Sample-Efficient Reinforcement Learning Is Feasible for Linearly Realizable MDPs with Limited Revisiting

Gen Li[*] Yuxin Chen[*] Yuejie Chi[†] Yuantao Gu[‡] Yuting Wei[§]
Princeton   Princeton   CMU   Tsinghua   UPenn

Abstract

Low-complexity models such as linear function representation play a pivotal role in enabling sample-efficient reinforcement learning (RL). The current paper pertains to a scenario with value-based linear representation, which postulates linear realizability of the optimal Q-function (also called the "linear $Q^\star$ problem"). While linear realizability alone does not allow for sample-efficient solutions in general, the presence of a large sub-optimality gap is a potential game changer, depending on the sampling mechanism in use. Informally, sample efficiency is achievable with a large sub-optimality gap when a generative model is available, but is unfortunately infeasible when we turn to standard online RL settings.

In this paper, we make progress towards understanding this linear $Q^\star$ problem by investigating a new sampling protocol, which draws samples in an online/exploratory fashion but allows one to backtrack and revisit previous states in a *controlled* and *infrequent* manner. This protocol is more flexible than the standard online RL setting, while being practically relevant and far more restrictive than the generative model. We develop an algorithm tailored to this setting, achieving a sample complexity that scales polynomially with the feature dimension, the horizon, and the inverse sub-optimality gap, but not the size of the state/action space. Our findings underscore the fundamental interplay between sampling protocols and low-complexity function representation in RL.

Keywords: reinforcement learning, linearly realizable optimal Q-functions, sub-optimality gap, state revisiting, sample efficiency

1 Introduction

Emerging reinforcement learning (RL) applications necessitate the design of sample-efficient solutions in order to accommodate the explosive growth of problem dimensionality. Given that the state space and the action space could both be unprecedentedly enormous, it is often infeasible to request a sample size exceeding the fundamental limit set forth by the ambient dimension in the tabular setting (which enumerates all combinations of state-action pairs). As a result, the quest for sample efficiency cannot be achieved in general without exploiting proper low-complexity structures underlying the problem of interest.

1.1 Linear function approximation

Among the studies of low-complexity models for RL, linear function approximation has attracted a flurry of recent activity, mainly due to the promise of dramatic dimension reduction in conjunction with its mathematical tractability (see, e.g., Bertsekas and Tsitsiklis (1995); Du et al. (2020a); Jin et al. (2020); Wen

---

[*]Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544, USA.

[†]Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA.

[‡]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.

[§]Department of Statistics and Data Science, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104, USA.

and Van Roy (2017); Yang and Wang (2019) and the references therein). Two families of linear function approximation merit particular attention, which we single out below. Here and throughout, we concentrate on a finite-horizon Markov decision process (MDP), and denote by $\mathcal{S}$, $\mathcal{A}$ and H its state space, action space, and horizon, respectively.

- *Model-based linear representation.* Jin et al. (2020); Yang and Wang (2019, 2020) studied a stylized scenario when both the probability transition kernel and the reward function of the MDP can be linearly parameterized. This type of MDPs is commonly referred to as linear MDPs. Letting $P_h(\cdot \mid s, a)$ denote the transition probability from state s when action a is executed at time step h, the linear MDP model postulates the existence of a set of predetermined d-dimensional feature vectors $\{\varphi_h(s, a) \in \mathbb{R}^d\}$ and a collection of unknown parameter matrices $\{\mu_h \in \mathbb{R}^{d \times |\mathcal{S}|}\}$ such that

$$\forall (s, a) \in \mathcal{S} \times \mathcal{A} \text{ and } 1 \leq h \leq H : \qquad P_h(\cdot \mid s, a) = \big(\varphi_h(s, a)\big)^\top \mu_h. \tag{1}$$

  Similar assumptions are imposed on the reward function as well. In other words, the linear MDP model posits that the probability transition matrix is low-rank (with rank at most d) with the column space known *a priori*, which forms the basis for sample size saving in comparison to the unstructured setting.

- *Value-based linear realizability.* Rather than relying on linear embedding of the model specification (namely, the transition kernel and the reward function), another class of linear representation assumes that the action-value function (or Q-function) can be well predicted by linear combinations of known feature vectors $\{\varphi_h(s, a) \in \mathbb{R}^d\}$. A concrete framework of this kind assumes linear realizability of the optimal Q-function (denoted by $Q_h^\star$ at time step h from now on), that is, there exist some unknown vectors $\{\theta_h^\star \in \mathbb{R}^d\}$ such that

$$Q_h^\star(s, a) = \big\langle \varphi_h(s, a), \theta_h^\star \big\rangle \tag{2}$$

  holds for any state-action pair (s, a) and any step h. It is self-evident that this framework seeks to represent the optimal Q-function — which is an $|\mathcal{S}||\mathcal{A}|H$-dimensional object — via H parameter vectors each of dimension d. Throughout this work, an MDP obeying this condition is said to be *an MDP with linearly realizable optimal Q-function*, or more concisely, *an MDP with linear $Q^\star$*.

1.2 Sample size barriers with linearly realizable $Q^\star$

The current paper focuses on MDPs with linearly realizable optimal Q-function $Q^\star$. In some sense, this is arguably the weakest assumption one can hope for; that is, if linear realiability of $Q^\star$ does not hold, then a linear function approximation should perhaps not be adopted in the first place. In stark contrast to linear MDPs that allow for sample-efficient RL (in the sense that the sample complexity is almost independent of $|\mathcal{S}|$ and $|\mathcal{A}|$ but instead depends only polynomially on d, H and possibly some sub-optimality gap), MDPs with linear $Q^\star$ do not admit a similar level of sample efficiency in general. To facilitate discussion, we summarize key existing results for a couple of settings. Here and below, the notation $f(d) = \Omega(g(d))$ means that $f(d)$ is at least on the same order as $g(d)$ when d tends to infinity.

- *Sample inefficiency under a generative model.* Even when a generative model or a simulator is available — so that the learner can query arbitrary state-action pairs to draw samples from (Kearns and Singh, 1999) — one can find a hard MDP instance in this class that requires at least $\min\big\{\exp(\Omega(d)), \exp(\Omega(H))\big\}$ samples regardless of the algorithm in use (Weisz et al., 2021b).

- *Sample efficiency with a sub-optimality gap under a generative model.* The aforementioned sample size barrier can be alleviated if, for each state, there exists a sub-optimality gap $\Delta_{\sf gap}$ between the value under the optimal action and that under any sub-optimal action. As asserted by Du et al. (2020a, Appendix C), a sample size that scales polynomially in d, H and $1/\Delta_{\sf gap}$ is sufficient to identify the optimal policy, assuming access to the generative model.

- *Sample inefficiency with a large sub-optimality gap in online RL.* Turning to the standard episodic online RL setting (so that in each episode, the learner is given an initial state and executes the MDP for H steps to obtain a sample trajectory), sample-efficient algorithms are, much to our surprise,

2

infeasible even in the presence of a large sub-optimality gap. As has been formalized in Wang et al. (2021b), it is possible to construct a hard MDP instance with a constant sub-optimality gap that cannot be solved without at least $\min\big\{\exp(\Omega(d)), \exp(\Omega(H))\big\}$ samples.

In conclusion, the linear $Q^\star$ assumption, while succinctly capturing the low-dimensional structure, still presents an undesirable hurdle for RL. The sampling mechanism commonly studied in standard RL formulations precludes sample-efficient solutions even when a favorable sub-optimality gap comes into existence.

1.3 Our contributions

Having observed the exponential separation between the generative model and standard online RL when it comes to the linear $Q^\star$ problem, one might naturally wonder whether there exist practically relevant sampling mechanisms — more flexible than standard online RL yet more practical than the generative model — that promise substantial sample size reduction. This motivates the investigation of the current paper, as summarized below.

- *A new sampling protocol: sampling with state revisiting.* We investigate a flexible sampling protocol that is built upon classical online/exploratory formalism but allows for *state revisiting* (detailed in Algorithm 1). In each episode, the learner starts by running the MDP for H steps, and is then allowed to revisit any previously visited state and re-run the MDP from there. The learner is allowed to revisit states for an arbitrary number of times, although executing this feature too often might inevitably incur an overly large sampling burden. This sampling protocol accommodates several realistic scenarios; for instance, it captures the "save files" feature in video games that allows players to record player progress and resume from the save points later on. In addition, state revisiting is reminiscent of Monte Carlo Tree Search implemented in various real-world applications, which assumes that the learner can go back to father nodes (i.e., previous states) (Silver et al., 2016). This protocol is also referred to as *local access to the simulator* in the recent work Yin et al. (2021).

- *A sample-efficient algorithm.* Focusing on the above sampling protocol, we propose a value-based method — called LinQ-LSVI-UCB — adapted from the LSVI-UCB algorithm (Jin et al., 2020). The algorithm implements the optimism principle in the face of uncertainty, while harnessing the knowledge of the sub-optimality gap to determine whether to backtrack and revisit states. The proposed algorithm provably achieves a sample complexity that scales polynomially in the feature dimension d, the horizon H, and the inverse sub-optimality gap $1/\Delta_{\mathsf{gap}}$, but is otherwise independent of the size of the state space and the action space.

2 Model and assumptions

In this section, we present precise problem formulation, notation, as well as a couple of key assumptions. Here and throughout, we denote by $|\mathcal{S}|$ the cardinality of a set $\mathcal{S}$, and adopt the notation $[n] \coloneqq \{1, \cdots, n\}$.

2.1 Basics of Markov decision processes

Finite-horizon MDP. The focus of this paper is the setting of a finite-horizon MDP, as represented by the quintuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \{P_h\}_{h=1}^{H}, \{r_h\}_{h=1}^{H}, H)$ (Agarwal et al., 2019). Here, $\mathcal{S} \coloneqq \{1, \cdots, |\mathcal{S}|\}$ represents the state space, $\mathcal{A} \coloneqq \{1, \cdots, |\mathcal{A}|\}$ denotes the action space, H indicates the time horizon of the MDP, P_h stands for the probability transition kernel at time step $h \in [H]$ (namely, $P_h(\cdot \,|\, s, a)$ is the transition probability from state s upon execution of action a at step h), whereas $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$ represents the reward function at step h (namely, we denote by $r_h(s, a)$ the immediate reward received at step h when the current state is s and the current action is a). For simplicity, it is assumed throughout that all rewards $\{r_h(s, a)\}$ are deterministic and reside within the range $[0, 1]$. Note that our analysis can also be straightforwardly extended to accommodate random rewards, which we omit here for the sake of brevity.

Policy, value function, and Q-function. We let $\pi = \{\pi_h\}_{1 \le h \le H}$ represent a policy or action selection rule. For each time step h, π_h represents a deterministic mapping from $\mathcal{S}$ to $\mathcal{A}$, namely, action $\pi_h(s)$ is taken at step h if the current state is s. The value function associated with policy π at step h is then defined as the cumulative reward received between steps h and H under this policy, namely,

$$\forall (s, h) \in \mathcal{S} \times [H]: \qquad V_h^\pi(s) := \mathbb{E}\left[\sum_{t=h}^H r_t(s_t, a_t) \,\Big|\, s_t = s\right]. \tag{3}$$

Here, the expectation is taken over the randomness of an MDP trajectory $\{s_t\}_{t=h}^H$ induced by policy π (namely, $a_t = \pi_t(s_t)$ and $s_{t+1} \sim P(\cdot \,|\, s_t, a_t)$ for any $h \le t \le H$). Similarly, the action-value function (or Q-function) associated with policy π is defined as

$$\forall (s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]: \qquad Q_h^\pi(s, a) := \mathbb{E}\left[\sum_{t=h}^H r_t(s_t, a_t) \,\Big|\, s_h = s, a_h = a\right], \tag{4}$$

which resembles the definition (3) except that the action at step h is frozen to be a. Our normalized reward assumption (i.e., $r_h(s, a) \in [0, 1]$) immediately leads to the trivial bounds

$$\forall (s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]: \qquad 0 \le V_h^\pi(s) \le H \qquad \text{and} \qquad 0 \le Q_h^\pi(s, a) \le H. \tag{5}$$

A recurring goal in reinforcement learning is to search for a policy that maximizes the value function and the Q-function. For notational simplicity, we define the optimal value function $V^\star = \{V_h^\star\}_{1 \le h \le H}$ and optimal Q-function $Q^\star = \{Q_h^\star\}_{1 \le h \le H}$ respectively as follows

$$\forall (s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]: \qquad V_h^\star(s) := \max_\pi V_h^\pi(s) \qquad \text{and} \qquad Q_h^\star(s, a) := \max_\pi Q_h^\pi(s, a),$$

with the optimal policy (i.e., the one that maximizes the value function) represented by $\pi^\star = \{\pi_h^\star\}_{1 \le h \le H}$.

