{E}Y'$ is strictly positive definite, and the matrix $(\mathbb{E}Y')^{-\frac{1}{2}}$ exists. We use it to define the new matrices

$$\begin{aligned} X_k &\overset{def}{=} (\mathbb{E}Y')^{-\frac{1}{2}} X'_k (\mathbb{E}Y')^{-\frac{1}{2}}, \quad \text{and} \\ Y &\overset{def}{=} \sum_{k=1}^K X_k = (\mathbb{E}Y')^{-\frac{1}{2}} \left(\sum_{k=1}^K X'_k \right) (\mathbb{E}Y')^{-\frac{1}{2}} = (\mathbb{E}Y')^{-\frac{1}{2}} (Y') (\mathbb{E}Y')^{-\frac{1}{2}}. \end{aligned} \tag{72}$$

Note that we have $\mathbb{E}[Y] = I$ by construction, so that the matrix Chernoff bound (69) can be applied. We observe that

$$\lambda_{\max}(X_k) = \|X_k\|_2 \le \| (\mathbb{E}Y')^{-\frac{1}{2}} X'_k (\mathbb{E}Y')^{-\frac{1}{2}} \|_2 \le \| (\mathbb{E}Y')^{-\frac{1}{2}} \|_2 \|X'_k\|_2 \| (\mathbb{E}Y')^{-\frac{1}{2}} \|_2 \le \frac{1}{\lambda} \left(L_Z + \frac{\lambda}{K} \right) \overset{def}{=} L.$$

Applying the bound (69) yields

$$(1-\epsilon)I \preceq Y \preceq (1+\epsilon)I.$$

with the stated probability. Finally, we can pre- and post-multiply by $(\mathbb{E}Y')^{\frac{1}{2}}$ and then use Eq. (72) so as to obtain

$$(1-\epsilon)\mathbb{E}Y' \preceq (\mathbb{E}Y')^{\frac{1}{2}} Y (\mathbb{E}Y')^{\frac{1}{2}} = Y' \preceq (1+\epsilon)\mathbb{E}Y'.$$

Recalling the definition (70) of Y', we see that this sandwich is equivalent to the stated claim (67). □

D.5. Concentration of Log Determinants

In this section, we prove the following claim:

Lemma 14 (Concentration of Log Determinants). *Let $\{x_i\}$ be i.i.d. vector random variables from some distribution such that $\|x_i\|_2 \le 1$. If $\lambda \gtrsim \log(\frac{dn}{\delta}) \ge 1$ and $G_1 \succeq \lambda I$ then with probability at least $1-\delta$ jointly for all $n = 1, 2, \dots$ it holds that*

$$\frac{1}{4} \log \frac{\det(G_1 + n\mathbb{E}xx^\top)}{\det G_1} - (8\sqrt{2}+4) \log \frac{8n^2}{\delta} \le \log \frac{\det\left(G_1 + \sum_{i=1}^n x_i x_i^\top\right)}{\det G_1} \le 8 \log \frac{\det(G_1 + n\mathbb{E}xx^\top)}{\det G_1} + 8 \log \frac{8n^2}{\delta}.$$

Let us now prove it. Let $G_i = G_1 + \sum_{j=1}^{i-1} x_j x_j^\top$. Using Lemma 11 in (Abbasi-Yadkori et al., 2011) we can write

$$\frac{1}{2}\sum_{i=1}^{n} \|x_i\|^2_{G_i^{-1}} \le \log \frac{\det\left(G_0 + \sum_{i=1}^{n} x_i x_i^\top\right)}{\det G_0} \le \sum_{i=1}^{n} \|x_i\|^2_{G_i^{-1}}.$$

Thus, we will now focus on bounding the sums of the quadratic functions. Proposition 1 together with a double union bound over n ensures that if $\lambda \gtrsim \log(\frac{dn}{\delta})$ then for all n with probability at least $1-\delta/2$, we have

$$\begin{aligned}\sum_{i=1}^{n} \|x_i\|^2_{G_i^{-1}} = \sum_{i=1}^{n} x_i G_i^{-1} x_i \le 2x_i \Big(G_1 + \mathbb{E}\sum_{j=1}^{i-1} x_j x_j^\top\Big)^{-1} x_i \\ = 2x_i \Big(\underbrace{G_1 + (i-1)\mathbb{E}xx^\top}_{=\mathbb{E}G_i}\Big)^{-1} x_i \;=\; 2\|x_i\|^2_{(\mathbb{E}G_i)^{-1}}.\end{aligned}$$

Similarly, under the same event specified by Proposition 1, we have

$$\begin{aligned}\sum_{i=1}^{n} \|x_i\|^2_{G_i^{-1}} = \sum_{i=1}^{n} x_i G_i^{-1} x_i \ge \frac{1}{2}x_i \Big(G_1 + \mathbb{E}\sum_{j=1}^{i-1} x_j x_j^\top\Big)^{-1} x_i \\ = \frac{1}{2}x_i \Big(\underbrace{G_1 + (i-1)\mathbb{E}xx^\top}_{=\mathbb{E}G_i}\Big)^{-1} x_i \;=\; \frac{1}{2}\|x_i\|^2_{(\mathbb{E}G_i)^{-1}}.\end{aligned}$$

