Relative Q-learning for Average-Reward Markov Decision Processes with Continuous States

Xiangyu Yang, Jiaqiao Hu, and Jian-Qiang Hu

***Abstract*— Markov decision processes are widely used for modeling sequential decision-making problems under uncertainty. We propose an online algorithm for solving a class of average-reward Markov decision processes with continuous state spaces in a model-free setting. The algorithm combines the classical relative Q-learning with an asynchronous averaging procedure, which permits the Q-value estimate at a state-action pair to be updated based on observations at other neighboring pairs sampled in subsequent iterations. These point estimates are then retained and used for constructing an interpolation-based function approximator that predicts the Q-function values at unexplored state-action pairs. We show that with probability one the sequence of function approximators converges to the optimal Q-function up to a constant. Numerical results on a simple benchmark example are reported to illustrate the algorithm.**

***Index Terms*— Dynamic systems and control; Markov processes; Online computation**

I. Introduction

Markov decision processes (MDPs) provide an important framework for studying sequential decision making problems under uncertainty. For discounted MDPs, many solution algorithms have been proposed, and there is a rich body of literature on this subject (e.g., [1], [2], [3]). When the discount factor is close to one and/or the system performance cannot be easily quantified in economic terms, it is often convenient and sometimes necessary to consider MDPs with average reward criterion [4]. Example applications of average-reward MDPs include the control of queueing networks [5], inventory management [6], automatic guided vehicles scheduling [7], and the optimization of networked systems [8]. Compared with discounted MDPs, average-reward MDPs receive less attention partly due to their analytical difficulties, such as the existence and structural properties of optimal policies; cf. [6], [9]. In addition, approximately solving the optimality equation for an average-reward problem in a model-free setting, e.g., when the immediate reward and/or transition dynamics are unknown, could also be very computationally challenging [10].

An effective class of methods for solving average-reward MDPs is based on adapting and extending the Q-learning algorithms [1] for discounted-reward problems. Early studies in, e.g., [11], [12], have shown promising performance of such an approach in a model-free environment. Reference [13] introduces the relative value iteration (RVI) Q-learning algorithm and provides the first convergence proof of Q-learning for average-reward problems. The idea of the algorithm is to approximate the RVI algorithm (see, e.g., [14], [4]) using a stochastic approximation (SA) recursion and then carry out Q-value updates by subtracting an offset that depends on a predetermined reference state or set of reference state-action pairs at each iteration step. Some recent developments based on RVI can be found in, e.g., [15], [16], [17]. Another class of algorithms directly learns the policy, i.e., the so-called policy gradient methods [18], [19], which are essentially simulation-based optimization techniques that work with parameterized policies. Vanilla policy gradients require that the system dynamics can be modeled or simulated and may suffer from high variance in gradient estimation. Consequently, their effective (online) model-free implementations often rely on the use of Q/value-function-based methods. For a detailed account of model-free algorithms for average-reward MDPs, we refer the reader to [20], [21]. We remark that as noted in [21], the majority of these algorithms have been developed for finite (or countable) state space problems, and there are few attempts aimed at addressing continuous-state MDPs with average reward criterion.

In this paper, we present a new model-free algorithm for solving a class of average-reward MDPs with continuous state spaces. The algorithm is also based on RVI and shares some similarities with the aforementioned RVI Q-learning method. However, unlike RVI Q-learning (which relies on enumerating all state-action pairs), in a continuous-state space, one must instead consider compact approximations of the tabular representation of the Q-function by working with only a countable number of state-action samples. In addition, since it is not possible for a state to be visited infinitely often along a single sample trajectory, another difficulty that arises in a continuous-state domain is how to obtain a reliable estimate of the Q-value at a state-action pair in an online method such as Q-learning. We address these issues through a novel combination of interpolation-based function approximation with an online

The work of X. Yang was supported by the China Postdoctoral Science Foundation under Grant 2023M732054, the Shandong Provincial Natural Science Foundation under Grant ZR2023QG159, and the Shandong Postdoctoral Science Foundation under Grant SDCX-RS-202303004. The work of J. Hu was supported by the U.S. National Science Foundation under Grant CMMI-2027527. The work of J.-Q. Hu was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 72033003, 72350710219 and 71720107003.

Xiangyu Yang is with the School of Management, Shandong University, Jinan, Shandong, P.R. China 250100 (e-mail: yangxiangyu@email.sdu.edu.cn).

Jiaqiao Hu is with the Department of Applied Mathematics & Statistics, State University of New York, Stony Brook, NY 11794-3600, U.S.A., (e-mail: jiaqiao.hu.1@stonybrook.edu).