2.2 Key assumptions

Linear realizability of $Q^\star$. In order to enable significant reduction of sample complexity, it is crucial to exploit proper low-dimensional structure of the problem. This paper is built upon linear realizability of the optimal Q-function $Q^\star$ as follows.

Assumption 1. *Suppose that there exist a collection of pre-determined feature maps*

$$\varphi = (\varphi_h)_{1 \le h \le H}, \qquad \varphi_h : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d \tag{6}$$

and a set of unknown vectors $\theta_h^\star \in \mathbb{R}^d$ ($1 \le h \le H$) such that

$$\forall (s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]: \qquad Q_h^\star(s, a) = \langle \varphi_h(s, a), \theta_h^\star \rangle. \tag{7}$$

In addition, we assume that

$$\forall (s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]: \qquad \|\varphi_h(s, a)\|_2 \le 1 \quad \text{and} \quad \|\theta_h^\star\|_2 \le 2H\sqrt{d}. \tag{8}$$

In other words, we assume that $Q^\star = \{Q_h^\star\}_{1 \le h \le H}$ can be embedded into a d-dimensional subspace encoded by φ, with $d \le |\mathcal{S}||\mathcal{A}|$. In fact, we shall often view d as being substantially smaller than the ambient dimension $|\mathcal{S}||\mathcal{A}|$ in order to capture the dramatic degree of potential dimension reduction. It is noteworthy that linear realizability of $Q^\star$ in itself is a considerably weaker assumption compared to the one commonly assumed for linear MDPs (Jin et al., 2020) (which assumes $\{P_h\}_{1 \le h \le H}$ and $\{r_h\}_{1 \le h \le H}$ are all linearly parameterized). The latter necessarily implies the former, while in contrast the former by no means implies the latter. Additionally, we remark that the assumption (8) is compatible with what is commonly assumed for linear MDPs; see, e.g., Jin et al. (2020, Lemma B.1) for a reasoning about why this bound makes sense.

Sub-optimality gap. As alluded to previously, another metric that comes into play in our theoretical development is the sub-optimality gap. Specifically, for each state s and each time step h, we define the following metric

$$\Delta_h(s) := \min_{a \notin \mathcal{A}_s^\star} \left\{ V_h^\star(s) - Q_h^\star(s,a) \right\} \qquad \text{with } \mathcal{A}_s^\star := \left\{ a : Q_h^\star(s,a) = V_h^\star(s) \right\}. \tag{9}$$

In words, $\Delta_h(s)$ quantifies the gap — in terms of the resulting Q-values — between the optimal action and the sub-optimal ones. It is worth noting that there might exist multiple optimal actions for a given (s,h) pair, namely, the set $\mathcal{A}_s^\star$ is not necessarily a singleton. Further, we define the minimum gap over all (s,h) pairs as follows

$$\Delta_{\text{gap}} := \min_{s,h \in \mathcal{S} \times [H]} \Delta_h(s), \tag{10}$$

and refer to it as the sub-optimality gap throughout this paper.

2.3 RL under sampling with state revisiting

In standard online episodic RL settings, the learner collects data samples by executing multiple length-H trajectories in the MDP $\mathcal{M}$ via suitably chosen policies; more concretely, in the n-th episode with a given initial state s_0^n, the agent executes a policy to generate a sample trajectory $\{(s_h^n, a_h^n)\}_{1 \leq h \leq H}$, where (s_h^n, a_h^n) denotes the state-action pair at time step h. This setting underscores the importance of trading off exploitation and exploration. As pointed out previously, however, this classical sampling mechanism could be highly inefficient for MDPs with linearly realizable $Q^\star$, even in the face of a constant sub-optimality gap (Wang et al., 2021b).

A new sampling protocol with state revisiting. In order to circumvent this sample complexity barrier, the current paper studies a more flexible sampling mechanism that allows one to revisit previous states in the same episode. Concretely, in each episode, the sampling process can be carried out in the following fashion:

Algorithm 1: Sampling protocol for an episode with state revisiting.

---
1 Input: initial state s_1.
2 Select a policy and sample a length-H trajectory $\{(s_t, a_t)\}_{1 \leq t \leq H}$.
3 repeat
4   Pick any previously visited state s_h in this episode;
5   Execute a new trajectory starting from s_h all the way up to step H, namely, $\{(s_t, a_t)\}_{h \leq t \leq H}$;
     here, we overload notation to simplify presentation.
6 until *the learner terminates it.*

As a distinguishing feature, the sampling mechanism described in Algorithm 1 allows one to revisit previous states and retake samples from there, which reveals more information regarding these states. To make apparent its practice relevance, we first note that the generative model proposed in Kakade (2003); Kearns and Singh (1999) — in which one can query a simulator with arbitrary state-action pairs to get samples — is trivially subsumed as a special case of this sampling mechanism. Moving on to a more complicated yet realistic scenario, consider role-playing video games which commonly include built-in "save files" features. This type of features allows the player to record its progress at any given point, so that it can resume the game from this save point later on. In fact, rebooting the game multiple times from a saved point allows an RL algorithm to conduct trial-and-error learning for this particular game point.

As a worthy note, while revisiting a particular state many times certainly yields information gain about this state, it also means that fewer samples can be allocated to other episodes if the total sampling budget is fixed. Consequently, how to design intelligent state revisiting schemes in order to optimize sample efficiency requires careful thinking.

Learning protocol and sample efficiency. We are now ready to describe the learning process — which consists of N episodes — and our goal.

- In the n-th episode ($1 \leq n \leq N$), the learner is given an initial state $s_1^{(n)}$ (assigned by nature), and executes the sampling protocol in Algorithm 1 until this episode is terminated.

- At the end of the n-th episode, the outcome of the learning process takes the form of a policy $\pi^{(n)}$, which is learned based on all information collected up to the end of this episode.

The quality of the learning outcome $\{\pi^{(n)}\}_{1 \leq n \leq N}$ is then measured by the cumulative regret over N episodes as follows:

$$\mathsf{Regret}(N) := \sum_{n=1}^{N} \left(V_1^{\star}\big(s_1^{(n)}\big) - V_1^{\pi^{(n)}}\big(s_1^{(n)}\big) \right), \tag{11}$$

which is what we aim to minimize under a given sampling budget. More specifically, for any target level $\varepsilon \in [0, H]$, the aim is to achieve

$$\frac{1}{N} \mathsf{Regret}(N) \leq \varepsilon$$

regardless of the initial states (which are chosen by nature), using a sample size T no larger than $\mathsf{poly}\big(d, H, \frac{1}{\varepsilon}, \frac{1}{\Delta_{\mathsf{gap}}}\big)$ (but independent of $|\mathcal{S}|$ and $|\mathcal{A}|$). Here and throughout, T stands for the total number of samples observed in the learning process; for instance, a new trajectory $\{(s_t, a_t)\}_{h \leq t \leq H}$ amounts to $H - h$ new samples. Due to the presence of state revisiting, there is a difference between our notions of regret / sample complexity and the ones used in standard online RL, which we shall elaborate on in the next section. An RL algorithm capable of achieving this level of sample complexity is declared to be sample-efficient, given that the sample complexity does not scale with the ambient dimension of the problem (which could be enormous in contemporary RL).

Remark 1 (From average regret to PAC guarantees and optimal policies.). There is some intimate connection between regret bounds and PAC guarantees that has been pointed out previously (e.g., Jin et al. (2018)). For instance, by fixing the initial state distribution to be identical (e.g., $s_1^{(n)} = s$ for all $1 \leq n \leq N$) and choosing the output policy $\widehat{\pi}$ uniformly at random from $\{\pi^{(n)} \mid 1 \leq n \leq N\}$, one can easily verify that this output policy $\widehat{\pi}$ is ε-optimal for state s, as long as $\frac{1}{N} \mathsf{Regret}(N) \leq \varepsilon$.

3 Algorithm and main results

In this section, we put forward an algorithm tailored to the sampling protocol described in Algorithm 1, and demonstrate its desired sample efficiency.

3.1 Algorithm

Our algorithm design is motivated by the method proposed in (Jin et al., 2020) for linear MDPs — called *least-squares value iteration with upper confidence bounds (LSVI-UCB)* — which follows the principle of "optimism in the face of uncertainty". In what follows, we shall begin by briefly reviewing the key update rules of LSVI-UCB, and then discuss how to adapt it to accommodate MDPs with linearly realizable $Q^{\star}$ when state revisiting is permitted.

Review: LSVI-UCB for linear MDPs. Let us remind the readers of the setting of linear MDPs. It is assumed that there exist unknown vectors $\mu_h(\cdot) = [\mu_h^{(1)}, \cdots, \mu_h^{(d)}]^{\top} \in \mathbb{R}^{d \times |\mathcal{S}|}$ and $w_h \in \mathbb{R}^d$ such that

$$\forall (s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H] : \qquad P_h(\cdot \mid s, a) = \big\langle \varphi(s, a), \mu_h(\cdot) \big\rangle \quad \text{and} \quad r_h(s, a) = \big\langle \varphi(s, a), w_h(s, a) \big\rangle.$$

In other words, both the probability transition kernel and the reward function can be linearly represented using the set of feature maps $\{\varphi(s, a)\}$.

LSVI-UCB can be viewed as a generalization of the UCBVI algorithm (Azar et al., 2017) (originally proposed for the tabular setting) to accommodate linear function approximation. In each episode, the learner draws a sample trajectory following the greedy policy w.r.t. the current Q-function estimate with UCB exploration; namely, an MDP trajectory $\{(s_h^n, a_h^n)\}_{1 \leq h \leq H}$ is observed in the n-th episode. Working

backwards (namely, going from step H all the way back to step 1), the LSVI-UCB algorithm in the n-th episode consists of the following key updates:

$$\Lambda_h \;\; \leftarrow \sum_{i=1}^{n} \varphi(s_h^i, a_h^i)\varphi(s_h^i, a_h^i)^\top + \lambda I, \tag{12a}$$

$$\theta_h \;\; \leftarrow \Lambda_h^{-1} \sum_{i=1}^{n} \varphi(s_h^i, a_h^i)\left\{ r_h(s_h^i, a_h^i) + \max_a Q_{h+1}(s_{h+1}^i, a) \right\}, \tag{12b}$$

$$Q_h(\cdot, \cdot) \;\; \leftarrow \min\left\{ \langle \theta_h, \varphi(\cdot, \cdot) \rangle + \beta\sqrt{\varphi(\cdot, \cdot)^\top \Lambda_h^{-1} \varphi(\cdot, \cdot)}, H \right\}, \tag{12c}$$

with the regularization parameter λ set to be 1. Informally speaking, θ_h (cf. (12b)) corresponds to the solution to a ridge-regularized least-squares problem — tailored to solving the Bellman optimality equation with linear parameterization — using all samples collected so far for step h, whereas the matrix Λ_h (cf. (12a)) captures the (properly regularized) covariance of $\varphi(\cdot, \cdot)$ associated with these samples. In particular, $\langle \theta_h, \varphi(\cdot, \cdot) \rangle$ attempts to estimate the Q-function by exploiting its linear representation for this setting, and the algorithm augments it by an upper confidence bound (UCB) bonus $\beta\sqrt{\varphi(\cdot, \cdot)^\top \Lambda_h^{-1} \varphi(\cdot, \cdot)}$ — a term commonly arising in the linear bandit literature (Lattimore and Szepesvári, 2020) — to promote exploration, where β is a hyper-parameter to control the level of exploration. As a minor remark, the update rule (12c) also ensures that the Q-function estimate never exceeds the trivial upper bound H.