Now consider the random variable $X_i = \|x_i\|^2_{(\mathbb{E}G_i)^{-1}} - \mathbb{E}\|x_i\|^2_{(\mathbb{E}G_i)^{-1}}$. Since $\mathbb{E}G_i \succeq I$, we have

$$\mathbb{E}X_i^2 \le \mathbb{E}\|x_i\|^4_{(\mathbb{E}G_i)^{-1}} \le \mathbb{E}\|x_i\|^2_{(\mathbb{E}G_i)^{-1}} \le 1.$$

Applying a Bernstein martingale inequality (cf. Theorem 1 from the paper (Beygelzimer et al., 2011)) and combining with the union bound yields

$$\left|\sum_{i=1}^{n} X_i\right| \le 2\sqrt{\left(\sum_{i=1}^{n} \mathbb{E}\|x_i\|^2_{(\mathbb{E}G_i)^{-1}}\right)\log(\tfrac{8n^2}{\delta})} + 2\log(\tfrac{8n^2}{\delta})$$

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

It remains to bound the sum of the predictable expectations under square root. Coupling Lemma 11 *(Information gain bounds)* with Lemma 11 *(Information gain bounds)* under the conditions $\alpha = 1, L = 2 > e - 1$ we obtain

$$\log \frac{\det(\mathbb{E}G_i + \mathbb{E}x_i x_i^\top)}{\det \mathbb{E}G_i} \le \mathbb{E}\|x_i\|^2_{(\mathbb{E}G_i)^{-1}} \le 2\log \frac{\det(\mathbb{E}G_i + \mathbb{E}x_i x_i^\top)}{\det \mathbb{E}G_i}. \tag{73}$$

Summing over $i \in [n]$, recalling $\mathbb{E}G_{i+1} = \mathbb{E}G_i + \mathbb{E}x_i x_i^\top$ and cancelling the terms in the telescoping sum gives

$$\log \frac{\det \mathbb{E}G_{n+1}}{\det G_1} \le \sum_{i=1}^{n} \mathbb{E}\|x_i\|^2_{(\mathbb{E}G_i)^{-1}} \le 2\log \frac{\det \mathbb{E}G_{n+1}}{\det G_1}.$$

We are now ready to show the upper bound. Removing the absolute value, using Eq. (73) to bound the quadratic sum one obtains with probability $1-\delta$

$$\begin{aligned}\sum_{i=1}^{n} \|x_i\|^2_{G_i^{-1}} &\le 4\log \frac{\det(\mathbb{E}G_{n+1})}{\det G_1} + 4\sqrt{2\log \frac{\det(\mathbb{E}G_{n+1})}{\det G_1} \log \frac{8n^2}{\delta}} + 4\log \frac{8n^2}{\delta} \\ &\le 8\log \frac{\det(\mathbb{E}G_{n+1})}{\det G_1} + 8\log \frac{8n^2}{\delta}.\end{aligned}$$

The last inequality follows from completing the square and using Cauchy-Schwartz to simplify the statement.

We show the lower bound on the similar fashion. By lifting the absolute value one obtains with probability at least $1-\delta$

$$\begin{aligned}\sum_{i=1}^{n} \|x_i\|^2_{G_i^{-1}} &\geq \frac{1}{2}\log\frac{\det(\mathbb{E}G_{n+1})}{\det G_1} - 4\sqrt{2\log\frac{\det(\mathbb{E}G_{n+1})}{\det G_1}\log\frac{8n^2}{\delta}} - 4\log\frac{8n^2}{\delta} \\ &\geq \frac{1}{4}\log\frac{\det(\mathbb{E}G_{n+1})}{\det G_1} - (8\sqrt{2}+4)\log\frac{8n^2}{\delta}.\end{aligned}$$

D.6. Constrained Loss Lemmas

In this section, let $\mathbb{E}$ denote the expectation operator for a pair $(X, Y) \sim \mu$, where $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$. Assume that the second moment matrix $\mathbb{E}[XX^\top]$ is strictly positive definite. Define the loss function $\mathcal{L}(\theta) = \mathbb{E}(\langle X,\, \theta\rangle - Y)^2$, along with the unconstrained minimizer $\theta^\star \overset{def}{=} \arg\min_{\theta\in\mathbb{R}^d}\mathcal{L}(\theta)$. Our first result gives an equivalent expression for the excess loss $\mathcal{L}(\theta) - \mathcal{L}(\theta^\star)$.