Jian-Qiang Hu is with the School of Management, Fudan University, Shanghai, P.R. China 200433 (e-mail: hujq@fudan.edu.cn).

averaging procedure adapted from the so-called shrinking ball method [22]. In particular, compared with some of the existing function approximation techniques, which often require the approximator to be a non-expansion (e.g., [23]) or linear in structure (e.g., [24]), an interpolation-based approximator offers more flexibility and has the advantage of allowing the Q-values at unvisited locations to be effectively predicted using estimates at previously sampled state-action pairs that lie in their vicinity, leading to reasonable control decisions even at states that have not been visited thus far. The shrinking ball method was originally introduced in [22] for solving continuous (static) simulation optimization problems. The key idea here is to incorporate this technique into Q-learning in the spirit of [25] so that the Q-value estimate at a given state-action pair can be continuously updated by averaging the performance at all other pairs collected along a single trajectory produced from a learning policy.

At each iteration of the algorithm, given a Q-function approximator, an initial point estimate of the Q-value at the current state-action pair is first formed by using a simulation-based version of the average-reward optimality equation (AROE). The estimate is then used in the asynchronous averaging (shrinking ball) procedure to improve the Q-value estimates at all other previously sampled pairs that are considered to be sufficiently close to the current pair. These data are fully retained and subsequently used in an interpolation-based fitting strategy for constructing a new Q-function approximator. We note that the update on the function approximator is only carried out at certain iterations called "interpolation times," which occur at a frequency that decreases with the number of algorithm iterations. Under appropriate conditions, we show that the sequence of function approximators converges uniformly with probability one (w.p.1) to the unique optimal Q-function, modulo an offset value that does not influence the determination of the optimal policy; cf. [26].

Currently, the majority of techniques advocated in the literature for solving continuous-state problems resort to some forms of state space aggregation. Reference [27] discusses upper bounds on the approximation errors of state space aggregation. Reference [28] employs an adaptive aggregation technique based on confidence intervals. Reference [29] considers a weighted kernel function approximator using local averaging methods. Other related work, although less relevant to the average-reward setting, include, for example, the adaptive state aggregation method [30] and the nearest neighbor regression method [31] for discounted MDPs. All these approaches require a finite discretization of the state space, which could lead to computational difficulties, either resulting in a solution that is not accurate enough or in a computing effort that becomes excessively demanding. Some discretization-free approaches are the recently introduced empirical relative value learning (ERVL) [32] and approximate relative value learning (ARVL) [23]. To the best of our knowledge, these algorithms seem to be the only existing discretization-free methods for average-reward MDPs with provable convergence guarantees. Nevertheless, both ERVL and ARVL are offline techniques that require the use of a large number of predetermined transition samples in order to obtain a good approximation of the value function. Our algorithm, in a sense, can also be viewed as a version of (RVI) Q-learning with adaptive state aggregation. However, it is a fully online, model-free method that approximates the entire Q-function based on a single sample trajectory produced from a learning policy, and consequently can be applied when the transition dynamics are either unknown or difficult to estimate.

The rest of this paper is organized as follows. Section II gives preliminaries on the average-reward MDP model and presents the proposed algorithm. In Section III, we analyze the algorithm and prove its almost sure convergence. A simple numerical example is provided in Section IV. Finally, Section V concludes this paper.

II. Relative Q-learning for Continuous-State Average-Reward MDPs

A. Preliminaries

We consider an infinite-horizon average-reward MDP described by a tuple (S, A, p, R), where the state space S is a compact and connected subset of Euclidean d-space $\mathbb{R}^d$, the action space A is a (discrete) finite set, $p(\cdot|s,a)$ is the Markov transition density function on S given a state-action pair $(s,a) \in S \times A$, and $R(\cdot,\cdot) : S \times A \to \mathbb{R}$ is the expected immediate reward function. For ease of exposition, we assume that all actions are admissible at any state. We consider a model-free setting, in which the expected reward $R(s,a)$ cannot be evaluated exactly and the transition density p is also unknown, so only the transition samples are available.

Let Π denote the set of all stationary Markov policies, where each element is a mapping $\pi : S \to \Delta_A$, with Δ_A being a $|A|$-dimensional probability simplex, and $\pi(\cdot|s)$ represents a probability distribution on the action space A at state s. Under a given policy π, the process evolves as follows: given the current state s_t at time t, an action a_t is first sampled according to $\pi(\cdot|s_t)$ and applied to the system, then a random reward $r(s_t, a_t, \omega_t)$ is earned. Throughout the paper, we assume $R(s_t, a_t) = \mathsf{E}[r(s_t, a_t, \omega_t)]$ for all t, where ω_t is a random vector independently drawn from some fixed distribution. Next the system transitions to a new state $s_{t+1} \sim p(\cdot|s_t, a_t)$. The long-run average reward under policy π is defined as

$$J^\pi(s) := \liminf_{T\to\infty} \frac{1}{T} \mathsf{E}\Big[\sum_{t=0}^{T-1} R(s_t, a_t) \mid s_0 = s\Big],$$

where T is the decision horizon, s is a given initial state, and $a_t \sim \pi(\cdot|s_t)$ for all $t \geq 0$. The goal is to determine a stationary policy $\pi^* \in \Pi$ that maximizes $J^\pi(s)$ for all initial states $s \in S$.