Our algorithm: LinQ-LSVI-UCB for linearly realizable $Q^\star$. Moving from linear MDPs to MDPs with linear $Q^\star$, we need to make proper modification of the algorithm. To facilitate discussion, let us introduce some helpful concepts.

- Whenever we start a new episode or revisit a state (and draw samples thereafter), we say that *a new path* is being collected. The total number of paths we have collected is denoted by K.

- For each k and each step h, we define a set of indices

$$\mathcal{I}_h^k := \left\{ i : \; 1 \le i \le k \mid \theta_h^i \text{ is updated in the } i\text{-th path at time step } h \right\}, \tag{13}$$

  which will also be described precisely in Algorithm 2. As we shall see, the cardinality of $\mathcal{I}_h^k$ is equal to the total number of new samples that have been collected at time step h up to the k-th path.

We are now ready to describe the proposed algorithm. For the k-th path, our algorithm proceeds as follows.

- *Sampling.* Suppose that we start from a state s_h^k at time step h. The learner adopts the greedy policy $\pi^k = \{\pi_j^k\}_{h \le j \le H}$ in accordance with the current Q-estimate $\{Q_j^{k-1}\}_{h \le j \le H}$, and observes a fresh sample trajectory $\{(s_j^k, a_j^k)\}_{h \le j \le H}$ as follows: for $j = h, h+1, \ldots, H$,

$$s_{j+1}^k \sim P_j(\cdot \mid s_j^k, a_j^k) \qquad \text{with} \qquad a_j^k = \pi_j^k(s_j^k) := \arg\max_a Q_j^{k-1}(s_j^k, a). \tag{14}$$

- *Backtrack and update estimates.* We then work backwards to update our Q-estimates and the θ-estimates (i.e., estimates for the linear representation of $Q^\star$), until the UCB bonus term (which reflects the estimated uncertainty level of the Q-estimate) drops below a threshold determined by the sub-optimality gap Δ_{gap}. More precisely, working backwards from $h = H$, we carry out the following calculations if certain conditions (to be described shortly) are met:

$$\Lambda_h^k \;\; \leftarrow \sum_{i \in \mathcal{I}_h^k} \varphi_h(s_h^i, a_h^i)\varphi_h(s_h^i, a_h^i)^\top + I, \tag{15a}$$

$$\theta_h^k \;\; \leftarrow \left(\Lambda_h^k\right)^{-1} \sum_{i \in \mathcal{I}_h^k} \varphi_h(s_h^i, a_h^i)\left\{ r_h(s_h^i, a_h^i) + \left\langle \varphi_{h+1}(s_{h+1}^i, a_{h+1}^i), \theta_{h+1}^k \right\rangle \right\}, \tag{15b}$$

Algorithm 2: LinQ-LSVI-UCB with state revisiting.
---
1 inputs: number of episodes N, sub-optimality gap Δ_{gap}, initial states $\{s_1^{(n)}\}_{1 \le n \le N}$.
2 initialization: $h = 0$, $n = 0$, $\theta_j^0 = 0$ and $\mathcal{I}_j^0 = \emptyset$ for all $1 \le j \le H$.
3 for $k = 1, 2, \cdots$ **do**
4 $\quad$ call $\pi^k \leftarrow$ `get-policy()`.
5 $\quad$ $h \leftarrow h + 1$.
6 $\quad$ **if** $h = 1$ **then**
7 $\quad\quad$ $\pi^{(n)} \leftarrow \pi^{k-1}$. // record the up-to-date policy learned in this episode.
8 $\quad\quad$ $n \leftarrow n + 1$. // start a new episode.
9 $\quad\quad$ **if** $n > N$ **then**
10 $\quad\quad\quad$ $K \leftarrow k - 1$ and **return**. // terminate after N episodes.
11 $\quad\quad$ Set the initial state $s_1^k = s_1^{(n)}$.
12 $\quad$ call `sampling()`. // collect new samples from step h; see Algorithm 3.
$\quad\quad$ /* backtrack and determine whether to revisit a state and redraw new samples.
$\quad\quad$ */
13 $\quad$ Set $\theta_{H+1}^k = 0$ and $h = H$.
14 $\quad$ **while** $h > 0$ *and* $b_{h+1}^{k-1}(s_{h+1}^k, a_{h+1}^k) < \Delta_{\text{gap}}/2$ *(cf. (15c))* **do**
15 $\quad\quad$ $\mathcal{I}_h^k \leftarrow \mathcal{I}_h^{k-1} \bigcup \{k\}$. // expand $\mathcal{I}_h^k$ whenever we need to update θ_h^k.
16 $\quad\quad$ Update θ_h^k according to (15b).
17 $\quad\quad$ $h \leftarrow h - 1$.
18 $\quad$ call `update-remaining()`. // keep remaining iterates unchanged; see Algorithm 3.

$$b_h^k(\cdot, \cdot) \; \leftarrow \beta \sqrt{\varphi_h(\cdot, \cdot)^\top \left(\Lambda_h^k\right)^{-1} \varphi_h(\cdot, \cdot)}, \tag{15c}$$

$$Q_h^k(\cdot, \cdot) \; \leftarrow \min\left\{ \langle \varphi_h(\cdot, \cdot), \theta_h^k \rangle + b_h^k(\cdot, \cdot), \, H \right\}. \tag{15d}$$

Here, we employ the pre-factor

$$\beta = c_\beta \sqrt{dH^4 \log \frac{KH}{\delta}} \tag{16}$$

to adjust the level of "optimism", where $c_\beta > 0$ is taken to be some suitably large constant. Crucially, whether the update (15b) — and hence (15d) — is executed depends on the size of the bonus term b_{h+1}^{k-1} of the last attempt at step $h + 1$. Informally, if the bonus term is sufficiently small compared to the sub-optimality gap, then we have confidence that the policy estimate (after time step h) can be trusted in the sense that it is guaranteed to generalize and perform well on unseen states.

The complete algorithm is summarized in Algorithm 2, with some basic auxiliary functions provided in Algorithm 3. To facilitate understanding, an illustration is provided in Figure 1.

Two immediate remarks are in order. In comparison to LSVI-UCB in Jin et al. (2020), the update rule (15b) for θ_h^k employs the linear representation $\langle \varphi_{h+1}(s_{h+1}^i, a_{h+1}^i), \theta_{h+1}^k \rangle$ *without the UCB bonus* as the Q-value estimate. This subtle difference turns out to be important in the analysis for MDPs with linear $Q^\star$. In addition, LSVI-UCB is equivalent to first obtaining a linear representation of transition kernel P (Agarwal et al., 2019) and then using it to build Q-function estimates and draw samples. In contrast, Algorithm 2 cannot be interpreted as a decoupling of model estimation and planning/exploration stage, and is intrinsically a value-based approach.

3.2 Theoretical guarantees

Equipped with the precise description of Algorithm 2, we are now positioned to present its regret bound and sample complexity analysis. Our result is this:

Algorithm 3: Simple auxiliary functions.

---

1 Function `sampling()`:

    `/* sampling from the beginning (if` $h = 0$`) or from a revisited state (if` $h > 0$`).`
    `*/`

    `/* do not update samples prior to step` h`.`   `*/`

2   **for** $j = 1, 2, \ldots, h-1$ **do**

3   Set $a_j^k = a_j^{k-1}$, and $s_{j+1}^k = s_{j+1}^{k-1}$.

    `/* a new round of sampling from step` h`.`   `*/`

4   **for** $j = h, h+1, \cdots, H$ **do**

5   Compute $Q_j^{k-1}(s_j^k, a)$ according to (15d).

6   Take $a_j^k = \arg\max_a Q_j^{k-1}(s_j^k, a)$, and draw $s_{j+1}^k \sim P_j(\cdot \mid s_j^k, a_j^k)$.

7 Function `update-remaining()`:

    `/* keep the estimates prior to step` h `unchanged.`   `*/`

8   **for** $j = 1, 2, \ldots, h$ **do**

9   Set $\theta_j^k = \theta_j^{k-1}$.

10 Function `get-policy()`:

    `/* update the Q-estimates.`   `*/`

11   **for** $1 \le h \le H$ **do**

12   Set $Q_h^{k-1}(\cdot, \cdot)$ according to (15d).

13 $\pi_h^k(\cdot) \leftarrow \arg\max_a Q_h^{k-1}(\cdot, a)$.

Theorem 1. *Suppose that Assumption 1 holds, and that $c_\beta \ge 8$ is some fixed constant. Then for any $0 < \delta < 1$, and any initial states $\left\{ s_1^{(n)} \right\}_{1 \le n \le N}$, Algorithm 2 achieves a regret (see (11)) obeying*

$$\frac{1}{N}\mathsf{Regret}(N) \le 8c_\beta \sqrt{\frac{d^2 H^7 \log^2 \frac{HT}{\delta}}{T}} \tag{17}$$

with probability at least $1 - \delta$, provided that $N \ge \frac{4c_\beta^2 d^2 H^5 \log^2 \frac{HT}{\delta}}{\Delta_{\mathsf{gap}}^2}$. In addition, the total number of state revisits satisfies

$$K - N \le \frac{4c_\beta^2 d^2 H^5 \log^2 \frac{KH}{\delta}}{\Delta_{\mathsf{gap}}^2}. \tag{18}$$

Theorem 1 characterizes the sample efficiency of the proposed algorithm. Specifically, the theorem implies that for any $0 < \varepsilon < H$, the average regret satisfies

$$\frac{1}{N}\mathsf{Regret}(N) \le \varepsilon$$

with probability exceeding $1 - \delta$, once the total number T of samples exceeds

$$T \ge \frac{64c_\beta^2 d^2 H^7 \log^2 \frac{HT}{\delta}}{\varepsilon^2}. \tag{19}$$

Several implications and further discussions of this result are in order.

Efficiency of the proposed algorithm. We first highlight several benefits of the proposed algorithm.

- *Sample efficiency.* While our sample complexity bound (19) scales as a polynomial function of both d and H, it does not rely on either the state space size $|\mathcal{S}|$ or the action space size $|\mathcal{A}|$. This hints at the dramatic sample size reduction when $|\mathcal{S}||\mathcal{A}|$ far exceeds the feature dimension d and the horizon H.
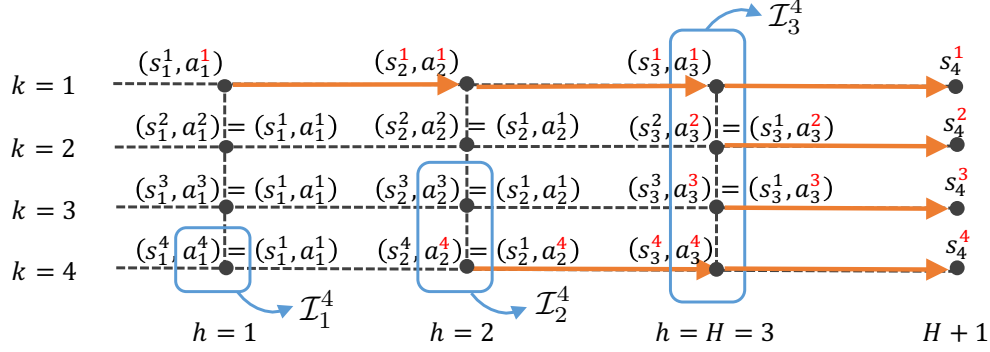
Figure 1: Illustration of the set $\mathcal{I}_h^K$ for a simple scenario where $N = 1$, $H = 3$, and the total number of paths is $K = 4$. After the 1st path, the sampling process revisits state s_3^1 twice (each time drawing one new sample), and then revisits state s_2^1 to draw two samples from there. This episode then terminates as the conditions are met for all steps. A red superscript indicates new state or new actions taken, and the orange line illustrates the sampling process. Here, we have $\mathcal{I}_3^4 = \{1, 2, 3, 4\}$, $\mathcal{I}_2^4 = \{3, 4\}$ and $\mathcal{I}_1^4 = \{4\}$, which record the paths where the linear representations are updated for each respective step.