Lemma 15 (Excess Loss)**.** *The excess loss can be written as*

$$\mathcal{L}(\theta) - \mathcal{L}(\theta^\star) = \|\theta - \theta^\star\|^2_{\mathbb{E}[XX^\top]}. \tag{74}$$

Proof. Since the loss is a strongly convex quadratic function, the minimizer must satisfy the zero-gradient condition

$$0 = \tfrac{1}{2}\nabla_\theta \mathcal{L}(\theta^\star) = \mathbb{E}[X(\langle X,\, \theta\rangle - Y)]\big|_{\theta=\theta^\star} \Longrightarrow \mathbb{E}[XX^\top]\theta^\star = \mathbb{E}[XY]. \tag{75}$$

We now use this relation to establish the claim. We have

$$\begin{aligned}\mathcal{L}(\theta) - \mathcal{L}(\theta^\star) &= \mathbb{E}(\langle X,\, \theta\rangle - Y)^2 - \mathbb{E}(\langle X,\, \theta^\star\rangle - Y)^2 \\ &= \mathbb{E}\Big[(\langle X,\, \theta\rangle - Y) - (\langle X,\, \theta^\star\rangle - Y)\Big]\Big[(\langle X,\, \theta\rangle - Y) + (\langle X,\, \theta^\star\rangle - Y)\Big] \\ &= \mathbb{E}\Big[(\theta-\theta^\star)^\top X\Big]\Big[X^\top(\theta-\theta^\star) - Y + 2\langle X,\, \theta^\star\rangle - Y\Big] \\ &= \Big[(\theta-\theta^\star)^\top(\mathbb{E}XX^\top)(\theta-\theta^\star)\Big] + 2\mathbb{E}\Big[(\theta-\theta^\star)^\top\left(XX^\top\theta^\star - XY\right)\Big] \\ &= \|\theta-\theta^\star\|^2_{\mathbb{E}XX^\top} + 2(\theta-\theta^\star)^\top \underbrace{\Big[\mathbb{E}(XX^\top)\theta^\star - \mathbb{E}[XY]\Big]}_{=0 \text{ by Eq. (75)}}.\end{aligned}$$

□

Lemma 16 (Excess Risk with Regularization)**.** *For a fixed $\lambda > 0$, define*

$$\mathcal{L}(w) = \tfrac{1}{2}\mathbb{E}_{(X,Y)}\big(\langle X,\, w\rangle - Y\big)^2, \quad \textit{and} \quad w^\star \in \underset{\|w\|_2\leq B}{\arg\min}\,\mathcal{L}(w).$$

Then for any scalar $M > 0$, we have

$$\|w - w^\star\|^2_{(M\mathbb{E}_X[XX^\top]+\lambda I)} \leq 2M\big(\mathcal{L}(w) - \mathcal{L}(w^\star)\big) + \lambda\|w - w^\star\|_2^2. \tag{76}$$

Proof. We adopt the shorthand $\mathbb{E}$ for $\mathbb{E}_{(X,Y)}$. We can write (two times) the excess risk as

$$\begin{aligned}2\Big[\mathcal{L}(w) - \mathcal{L}(w^\star)\Big] &= \mathbb{E}\left(\langle X,\, w\rangle - y\right)^2 - \mathbb{E}\left(\langle X,\, w\rangle^\star - y\right)^2 \\ &= \mathbb{E}\Big[\left(\langle X,\, w\rangle - y\right) - \left(\langle X,\, w\rangle^\star - Y\right)\Big]\Big[\langle X,\, w\rangle - Y + \langle X,\, w\rangle^\star - Y\Big] \\ &= \mathbb{E}\Big[X^\top(w - w^\star)\Big]\Big[\langle X,\, w\rangle - Y + \langle X,\, w\rangle^\star - y\Big] \\ &= \mathbb{E}\Big[X^\top(w - w^\star)\Big]\Big[X^\top(w-w^\star) + \langle X,\, w\rangle^\star - y + \langle X,\, w^\star\rangle - Y\Big] \\ &= (w-w^\star)^\top\,\mathbb{E}\left(XX^\top\right)(w-w^\star) + 2\mathbb{E}\Big[x^\top(w-w^\star)\Big]\Big[\langle X,\, w^\star\rangle - Y\Big].\end{aligned} \tag{77}$$

The optimality condition for $w^\star$ reads

$$\mathbb{E}\Big[\big(\langle X,\, w^\star\rangle - Y\big) X^\top\Big](w - w^\star) \geq 0 \qquad \text{for any feasible } w.$$

Applying this inequality to equation (77) yields

$$\begin{aligned} 2\Big[\mathcal{L}(w) - \mathcal{L}(w^\star)\Big] &\geq (w - w^\star)^\top \, \mathbb{E}\left(XX^\top\right)(w - w^\star) \\ &= (w - w^\star)^\top \left[\mathbb{E}XX^\top + \frac{\lambda}{M} I\right](w - w^\star) - \frac{\lambda}{M}\|w - w^\star\|_2^2, \end{aligned}$$

as claimed. □