- *A small number of state revisits.* Theorem 1 develops an upper bound (18) on the total number of state revisits, which is gap-dependent but otherwise independent of the target accuracy level ε. As a consequence, as the sample size T increases (or when ε decreases), the ratio of the number of state revisits to the sample size becomes vanishingly small, meaning that the true sampling process in our algorithm becomes increasingly closer to the standard online RL setting.

- *Computational complexity and memory complexity.* The computational bottleneck of the proposed algorithm lies in the update of θ_h^k and b_h^k (see (15b) and (15c), respectively), which consists of solving a linear systems of equations and can be accomplished using, say, the conjugate gradient method in time on the order of d^2 (up to logarithmic factor). In addition, one needs to search over all actions when drawing samples, so the algorithm necessarily depends on $|\mathcal{A}|$. In total, the algorithm has a runtime no larger than $\widetilde{O}(d^2|\mathcal{A}|T)$. In addition, implementing our algorithm requires $O(d^2 H)$ units of memory.

Cumulative regret over paths. Due to the introduction of state revisiting, there are two possible ways to accumulate regrets: over the episodes or over the paths. While our analysis so far adopts the former (see (11)), it is not difficult to translate our regret bound over the episodes to the one over the paths. To be more precise, let us denote the regret over paths as follows for distinguishing purposes:

$$\mathsf{Regret}_{\mathsf{path}}(K) := \sum_{k=1}^{K} \left(V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k) \right). \tag{20}$$

A close inspection of our analysis readily reveals the following regret upper bound

$$\mathsf{Regret}_{\mathsf{path}}(K) \leq 4c_\beta \sqrt{d^2 H^6 K \log^2 \frac{HT}{\delta}} + \frac{4c_\beta^2 d^2 H^6 \log^2 \frac{KH}{\delta}}{\Delta_{\mathsf{gap}}^2} \tag{21a}$$

with probability exceeding $1 - \delta$; see Section 5.2 for details. This bound confirms that the regret over the paths exhibits a scaling of at most $\sqrt{K}$.

Logarithmic regret. As it turns out, our analysis further leads to a significantly strengthened upper bound on the expected regret. As we shall solidify in Section 5.2, the regret incurred by our algorithm satisfies the following upper bound

$$\mathbb{E}\big[\mathsf{Regret}(N)\big] \leq \mathbb{E}\big[\mathsf{Regret}_{\mathsf{path}}(K)\big] \leq \frac{17c_\beta^2 d^2 H^7 \log^2(KH)}{\Delta_{\mathsf{gap}}^2}, \tag{21b}$$

10

largely owing to the presence of the gap assumption. This implies that the expected regret scales only logarithmically in the number of paths K, which could often be much smaller than the previous bound (21a). In fact, this is consistent with the recent literature regarding logarithmic regrets under suitable gap assumptions (e.g., Simchowitz and Jamieson (2019); Yang et al. (2021)).

Comparison to the case with a generative model. We find it helpful to compare our findings with the algorithm developed in the presence of a generative model. In a nutshell, the algorithm described in Du et al. (2020a, Appendix C) starts by identifying a "well-behaved" basis of the feature vectors, and then queries the generative model to sample the state-action pairs related to this basis. In contrast, our sampling protocol (cf. Algorithm 1) is substantially more restrictive and does not give us the freedom to sample such a basis. In fact, our algorithm is exploratory in nature, which is more challenging to analyze than the case with a generative model.

We shall also take a moment to point out a key technical difference between our approach and the algorithm put forward in Du et al. (2020a). A key insight in Du et al. (2020a) is that: by sampling each *anchor* state-action pair for $\mathsf{poly}(1/\Delta_{\mathsf{gap}})$ times, one can guarantee sufficiently accurate Q-estimates in all state-action pairs, which in turn ensures $\pi_k = \pi^\star$ for all state-action pairs in all future estimates. This, however, is not guaranteed in our algorithm when it comes to the state revisiting setting. Fortunately, the gap condition helps ensure that there are at most $\mathsf{poly}(1/\Delta_{\mathsf{gap}})$ number of samples such that $\pi_k \neq \pi^\star$, although the discrepancy might happen at any time throughout the execution of the algorithm (rather than only happening at the beginning). In addition, careful use of state revisiting helps avoid these sub-optimal estimates by resetting for at most $\mathsf{poly}(1/\Delta_{\mathsf{gap}})$ times, which effectively prevents error blowup.

Comparison to prior works in the presence of state revisiting. Upon closer examination, the sampling mechanism of Weisz et al. (2021a) considers another kind of state revisiting strategy and turns out to be quite similar to ours, which accesses a batch of samples $\{(s_h, a_h, s_{h+1}^i)\}_{i \geq 1}$ for the current state s_h with all actions $a_h \in \mathcal{A}$. Assuming only $V^\star$ is linearly realizable, their sample complexity is on the order of $(dH)^{|\mathcal{A}|}$, and hence its sample efficiency depends highly on the condition that $\mathcal{A} = O(1)$. Additionally, Du et al. (2020b) proposed an algorithm — tailored to a setting with deterministic transitions — that requires sampling each visited state multiple times (and hence can be accomplished when state revisiting is permitted); this algorithm might be extendable to accommodate stochastic transitions.

4 Additional related works

Non-asymptotic sample complexity guarantees for RL algorithms have been studied extensively in the tabular setting over recent years, e.g., Agarwal et al. (2020b,c); Azar et al. (2013, 2017); Cen et al. (2020); Dann and Brunskill (2015); Even-Dar and Mansour (2003); Jaksch et al. (2010); Li et al. (2021a,b, 2020a,b); Osband et al. (2016); Sidford et al. (2018); Wainwright (2019); Yang et al. (2021); Zhang et al. (2020), which have been, to a large extent, well-understood. The sample complexity typically scales at least linearly with respect to the state space size $|\mathcal{S}|$ and the action space size $|\mathcal{A}|$, and therefore, falls short of being sample-efficient when the state/action space is of astronomical size. In contrast, theoretical investigation of RL with function approximations is still in its infancy due to the complicated interaction between the dynamic of the MDP with the function class. In fact, theoretical support remains highly inadequate even when it comes to linear function approximation. For example, plain Q-learning algorithms coupled with linear approximation might easily diverge (Baird, 1995). It is thus of paramount interest to investigate how to design algorithms that can efficiently exploit the low-dimensional structure without compromising learning accuracy. In what follows, we shall discuss some of the most relevant results to ours. The reader is also referred to the summaries of recent literature in Du et al. (2021, 2020a).

Linear MDP. Jin et al. (2020); Yang and Wang (2019) proposed the linear MDP model, which can be regarded as a generalization of the linear bandit model (Abbasi-Yadkori et al., 2011; Dimakopoulou et al., 2019) and has attracted enormous recent activity (see e.g., Du et al. (2020a); Hao et al. (2020); He et al. (2020); Touati and Vincent (2020); Wang et al. (2021a, 2020a, 2019); Wei et al. (2021); Yang and Wang (2020); Zanette et al. (2020a) and the references therein). The results have further been generalized to

scenarios with much larger feature dimension by exploiting proper kernel function approximation (Long and Han, 2021; Yang et al., 2020).

From completeness to realizability. Du et al. (2020a) considered the policy completeness assumption, which assumes that the Q-functions of *all policies* reside within a function class that contains all functions that are linearly representable in a known low-dimensional feature space. In particular, Du et al. (2020a); Lattimore et al. (2020) examined how the model misspecification error propagates and impacts the sample efficiency of policy learning. A related line of works assumed that the linear function class is closed or has low approximation error under the Bellman operator, referred to as low inherent Bellman error (Munos, 2005; Shariff and Szepesvári, 2020; Zanette et al., 2019, 2020b).

These assumptions remain much stronger than the *realizability* assumption considered herein, where only the optimal Q-function $Q^\star$ is assumed to be linearly representable. Du et al. (2020b); Wen and Van Roy (2017) showed that sample-efficient RL is feasible in deterministic systems, which has been extended to stochastic systems with low variance in Du et al. (2019) under additional gap assumptions. In addition, Weisz et al. (2021b) established exponential sample complexity lower bounds under the generative model when only $Q^\star$ is linearly realizable; their construction critically relied on making the action set exponentially large. When restricted to a constant-size action space, Weisz et al. (2021a) provided a sample-efficient algorithm when only $V^\star$ is linearly realizable, where their sampling protocol essentially matches ours. Recently, Du et al. (2021) introduced the bilinear class and proposed sample-efficient algorithms when both $V^\star$ and $Q^\star$ are linearly realizable in the online setting.

Beyond linear function approximation. Moving beyond linear function approximation, another line of works (Ayoub et al., 2020; Zhou et al., 2020) investigated mixtures of linear MDPs. Moreover, additional efforts have been dedicated to studying the low-rank MDP model (without knowing *a priori* the feature space), which aims to learn the low-dimensional features as part of the RL problem; partial examples include Agarwal et al. (2020a); Modi et al. (2021). We conclude by mentioning in passing other attempts in identifying tractable families of MDPs with structural assumptions, such as Jiang et al. (2017); Jin et al. (2021); Osband and Van Roy (2014); Wang et al. (2020b).

5 Analysis

Before proceeding, we introduce several convenient notation to be used throughout the proof. As before, the total number of paths that have been sampled is denoted by K. For any $(s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$, we abbreviate

$$P_{h,s,a} := P_h(\cdot \mid s, a) \in \mathbb{R}^{|\mathcal{S}|}. \tag{22}$$

For any time step h in the k-th path, we define the empirical distribution vector $P_h^k \in \mathbb{R}^{|\mathcal{S}|}$ such that

$$P_h^k(s) := \begin{cases} 1, & \text{if } s = s_{h+1}^k; \\ 0, & \text{if } s \neq s_{h+1}^k. \end{cases} \tag{23}$$

The value function estimate $V_h^k : \mathcal{S} \to \mathbb{R}$ at time step h after observing the k-th path is defined as

$$\forall (s, h, k) \in \mathcal{S} \times [H] \times [K]: \qquad V_h^k(s) := \max_{a \in \mathcal{A}} Q_h^k(s, a), \tag{24}$$

where the iterate Q_h^k is defined in (15d).

Further, we remind the reader the crucial notation $\mathcal{I}_h^k$ introduced in (13), which represents the set of paths between the 1st and the k-th paths that update the estimate of $\theta_h^\star$. We have the following basic facts.

Lemma 1. *For all $1 \leq k \leq K$ and $1 \leq h \leq H$, one has*

$$\mathcal{I}_h^k \subseteq \mathcal{I}_{h+1}^k. \tag{25}$$

In addition,

$$\left| \mathcal{I}_1^K \right| = N \qquad and \qquad \left| \mathcal{I}_H^K \right| = K. \tag{26}$$

Proof. This lemma is somewhat self-evident from our construction, and hence we only provide brief explanation. The first claim (25) holds true since if θ_h^i is updated in the i-th path, then θ_j^i ($j \geq h$) must also be updated. The second claim (26) arises immediately from the definition of N (i.e., the number of episodes) and K (i.e., the number of paths). $\qquad\square$

5.1 Main steps for proving Theorem 1

In order to bound the regret for our proposed estimate, we first make note of an elementary relation that follows immediately from our construction:

$$\sum_{n=1}^{N} \left(V_1^\star(s_1^{(n)}) - V_1^{\pi^{(n)}}(s_1^{(n)}) \right) = \sum_{k \in \mathcal{I}_1^K} \left(V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k) \right), \tag{27}$$

where we recall the definition of $\mathcal{I}_1^K$ in (13). It thus comes down to bounding the right-hand side of (27).

Step 1: showing that Q_h^k is an optimistic view of $Q_h^\star$. Before proceeding, let us first develop a sandwich bound pertaining to the estimate error of the estimate Q_h^k delivered by Algorithm 2. The proof of this result is postponed to Section A.1.

Lemma 2. *Suppose that $c_\beta \geq 8$. With probability at least $1 - \delta$, the following bound*

$$0 \leq Q_h^k(s, a) - Q_h^\star(s, a) \leq 2b_h^k(s, a) \tag{28}$$

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

In words, this lemma makes apparent that Q_h^k is an over-estimate of $Q_h^\star$, with the estimation error dominated by the UCB bonus term b_h^k. This lemma forms the basis of the optimism principle.

Step 2: bounding the term on the right-hand side of (27). To control the difference $V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k)$ for each k, we establish the following two properties. First, combining Lemma 2 with the definition (24) (i.e., $V_h^{k-1}(s_h^k) = \max_a Q_h^{k-1}(s_h^k, a)$), one can easily see that

$$V_h^{k-1}(s_h^k) \geq Q_h^{k-1}\big(s_h^k, \pi^\star(s_h^k, h)\big) \geq Q_h^\star\big(s_h^k, \pi_h^\star(s_h^k)\big) = V_h^\star(s_h^k), \tag{29}$$

namely, V_h^{k-1} is an over-estimate of $V_h^\star$. In addition, from the definition $a_h^k = \pi_h^k(s_h^k) = \arg\max_a Q_h^{k-1}(s_h^k, a)$, one can decompose the difference $V_h^{k-1}(s_h^k) - V_h^{\pi^k}(s_h^k)$ as follows

$$\begin{aligned}
V_h^{k-1}(s_h^k) - V_h^{\pi^k}(s_h^k) &= Q_h^{k-1}(s_h^k, a_h^k) - Q_h^{\pi^k}(s_h^k, a_h^k) \\
&= Q_h^{k-1}(s_h^k, a_h^k) - Q_h^\star(s_h^k, a_h^k) + Q_h^\star(s_h^k, a_h^k) - Q_h^{\pi^k}(s_h^k, a_h^k) \\
&= Q_h^{k-1}(s_h^k, a_h^k) - Q_h^\star(s_h^k, a_h^k) + P_{h, s_h^k, a_h^k}(V_{h+1}^\star - V_{h+1}^{\pi^k}) \\
&= Q_h^{k-1}(s_h^k, a_h^k) - Q_h^\star(s_h^k, a_h^k) + \big(P_{h, s_h^k, a_h^k} - P_h^k\big)(V_{h+1}^\star - V_{h+1}^{\pi^k}) + V_{h+1}^\star(s_{h+1}^k) - V_{h+1}^{\pi^k}(s_{h+1}^k), \tag{30}
\end{aligned}$$

where the third line invokes Bellman equation $Q^\pi(s, a) = r(s, a) + P_{h, s, a}V^\pi$ for any π, and the last line makes use of the notation (23). Combining the above two properties leads to

$$\begin{aligned}
\sum_{k \in \mathcal{I}_1^K} \left[V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k) \right] &\leq \sum_{k \in \mathcal{I}_1^K} \left[V_1^{k-1}(s_1^k) - V_1^{\pi^k}(s_1^k) \right] \\
&= \sum_{k \in \mathcal{I}_1^K} \left[V_2^\star(s_2^k) - V_2^{\pi^k}(s_2^k) \right] + \sum_{k \in \mathcal{I}_1^K} \left[Q_1^{k-1}(s_1^k, a_1^k) - Q_1^\star(s_1^k, a_1^k) + (P_{1, s_1^k, a_1^k} - P_1^k)(V_2^\star - V_2^{\pi^k}) \right] \\
&\leq \sum_{k \in \mathcal{I}_2^K} \left[V_2^\star(s_2^k) - V_2^{\pi^k}(s_2^k) \right] + \sum_{k \in \mathcal{I}_1^K} \left[Q_1^{k-1}(s_1^k, a_1^k) - Q_1^\star(s_1^k, a_1^k) + (P_{1, s_1^k, a_1^k} - P_1^k)(V_2^\star - V_2^{\pi^k}) \right],
\end{aligned}$$

13

where the last line comes from the observation that $\mathcal{I}_h^K \subseteq \mathcal{I}_{h+1}^K$ (see Lemma 1). Applying the above relation recursively and using the fact that $V_{H+1}^\pi = 0$ for any π, we see that with probability at least $1 - \delta$,

$$\sum_{k \in \mathcal{I}_1^K} \left[V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k) \right] \leq \sum_{h=1}^H \sum_{k \in \mathcal{I}_h^K} \left[Q_h^{k-1}(s_h^k, a_h^k) - Q_h^\star(s_h^k, a_h^k) + (P_{h, s_h^k, a_h^k} - P_h^k)(V_{h+1}^\star - V_{h+1}^{\pi^k}) \right]$$

$$\leq \sum_{h=1}^H \sum_{k \in \mathcal{I}_h^K} \left[2b_h^{k-1}(s_h^k, a_h^k) + (P_{h, s_h^k, a_h^k} - P_h^k)(V_{h+1}^\star - V_{h+1}^{\pi^k}) \right]$$

$$= \underbrace{\sum_{h=1}^H \sum_{k \in \mathcal{I}_h^K} 2b_h^{k-1}(s_h^k, a_h^k)}_{=: \xi_1} + \underbrace{\sum_{h=1}^H \sum_{k \in \mathcal{I}_h^K} (P_{h, s_h^k, a_h^k} - P_h^k)(V_{h+1}^\star - V_{h+1}^{\pi^k})}_{=: \xi_2},$$

where the second inequality invokes Lemma 2. Therefore, it is sufficient to bound ξ_1 and ξ_2 separately, which we accomplish as follows.

- Regarding the term ξ_2, we first make the observation that $\left\{ (P_{h, s_h^k, a_h^k} - P_h^k)(V_{h+1}^\star - V_{h+1}^{\pi^k}) \right\}$ forms a martingale difference sequence, as π_j^k is determined by $Q_j^{k-1}(s_j^k, a)$. Moreover, the sequence satisfies the trivial bound
$$\left| (P_{h, s_h^k, a_h^k} - P_h^k)(V_{h+1}^\star - V_{h+1}^{\pi^k}) \right| \leq H.$$

These properties allow us to apply the celebrated Azuma-Hoeffding inequality (Azuma, 1967), which together with the trivial upper bound $\sum_{h=1}^H |\mathcal{I}_h^K| \leq KH$ ensures that

$$|\xi_2| = \left| \sum_{h=1}^H \sum_{k \in \mathcal{I}_h^K} (P_{h, s_h^k, a_h^k} - P_h^k)(V_{h+1}^\star - V_{h+1}^{\pi^k}) \right| \leq H \sqrt{HK \log \frac{2}{\delta}} \tag{31}$$

with probability at least $1 - \delta$.

- Turning to the term ξ_1, we apply the Cauchy-Schwarz inequality to derive

$$\xi_1 = \sum_{h=1}^H \sum_{k \in \mathcal{I}_h^K} 2\beta \sqrt{\varphi_h(s_h^k, a_h^k)^\top \left(\Lambda_h^{k-1} \right)^{-1} \varphi_h(s_h^k, a_h^k)}$$

$$\leq 2\beta \sqrt{KH} \sqrt{\sum_{h=1}^H \sum_{k \in \mathcal{I}_h^K} \varphi_h(s_h^k, a_h^k)^\top \left(\Lambda_h^{k-1} \right)^{-1} \varphi_h(s_h^k, a_h^k)}. \tag{32}$$

To further control the right-hand side of (32), we resort to Lemma 4 in Section A.3 — a result borrowed from (Abbasi-Yadkori et al., 2011) — which immediately leads to

$$\xi_1 \leq 2\beta \sqrt{KH} \cdot \sqrt{2Hd \log(KH)} = 2H\beta \sqrt{2dK \log(KH)}. \tag{33}$$

Putting everything together gives

$$\sum_{k \in \mathcal{I}_1^K} \left[V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k) \right] \leq 2H\beta \sqrt{2dK \log(KH)} + \sqrt{H^3 K \log \frac{2}{\delta}} \leq 4c_\beta \sqrt{d^2 H^6 K \log^2 \frac{KH}{\delta}}, \tag{34}$$

where the last inequality makes use of the definition $\beta := c_\beta \sqrt{dH^4 \log \frac{KH}{\delta}}$.

14

Step 3: bounding the number of state revisits. To this end, we make the observation that $N = |\mathcal{I}_1^K|$ and $K = |\mathcal{I}_H^K|$ (see Lemma 1). With this in mind, we can bound the total number $K - N$ of state revisits as follows:

$$K - N = |\mathcal{I}_H^K| - |\mathcal{I}_1^K| = \sum_{h=1}^{H} \left(|\mathcal{I}_{h+1}^K| - |\mathcal{I}_h^K| \right) \leq \frac{4c_\beta^2 d^2 H^5 \log^2 \frac{KH}{\delta}}{\Delta_{\mathsf{gap}}^2}. \tag{35}$$

Here, the above inequality is a consequence of the auxiliary lemma below, whose proof is provided in Section A.2.

Lemma 3. *Suppose that $KH \geq 2$. For all $1 \leq h < H$, the following condition*

$$\left| \mathcal{I}_{h+1}^K \setminus \mathcal{I}_h^K \right| \leq \frac{4c_\beta^2 d^2 H^4 \log^2 \frac{KH}{\delta}}{\Delta_{\mathsf{gap}}^2} \tag{36}$$

holds, where $c_\beta > 0$ is the pre-constant defined in (16).

Step 4: sample complexity analysis. Recall that T stands for the total number of samples collected, which clearly satisfies $T \geq K$. Consequently, the above results (35) and (34) taken collectively lead to

$$0 \leq K - N \leq \frac{4c_\beta^2 d^2 H^5 \log^2 \frac{TH}{\delta}}{\Delta_{\mathsf{gap}}^2} \leq N \qquad \Longrightarrow \qquad N \leq K \leq 2N, \tag{37}$$

provided that $N \geq \frac{4c_\beta^2 d^2 H^5 \log^2 \frac{HT}{\delta}}{\Delta_{\mathsf{gap}}^2}$. This together with the fact $T \leq KH$ implies that

$$T \leq KH \leq 2NH \qquad \Longrightarrow \qquad N \geq \frac{T}{2H}. \tag{38}$$

As a result, we can invoke (34) to obtain

$$\frac{1}{N} \sum_{n=1}^{N} \left(V_1^\star(s_1^{(n)}) - V_1^{\pi^{(n)}}(s_1^{(n)}) \right) = \frac{1}{N} \sum_{k \in \mathcal{I}_1^K} \left[V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k) \right] \leq 4c_\beta \sqrt{\frac{d^2 H^6 K \log^2 \frac{HT}{\delta}}{N^2}}$$

$$\leq 8c_\beta \sqrt{\frac{d^2 H^7 \log^2 \frac{HT}{\delta}}{T}}, \tag{39}$$

where the last relation arises from both (37) and (38). This concludes the proof.

5.2 Analysis for regret over the paths (proof of (21))

Proof of (21a). Invoking the crude bound $V^\star(s_1^k) - V^{\pi^k}(s_1^k) \leq H$ leads to

$$\begin{aligned}
\mathsf{Regret}_{\mathsf{path}}(K) &= \sum_{k \in \mathcal{I}_1^K} \left[V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k) \right] + \sum_{k \notin \mathcal{I}_1^K} \left[V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k) \right] \\
&\leq \sum_{k \in \mathcal{I}_1^K} \left[V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k) \right] + H \left(K - |\mathcal{I}_1^K| \right) \\
&= \sum_{k \in \mathcal{I}_1^K} \left[V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k) \right] + H(K - N) \\
&\leq 4c_\beta \sqrt{d^2 H^6 K \log^2 \frac{HT}{\delta}} + \frac{4c_\beta^2 d^2 H^6 \log^2 \frac{TH}{\delta}}{\Delta_{\mathsf{gap}}^2},
\end{aligned} \tag{40}$$

where the penultimate line relies on (26), and the last relation holds due to (34) and (35).

15

Proof of (21b) **(logarithmic regret).** As it turns out, this logarithmic regret bound (w.r.t. K) can be established by combining our result with a result derived in Yang et al. (2021). To be precise, by defining $\Delta_h(s,a) := V_h^\star(s) - Q_h^\star(s,a)$, we make the following observation:

$$\mathsf{Regret}_{\mathsf{path}}(K) = \sum_{k=1}^K \left(V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k)\right) = \sum_{k=1}^K \mathbb{E}\left[\sum_{h=1}^H \Delta_h(s_h^k, a_h^k) \mid \pi^k, s_1^k\right],$$

which has been derived in Yang et al. (2021, Equation (1)). Unconditioning gives

$$\mathbb{E}\left[\mathsf{Regret}_{\mathsf{path}}(K)\right] = \mathbb{E}\left[\sum_{k=1}^K \left(V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k)\right)\right] = \mathbb{E}\left[\sum_{k=1}^K \sum_{h=1}^H \Delta_h(s_h^k, a_h^k)\right] = \mathbb{E}\left[\sum_{h=1}^H \sum_{k=1}^K \Delta_h(s_h^k, a_h^k)\right]$$

$$= \mathbb{E}\left[\sum_{h=1}^H \sum_{k \in \mathcal{I}_h^K} \Delta_h(s_h^k, a_h^k)\right] + \mathbb{E}\left[\sum_{h=1}^H \sum_{k \notin \mathcal{I}_h^K} \Delta_h(s_h^k, a_h^k)\right]. \tag{41}$$

In addition, we make note of the fact that: with probability at least $1 - \delta$, one has

$$\Delta_h(s_h^k, a_h^k) = 0 \qquad \text{for all} \quad k \in \mathcal{I}_h^K,$$

which follows immediately from the update rule of Algorithm 2 (cf. line 14) and Lemma 2. This taken collectively with the trivial bound $\Delta_h(s_h^k, a_h^k) \leq H$ gives

$$\mathbb{E}\left[\sum_{h=1}^H \sum_{k \in \mathcal{I}_h^K} \Delta_h(s_h^k, a_h^k)\right] \leq (1-\delta) \cdot 0 + \delta \cdot \sum_{h=1}^H \sum_{k \in \mathcal{I}_h^K} H \leq H^2 K \delta.$$

Substitution into (41) yields

$$\mathbb{E}\left[\mathsf{Regret}_{\mathsf{path}}(K)\right] \leq H^2 K \delta + \sum_{h=1}^H \sum_{k \notin \mathcal{I}_h^K} \mathbb{E}\left[\Delta_h(s_h^k, a_h^k)\right] \leq H^2 K \delta + \sum_{h=1}^H \sum_{k \notin \mathcal{I}_h^K} H$$

$$\leq H^2 K \delta + H^2(K - N) \leq H^2 K \delta + \frac{4 c_\beta^2 d^2 H^7 \log^2 \frac{TH}{\delta}}{\Delta_{\mathsf{gap}}^2}.$$

Here, the second inequality follows from the trivial upper bound $\Delta_h(s_h^k, a_h^k) \leq H$, the third inequality holds true since $K - |\mathcal{I}_h^K| \leq K - |\mathcal{I}_H^K| = K - N$ (see Lemma 1), whereas the last inequality is valid due to (35). Taking $\delta = 1/K$ and recalling that $T \leq KH$, we arrive at the advertised logarithmic regret bound:

$$\mathbb{E}\left[\mathsf{Regret}_{\mathsf{path}}(K)\right] \leq \frac{17 c_\beta^2 d^2 H^7 \log^2(KH)}{\Delta_{\mathsf{gap}}^2}.$$

6 Discussion

In this paper, we have made progress towards understanding the plausibility of achieving sample-efficient RL when the optimal Q-function is linearly realizable. While prior works suggested an exponential sample size barrier in the standard online RL setting even in the presence of a constant sub-optimality gap, we demonstrate that this barrier can be conquered by permitting state revisiting (also called local access to generative models). An algorithm called LinQ-LSVI-UCB has been developed that provably enjoys a reduced sample complexity, which is polynomial in the feature dimension, the horizon and the inverse sub-optimality gap, but otherwise independent of the dimension of the state/action space.

Note, however, that linear function approximation for online RL remains a rich territory for further investigation. In contrast to the tabular setting, the feasibility and limitations of online RL might vary drastically across different families of linear function approximation. There are numerous directions that call for further theoretical development in order to obtain a more complete picture. For instance, can we

identify other flexible, yet practically relevant, online RL sampling mechanisms that also allow for sample size reduction? Can we derive the information-theoretic sampling limits for various linear function approximation classes, and characterize the fundamental interplay between low-dimensional representation and sampling constraints? Moving beyond linear realizability assumptions, a very recent work Yin et al. (2021) showed that a gap-independent sample size reduction is feasible by assuming that Q^π is linearly realizable for any policy π. However, what is the sample complexity limit for this class of function approximation remains largely unclear, particularly when state revisiting is not permitted. All of these are interesting questions for future studies.

Acknowledgements

A   Proof of technical lemmas

A.1   Proof of Lemma 2

Step 1: decomposition of $\theta_h^k - \theta_h^\star$.   To begin with, recalling the update rule (15b), we have the following decomposition

$$
\theta_h^k - \theta_h^\star = \left(\Lambda_h^k\right)^{-1}\left\{ \sum_{i \in \mathcal{I}_h^k} \varphi_h(s_h^i, a_h^i)\left[r_h(s_h^i, a_h^i) + \left\langle \varphi_{h+1}(s_{h+1}^i, a_{h+1}^i), \theta_{h+1}^k \right\rangle\right] - \Lambda_h^k \theta_h^\star \right\}
$$

$$
= \left(\Lambda_h^k\right)^{-1}\left\{ \sum_{i \in \mathcal{I}_h^k} \varphi_h(s_h^i, a_h^i)\left[\left\langle \varphi_{h+1}(s_{h+1}^i, a_{h+1}^i), \theta_{h+1}^k \right\rangle - P_{h, s_h^i, a_h^i} V_{h+1}^\star\right] - \theta_h^\star \right\}. \tag{42}
$$

To see why the second identity holds, note that from the definition of Λ_h^k (cf. (15a)) we have

$$
\Lambda_h^k \theta_h^\star = \sum_{i \in \mathcal{I}_h^k} \varphi_h(s_h^i, a_h^i)\left(\varphi_h(s_h^i, a_h^i)\right)^\top \theta_h^\star + \theta_h^\star
$$

$$
= \sum_{i \in \mathcal{I}_h^k} \varphi_h(s_h^i, a_h^i) Q_h^\star(s_h^i, a_h^i) + \theta_h^\star
$$

$$
= \sum_{i \in \mathcal{I}_h^k} \varphi_h(s_h^i, a_h^i)\left[r_h(s_h^i, a_h^i) + P_{h, s_h^i, a_h^i} V_{h+1}^\star\right] + \theta_h^\star,
$$

where the second and the third identities invoke the linear realizability assumption of $Q_h^\star$ and the Bellman equation, respectively.

As a result of (42), to control the difference $\theta_h^k - \theta_h^\star$, it is sufficient to bound $\left\langle \varphi_{h+1}(s_{h+1}^i, a_{h+1}^i), \theta_{h+1}^k \right\rangle - P_{h, s_h^i, a_h^i} V_{h+1}^\star$. Towards this, we start with the following decomposition

$$
\left\langle \varphi_{h+1}(s_{h+1}^i, a_{h+1}^i), \theta_{h+1}^k \right\rangle - P_{h, s_h^i, a_h^i} V_{h+1}^\star = \left\langle \varphi_{h+1}(s_{h+1}^i, a_{h+1}^i), \theta_{h+1}^k \right\rangle - Q_{h+1}^\star(s_{h+1}^i, a_{h+1}^i)
$$

$$
+ Q_{h+1}^\star(s_{h+1}^i, a_{h+1}^i) - V_{h+1}^\star(s_{h+1}^i) + V_{h+1}^\star(s_{h+1}^i) - P_{h, s_h^i, a_h^i} V_{h+1}^\star.
$$

For notational simplicity, let us define

$$\varepsilon_h^k := \left[\langle \varphi_h(s_h^i, a_h^i), \theta_h^k \rangle - Q_h^\star(s_h^i, a_h^i) \right]_{i \in \mathcal{I}_h^k} \qquad \in \mathbb{R}^{|\mathcal{I}_h^k|}, \tag{43a}$$

$$\delta_h^k := \left[Q_{h+1}^\star(s_{h+1}^i, a_{h+1}^i) - V_{h+1}^\star(s_{h+1}^i) \right]_{i \in \mathcal{I}_h^k} \qquad \in \mathbb{R}^{|\mathcal{I}_h^k|}, \tag{43b}$$

$$\xi_h^k := \left[V_{h+1}^\star(s_{h+1}^i) - P_{h, s_h^i, a_h^i} V_{h+1}^\star \right]_{i \in \mathcal{I}_h^k} \qquad \in \mathbb{R}^{|\mathcal{I}_h^k|}, \tag{43c}$$

$$\Phi_h^k := \left[\varphi_h(s_h^i, a_h^i) \right]_{i \in \mathcal{I}_h^k} \qquad \in \mathbb{R}^{d \times |\mathcal{I}_h^k|}. \tag{43d}$$

Here and throughout, for any $z = [z_i]_{1 \leq i \leq K}$, the vector $[z_i]_{i \in \mathcal{I}_h^k}$ denotes a subvector of z formed by the entries with indices coming from $\mathcal{I}_h^k$; for any set of vectors $w_1, \cdots, w_K$, the matrix $[w_i]_{i \in \mathcal{I}_h^k}$ represents a submatrix of $[w_1, \cdots, w_K]$ whose columns are formed by the vectors with indices coming from $\mathcal{I}_h^k$. Armed with this set of notation, $\theta_h^k - \theta_h^\star$ can be succinctly expressed as

$$\theta_h^k - \theta_h^\star = \left(\Lambda_h^k \right)^{-1} \left\{ \Phi_h^k \left(\left[\varepsilon_{h+1}^k \right]_{i \in \mathcal{I}_h^k} + \delta_h^k + \xi_h^k \right) - \theta_h^\star \right\}, \tag{44}$$

where we further define

$$\left[\varepsilon_{h+1}^k \right]_{i \in \mathcal{I}_h^k} := \left[\langle \varphi_{h+1}(s_{h+1}^i, a_{h+1}^i), \theta_{h+1}^k \rangle - Q_{h+1}^\star(s_{h+1}^i, a_{h+1}^i) \right]_{i \in \mathcal{I}_{h+1}^k \cap \mathcal{I}_h^k};$$

in other words, we consider the vector ε_{h+1}^k when restricted to the index set $\mathcal{I}_{h+1}^k \cap \mathcal{I}_h^k$. Recognizing that $\mathcal{I}_h^k \subseteq \mathcal{I}_{h+1}^k$ (see Lemma 1), we can also simply write

$$\left[\varepsilon_{h+1}^k \right]_{i \in \mathcal{I}_h^k} = \left[\langle \varphi_{h+1}(s_{h+1}^i, a_{h+1}^i), \theta_{h+1}^k \rangle - Q_{h+1}^\star(s_{h+1}^i, a_{h+1}^i) \right]_{i \in \mathcal{I}_h^k}.$$

Step 2: decomposition of $Q_h^k(s, a) - Q_h^\star(s, a)$. We now employ the above decomposition of $\theta_h^k - \theta_h^\star$ to help control $Q_h^k(s, a) - Q_h^\star(s, a)$ — the target quantity of Lemma 2. By virtue of the relation (44), our estimate $\langle \varphi_h(s, a), \theta_h^k \rangle$ of the linear representation $Q_h^\star(s, a) = \langle \varphi_h(s, a), \theta_h^\star \rangle$ satisfies

$$\left| \langle \varphi_h(s, a), \theta_h^k \rangle - Q_h^\star(s, a) \right| = \left| \langle \varphi_h(s, a), \theta_h^k - \theta_h^\star \rangle \right|$$
$$= \left| \varphi_h(s, a)^\top \left(\Lambda_h^k \right)^{-1} \left\{ \Phi_h^k \left(\left[\varepsilon_{h+1}^k \right]_{i \in \mathcal{I}_h^k} + \delta_h^k + \xi_h^k \right) - \theta_h^\star \right\} \right|$$
$$\leq \left\| \left(\Lambda_h^k \right)^{-1/2} \varphi_h(s, a) \right\|_2 \cdot$$
$$\left(\left\| \left(\Lambda_h^k \right)^{-1/2} \Phi_h^k \right\| \left\{ \left\| \varepsilon_{h+1}^k \right\|_2 + \left\| \delta_h^k \right\|_2 \right\} + \left\| \left(\Lambda_h^k \right)^{-1/2} \Phi_h^k \xi_h^k \right\|_2 + \left\| \left(\Lambda_h^k \right)^{-1/2} \right\| \left\| \theta_h^\star \right\|_2 \right),$$

where $\|M\|$ denotes the spectral norm of a matrix M. Here, the last inequality follows from the Cauchy-Schwarz inequality and the triangle inequality. Now, from the definition

$$\Lambda_h^k = \sum_{i \in \mathcal{I}_h^k} \varphi_h(s_h^i, a_h^i) \varphi_h(s_h^i, a_h^i)^\top + I = \Phi_h^k (\Phi_h^k)^\top + I, \tag{45}$$

it is easily seen that $\left\| \left(\Lambda_h^k \right)^{-1/2} \right\| \leq 1$ and $\left\| \left(\Lambda_h^k \right)^{-1/2} \Phi_h^k \right\| \leq 1$. Consequently, it is guaranteed that

$$\left| \langle \varphi_h(s, a), \theta_h^k \rangle - Q_h^\star(s, a) \right|$$
$$\leq \left(\left\| \varepsilon_{h+1}^k \right\|_2 + \left\| \delta_h^k \right\|_2 + \left\| \left(\Lambda_h^k \right)^{-1/2} \Phi_h^k \xi_h^k \right\|_2 + \left\| \theta_h^\star \right\|_2 \right) \cdot \left\| \left(\Lambda_h^k \right)^{-1/2} \varphi_h(s, a) \right\|_2. \tag{46}$$

In the sequel, we seek to establish, by induction, that

$$\left| \langle \varphi_h(s, a), \theta_h^k \rangle - Q_h^\star(s, a) \right| \leq b_h^k(s, a). \tag{47}$$

If this condition were true, then combining this with the definition (see (15d))

$$Q_h^k(s,a) = \min\left\{\langle \varphi_h(s,a), \theta_h^k\rangle + b_h^k(s,a),\, H\right\}$$

and the constraint $Q_h^\star(s,a) \le H$ would immediately lead to

$$Q_h^k(s,a) - Q_h^\star(s,a) \le \left|\langle \varphi_h(s,a), \theta_h^k\rangle + b_h^k(s,a) - Q_h^\star(s,a)\right| \le 2b_h^k(s,a),$$

$$Q_h^k(s,a) - Q_h^\star(s,a) \ge \begin{cases} 0, & \text{if } Q_h^k(s,a) \ge H, \\ b_h^k(s,a) - \left|\langle \varphi_h(s,a), \theta_h^k\rangle - Q_h^\star(s,a)\right| \ge 0, & \text{else,} \end{cases}$$

as claimed in the inequality (28) of this lemma. Consequently, everything boils down to establishing (47), which forms the main content of the next setp.

Step 3: proof of the inequality (47). The proof of this inequality proceeds by bounding each term in the relation (46).

To start with, we establish an upper bound on $\left\|(\Lambda_h^k)^{-1/2}\Phi_h^k\xi_h^k\right\|_2$ as required in our advertised inequality (47); that is, with probability at least $1-\delta$,

$$\left\|(\Lambda_j^k)^{-1/2}\Phi_j^k\xi_j^k\right\|_2 \le \sqrt{H^2\left(d\log(KH) + 2\log\frac{KH}{\delta}\right)} \le 2\sqrt{H^2 d\log\frac{KH}{\delta}} \tag{48}$$

holds simultaneously for all $1 \le j \le H$ and $1 \le k \le K$. Towards this end, let us define

$$X_i := V_{j+1}^\star(s_{j+1}^i) - P_{j,s_j^i,a_j^i}V_{j+1}^\star. \tag{49}$$

It is easily seen that $\{X_i\}_{i\in\mathcal{I}_j^K}$ forms a martingale sequence. In addition, we have the trivial upper bound $|X_i| \le H$. Therefore, applying the concentration inequality for self-normalized processes (see Lemma 4 in Section A.3), we can deduce that

$$\left\|(\Lambda_j^k)^{-1/2}\Phi_j^k\xi_j^k\right\|_2 = \left\|(\Lambda_j^k)^{-1/2}\sum_{i\in\mathcal{I}_j^k}\varphi_j(s_j^i,a_j^i)X_i\right\|_2 \le \sqrt{H^2\log\left(\frac{\det(\Lambda_j^k)}{\det(\Lambda_j^0)\,\delta^2}\right)}$$

$$\le \sqrt{H^2\left(d\log(KH) + 2\log\frac{1}{\delta}\right)} \tag{50}$$

holds with probability at least $1-\delta$. Here, the first inequality comes from Lemma 5 in Section A.3, whereas the second inequality is a consequence of Lemma 4 in Section A.3. Taking the union bound over $j = 1,\ldots,H$ and $k = 1,\ldots,K$ yields the required relation (48).

Armed with the above inequality, we can move on to establish the inequality (47) by induction, working backwards. First, we observe that when $h = H+1$, the inequality (47) holds true trivially (due to the initialization $\theta_{H+1}^k = 0$ and the fact $Q_{H+1}^\star = 0$). Next, let us assume that the claim holds for $h+1,\ldots,H+1$, and show that the claim continues to hold for step h. To this end, it is sufficient to bound the terms in (46) separately.

- **The term** $\|\delta_j^k\|_2$. From the induction hypothesis, the inequality (47) holds for $h+1,\ldots,H+1$. For any j obeying $h \le j \le H$ and any $i \in \mathcal{I}_j^k$, this in turn guarantees that

$$V_{j+1}^\star(s_{j+1}^i) = Q_{j+1}^\star\left(s_{j+1}^i, \pi^\star(s_{j+1}^i)\right) \le Q_{j+1}^{i-1}\left(s_{j+1}^i, \pi^\star(s_{j+1}^i)\right) \le Q_{j+1}^{i-1}(s_{j+1}^i, a_{j+1}^i)$$
$$\le \langle \varphi_h(s_{j+1}^i, a_{j+1}^i), \theta_{j+1}^{i-1}\rangle + b_{j+1}^{i-1}(s_{j+1}^i, a_{j+1}^i).$$

  Here, the second inequality follows since a_{j+1}^i is chosen to be an action maximizing $Q_{j+1}^{i-1}(s_{j+1}^i, \cdot)$. In the meantime, the induction hypothesis (47) for $h+1,\ldots,H+1$ also implies

$$Q_{j+1}^\star(s_{j+1}^i, a_{j+1}^i) \ge \langle \varphi_h(s_{j+1}^i, a_{j+1}^i), \theta_{j+1}^{i-1}\rangle - b_{j+1}^{i-1}(s_{j+1}^i, a_{j+1}^i)$$

19

for all j obeying $h \leq j \leq H$. Taken collectively, the above two inequalities demonstrate that

$$0 \leq V_{j+1}^\star(s_{j+1}^i) - Q_{j+1}^\star(s_{j+1}^i, a_{j+1}^i) \leq 2b_{j+1}^{i-1}(s_{j+1}^i, a_{j+1}^i), \qquad i \in \mathcal{I}_j^k,$$

where the first inequality holds trivially since $V_{j+1}^\star(s) = \max_a Q_{j+1}^\star(s, a)$. Then, given that $i \in \mathcal{I}_j^k$, one necessarily has $b_{j+1}^{i-1}(s_{j+1}^i, a_{j+1}^i) < \Delta_{\mathsf{gap}}/2$, which combined with the sub-optimality gap assumption (9) implies that a_{j+1}^i cannot be a sub-optimal action in state s_{j+1}^i. Consequently, we reach

$$V_{j+1}^\star(s_{j+1}^i) - Q_{j+1}^\star(s_{j+1}^i, a_{j+1}^i) = 0 \qquad \text{for all } i \in \mathcal{I}_j^k,$$

and as a result,

$$\delta_j^k = 0. \tag{51}$$

- **The term** $\|\varepsilon_{h+1}^k\|_2$. Recall that the decomposition (44) together with the assumption $Q_h^\star(s, a) = \langle \varphi_h(s, a), \theta_h^\star \rangle$ allows us to write ε_h^k (cf. (43a)) as follows

$$\varepsilon_h^k = \left(\Phi_h^k\right)^\top (\theta_h^k - \theta_h^\star) = \left(\Phi_h^k\right)^\top \left(\Lambda_h^k\right)^{-1} \left\{ \Phi_h^k \left(\left[\varepsilon_{h+1}^k\right]_{i \in \mathcal{I}_h^k} + \delta_h^k + \xi_h^k \right) - \theta_h^\star \right\}.$$

Combining this with the basic properties $\|(\Phi_h^k)^\top (\Lambda_h^k)^{-1/2}\| \leq 1$, $\|(\Lambda_h^k)^{-1/2}\| \leq 1$, and $\mathcal{I}_h^K \subseteq \mathcal{I}_{h+1}^K$ yields

$$\|\varepsilon_h^k\|_2 = \left\| \left(\Phi_h^k\right)^\top \left(\Lambda_h^k\right)^{-1} \left\{ \Phi_h^k \left(\left[\varepsilon_{h+1}^k\right]_{i \in \mathcal{I}_h^k} + \delta_h^k + \xi_h^k \right) - \theta_h^\star \right\} \right\|_2$$

$$\leq \left\|\varepsilon_{h+1}^k\right\|_2 + \left\|\delta_h^k\right\|_2 + \left\| \left(\Lambda_h^k\right)^{-1/2} \Phi_h^k \xi_h^k \right\|_2 + \left\|\theta_h^\star\right\|_2. \tag{52}$$

Applying this inequality recursively leads to

$$\|\varepsilon_h^k\|_2 \leq \left\|\varepsilon_{H+1}^k\right\|_2 + \sum_{h \leq j \leq H} \left[\|\delta_j^k\|_2 + \left\| \left(\Lambda_h^k\right)^{-1/2} \Phi_j^k \xi_j^k \right\|_2 + \|\theta_j^\star\|_2 \right]$$

$$\leq 4\sqrt{dH^4 \log \frac{KH}{\delta}}, \tag{53}$$

where the last inequality holds by putting together the property $\varepsilon_{H+1}^k = 0$, the inequalities (48) and (51), and the assumption that $\|\theta_j^\star\|_2 \leq 2H\sqrt{d}$ (see (8)).

Combining the inequalities (48), (51), (53) with the relation (46), we arrive at

$$\left|\langle \varphi_h(s, a), \theta_h^k \rangle - Q_h^\star(s, a)\right| \leq \left(4\sqrt{dH^4 \log \frac{KH}{\delta}} + 2\sqrt{H^2 d \log \frac{KH}{\delta}} + 2H\sqrt{d} \right) \left\| \left(\Lambda_h^k\right)^{-1/2} \varphi_h(s, a) \right\|_2$$

$$\leq c_\beta \sqrt{dH^4 \log \frac{KH}{\delta}} \left\| \left(\Lambda_h^k\right)^{-1/2} \varphi_h(s, a) \right\|_2 = b_h^k(s, a),$$

provided that $c_\beta \geq 8$. This completes the induction step of (47), thus concluding the proof of Lemma 2.

A.2   Proof of Lemma 3

Consider any $1 \leq h < H$. In view of the definition (15c) of the UCB bonus b_h^k, one obtains

$$\sum_{k: k+1 \in \mathcal{I}_{h+1}^K} \left[b_{h+1}^{k-1}(s_{h+1}^k, a_{h+1}^k)\right]^2 \leq \sum_{k: k+1 \in \mathcal{I}_{h+1}^K} c_\beta^2 dH^4 \left[\varphi_{h+1}(s_{h+1}^k, a_{h+1}^k)\right]^\top \left(\Lambda_{h+1}^{k-1}\right)^{-1} \varphi_{h+1}(s_{h+1}^k, a_{h+1}^k) \log \frac{KH}{\delta}$$

$$\leq c_\beta^2 d^2 H^4 \log \frac{KH}{\delta} \log \frac{K+d}{d} \leq c_\beta^2 d^2 H^4 \log^2 \frac{KH}{\delta} =: B, \tag{54}$$

where the second inequality comes from a standard result stated in Lemma 4 of Section A.3, and the last inequality is valid as long as $KH \geq 2$.

As a direct consequence of the upper bound (54), there are no more than $\frac{B}{(\Delta_{\mathsf{gap}}/2)^2}$ elements in the set $\mathcal{I}_{h+1}^K$ obeying $b_{h+1}^{k-1}(s_{h+1}^k, a_{h+1}^k) \geq \Delta_{\mathsf{gap}}/2$. This combined with the update rule in Line 14 of Algorithm 2 immediately yields

$$\left|\mathcal{I}_{h+1}^K \setminus \mathcal{I}_h^K\right| \leq \frac{B}{(\Delta_{\mathsf{gap}}/2)^2} = \frac{4c_\beta^2 d^2 H^4 \log^2 \frac{KH}{\delta}}{\Delta_{\mathsf{gap}}^2},$$

since, by construction, any k with $k+1$ belonging to $\mathcal{I}_{h+1}^K \setminus \mathcal{I}_h^K$ must violate the condition $b_{h+1}^{k-1}(s_{h+1}^k, a_{h+1}^k) < \Delta_{\mathsf{gap}}/2$. This completes the proof.

A.3   Auxiliary lemmas

In this section, we provide a couple of auxiliary lemmas that have been frequently invoked in the literature of linear bandits and linear MDPs. The first result is concerned with the interplay between the feature map and the (regularized) covariance matrix Λ_h^k.

Lemma 4 (Abbasi-Yadkori et al. (2011)). *Under the assumption* (8) *and the definition* (15a)*, the following relationship holds true*

$$\sum_{i \in \mathcal{I}_h^k} \varphi_h(s_h^i, a_h^i)^\top \left(\Lambda_h^{i-1}\right)^{-1} \varphi_h(s_h^i, a_h^i) \leq 2 \log\left(\frac{\det\left(\Lambda_h^k\right)}{\det\left(\Lambda_h^0\right)}\right) \leq 2d \log\left(\frac{k}{d} + 1\right). \tag{55}$$

Proof. The first inequality is an immediate consequence of Abbasi-Yadkori et al. (2011) or Jin et al. (2020, Lemma D.2). Regarding the second inequality, let $\lambda_i \geq 1$ be the i-th largest eigenvalue of the positive-semidefinite matrix Λ_h^k. From the AM-GM inequality, it is seen that

$$\det\left(\Lambda_h^k\right) = \prod_{i=1}^d \lambda_i \leq \left(\frac{\sum_{i=1}^d \lambda_i}{d}\right)^d \leq \left(\frac{k+d}{d}\right)^d, \tag{56}$$

where the last inequality arises since (in view of the assumption (8))

$$\sum_{i=1}^d \lambda_i = \mathsf{Tr}\left(\Lambda_h^k\right) = d + \sum_{i \in \mathcal{I}_h^k} \left\|\varphi_h(s_h^i, a_h^i)\right\|_2^2 \leq d + k.$$

$\square$

The second result delivers a concentration inequality for the so-called self-normalized processes.

Lemma 5 (Abbasi-Yadkori et al. (2011)). *Assume that $\{X_t \in \mathbb{R}\}_{t=1}^\infty$ is a martingale w.r.t. the filtration $\{\mathcal{F}_t\}_{t=0}^\infty$ obeying*

$$\mathbb{E}\left[X_t \mid \mathcal{F}_{t-1}\right] = 0, \qquad \text{and} \qquad \mathbb{E}\left[e^{\lambda X_t} \mid \mathcal{F}_{t-1}\right] \leq e^{\frac{\lambda^2 \sigma^2}{2}}, \ \forall \ \lambda \in \mathbb{R}.$$

In addition, suppose that $\varphi_t \in \mathbb{R}^d$ is a random vector over $\mathcal{F}_{t-1}$, and define $\Lambda_t = \Lambda_0 + \sum_{i=1}^t \varphi_i \varphi_i^\top \in \mathbb{R}^{d \times d}$. Then with probability at least $1 - \delta$, it follows that

$$\left\|\left(\Lambda_t\right)^{-1/2} \sum_{i=1}^t \varphi_i X_i\right\|_2^2 \leq \sigma^2 \log\left(\frac{\det(\Lambda_t)}{\det(\Lambda_0)\delta^2}\right) \qquad \text{for all } t \geq 0.$$

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *NIPS*, volume 11, pages 2312–2320.

Agarwal, A., Jiang, N., Kakade, S. M., and Sun, W. (2019). Reinforcement learning: Theory and algorithms.

Agarwal, A., Kakade, S., Krishnamurthy, A., and Sun, W. (2020a). FLAMBE: Structural complexity and representation learning of low rank MDPs. *arXiv preprint arXiv:2006.10814*.

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

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. (2020c). Optimality and approximation with policy gradient methods in Markov decision processes. In *Conference on Learning Theory*, pages 64–66. PMLR.

Ayoub, A., Jia, Z., Szepesvari, C., Wang, M., and Yang, L. (2020). Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pages 463–474. PMLR.

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

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

Azuma, K. (1967). Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal, Second Series*, 19(3):357–367.

Baird, L. (1995). Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier.

Bertsekas, D. P. and Tsitsiklis, J. N. (1995). Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE conference on decision and control*, volume 1, pages 560–564. IEEE.

Cen, S., Cheng, C., Chen, Y., Wei, Y., and Chi, Y. (2020). Fast global convergence of natural policy gradient methods with entropy regularization. *arXiv preprint arXiv:2007.06558*.

Dann, C. and Brunskill, E. (2015). Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2818–2826.

Dimakopoulou, M., Zhou, Z., Athey, S., and Imbens, G. (2019). Balanced linear contextual bandits. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 3445–3453.

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

Du, S. S., Kakade, S. M., Wang, R., and Yang, L. F. (2020a). Is a good representation sufficient for sample efficient reinforcement learning? In *International Conference on Learning Representations*.

Du, S. S., Lee, J. D., Mahajan, G., and Wang, R. (2020b). Agnostic Q-learning with function approximation in deterministic systems: Tight bounds on approximation error and sample complexity. *Neural Information Processing Systems*.

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

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

Hao, B., Duan, Y., Lattimore, T., Szepesvári, C., and Wang, M. (2020). Sparse feature selection makes batch reinforcement learning more sample efficient. *arXiv preprint arXiv:2011.04019*.

He, J., Zhou, D., and Gu, Q. (2020). Logarithmic regret for reinforcement learning with linear function approximation. *arXiv preprint arXiv:2011.11566*.

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

Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., and Schapire, R. E. (2017). Contextual decision processes with low Bellman rank are PAC-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR.

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

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

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

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

Kearns, M. J. and Singh, S. P. (1999). Finite-sample convergence rates for Q-learning and indirect algorithms. In *Advances in neural information processing systems*, pages 996–1002.

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

Lattimore, T., Szepesvari, C., and Weisz, G. (2020). Learning with good feature representations in bandits and in RL with a generative model. In *International Conference on Machine Learning*, pages 5662–5670. PMLR.

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

Li, G., Shi, L., Chen, Y., Gu, Y., and Chi, Y. (2021b). Breaking the sample complexity barrier to regret-optimal model-free reinforcement learning. *accepted to Neural Information Processing Systems (NeurIPS)*.

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

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

Long, J. and Han, J. (2021). An l^2 analysis of reinforcement learning in high dimensions with kernel and neural network approximation. *arXiv preprint arXiv:2104.07794*.

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

Munos, R. (2005). Error bounds for approximate value iteration. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, pages 1006–1011.

Osband, I. and Van Roy, B. (2014). Model-based reinforcement learning and the Eluder dimension. In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 1*, pages 1466–1474.

Osband, I., Van Roy, B., and Wen, Z. (2016). Generalization and exploration via randomized value functions. In *International Conference on Machine Learning*, pages 2377–2386. PMLR.

Shariff, R. and Szepesvári, C. (2020). Efficient planning in large MDPs with weak linear function approximation. *arXiv preprint arXiv:2007.06184*.

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

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

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

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

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

Wang, B., Yan, Y., and Fan, J. (2021a). Sample-efficient reinforcement learning for linearly-parameterized mdps with a generative model. *arXiv preprint arXiv:2105.14016*.

Wang, R., Du, S. S., Yang, L. F., and Salakhutdinov, R. (2020a). On reward-free reinforcement learning with linear function approximation. *arXiv preprint arXiv:2006.11274*.

Wang, R., Salakhutdinov, R. R., and Yang, L. (2020b). Reinforcement learning with general value function approximation: Provably efficient approach via bounded Eluder dimension. *Advances in Neural Information Processing Systems*, 33.

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

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

Wei, C.-Y., Jahromi, M. J., Luo, H., and Jain, R. (2021). Learning infinite-horizon average-reward MDPs with linear function approximation. In *International Conference on Artificial Intelligence and Statistics*, pages 3007–3015.

Weisz, G., Amortila, P., Janzer, B., Abbasi-Yadkori, Y., Jiang, N., and Szepesvári, C. (2021a). On query-efficient planning in MDPs under linear realizability of the optimal state-value function. *arXiv preprint arXiv:2102.02049*.

Weisz, G., Amortila, P., and Szepesvári, C. (2021b). Exponential lower bounds for planning in MDPs with linearly-realizable optimal action-value functions. In *Algorithmic Learning Theory*, pages 1237–1264. PMLR.

Wen, Z. and Van Roy, B. (2017). Efficient reinforcement learning in deterministic systems with value function generalization. *Mathematics of Operations Research*, 42(3):762–782.

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

Yang, L. and Wang, M. (2019). Sample-optimal parametric Q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004.

Yang, L. and Wang, M. (2020). Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pages 10746–10756. PMLR.

Yang, Z., Jin, C., Wang, Z., Wang, M., and Jordan, M. I. (2020). Bridging exploration and general function approximation in reinforcement learning: Provably efficient kernel and neural value iterations. *arXiv preprint arXiv:2011.04622*.

Yin, D., Hao, B., Abbasi-Yadkori, Y., Lazić, N., and Szepesvári, C. (2021). Efficient local planning with linear function approximation. *arXiv preprint arXiv:2108.05533*.

Zanette, A., Brandfonbrener, D., Brunskill, E., Pirotta, M., and Lazaric, A. (2020a). Frequentist regret bounds for randomized least-squares value iteration. In *International Conference on Artificial Intelligence and Statistics*, pages 1954–1964. PMLR.

Zanette, A., Lazaric, A., Kochenderfer, M., and Brunskill, E. (2019). Limiting extrapolation in linear approximate value iteration. *Neural Information Processing Systems*.

Zanette, A., Lazaric, A., Kochenderfer, M., and Brunskill, E. (2020b). Learning near optimal policies with low inherent Bellman error. In *International Conference on Machine Learning*, pages 10978–10989. PMLR.

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

Zhou, D., Gu, Q., and Szepesvari, C. (2020). Nearly minimax optimal reinforcement learning for linear mixture Markov decision processes. *arXiv preprint arXiv:2012.08507*.