It has been shown that under appropriate conditions (see, e.g., [33], [6] and references therein), the optimal average reward does not depend on the initial state s and satisfies the AROE:

$$J^* + V^*(s) = \max_a \{R(s,a) + \mathsf{E}_{y\sim p(\cdot|s,a)}[V^*(y)]\}, \qquad (1)$$

where $V^*(\cdot)$ is a bounded real-valued function and J^* is the optimal average reward such that $J^* \geq \sup_\pi J^\pi(s)$, $\forall s \in S$. Any maximizer of the right side of (1) defines a stationary

policy that is optimal over all states, i.e., if an action $a^* \sim \pi^*(\cdot|s)$ satisfies

$$J^* + V^*(s) = R(s,a^*) + \mathsf{E}_{y\sim p(\cdot|s,a^*)}[V^*(y)] \quad \forall s \in S,$$

then π^* is optimal and $J^{\pi^*}(s) = J^*$ for all $s \in S$. Hereafter, we suppress the subscript in the expectation for notational convenience.

Since the function V^* in (1) is unique up to a constant, we focus on the relative value function $V^r(s) := V^*(s) - V^*(s_{\mathbf{0}})$ with $V^r(s_{\mathbf{0}}) \equiv 0$, where $s_{\mathbf{0}} \in S$ is an arbitrary preselected state. Therefore, the AROE can be expressed in terms of the relative value function as

$$J^* + V^r(s) = \max_a\{R(s,a) + \mathsf{E}[V^r(y)]\}. \tag{2}$$

Define the optimal Q-function as $Q^*(s,a) := R(s,a) + \mathsf{E}[V^r(y)]$. Then we can state (2) in the following equivalent form:

$$J^* + Q^*(s,a) = R(s,a) + \mathsf{E}[\max_b Q^*(y,b)] \tag{3}$$

Note that since $V^r(s_{\mathbf{0}}) = \max_a Q^*(s_{\mathbf{0}},a) - J^* = 0$, we clearly have $J^* = \max_a Q^*(s_{\mathbf{0}},a)$.

B. Algorithm Description

As in RVI Q-learning, our algorithm works with a learning policy and uses the transition samples generated from the policy to construct a sequence of interpolation-based function approximators that iteratively approximates the solution to (3). With a slight abuse of notation, we use $\{\pi_t(\cdot|s)\}$ to present a collection of prespecified learning policies, where $\pi_t(\cdot|s)$ gives the probability that an action should be selected when state s is encountered at time t. Let $\{t_k\}$ be a sequence of interpolation times at which the function approximator is updated, where the index k is the number of updates. Let $B(s,r)$ be an open ball in $\mathbb{R}^d$ with center s and radius r. For a sequence of positive real numbers $\{r_t\}$, define $I_t(s_l,a_l) = \mathbb{1}\{s_t \in B(s_l,r_t)\} \cdot \mathbb{1}\{a_t = a_l\}$, indicating whether the current pair (s_t,a_t) falls in the neighborhood of a previously sampled pair (s_l,a_l), where $l < t$ and $\mathbb{1}\{\cdot\}$ is the indicator function. Whenever $I_t(s_l,a_l) = 1$, we say that the neighborhood of (s_l,a_l) has been visited at time t. For any state-action pair (s_l,a_l) sampled at time l, let $N_t(s_l,a_l) = \sum_{j=l+1}^{t} I_j(s_l,a_l)$ denote the number of times the neighborhoods of (s_l,a_l) have been visited between time $l+1$ and time t. We also let $N_t^k(s_l,a_l) = \sum_{j=t_{k-1}+1}^{t} I_j(s_l,a_l) = N_t(s_l,a_l) - N_{t_{k-1}}(s_l,a_l)$ for all $t \in [t_{k-1}+1, t_k]$, which represents the number of times the neighborhoods of (s_l,a_l) have been visited since the most recent interpolation time prior to time t.

Let Q_k be the function approximator of Q^* constructed at the k-th interpolation time t_k. For all $t \in [t_{k-1}+1, t_k]$, the estimated Q-value at (s_t,a_t), denoted by $\tilde{Q}_t(s_t,a_t)$, is obtained using a simulation-based version of (3) with the current approximation Q_{k-1} replacing the true Q-function Q^*, that is,

$$\begin{aligned}\tilde{Q}_t(s_t,a_t) &= r(s_t,a_t,\omega_t) \\ &\quad + \max_b Q_{k-1}(s_{t+1},b) - \max_b Q_{k-1}(s_{\mathbf{0}},b).\end{aligned} \tag{4}$$

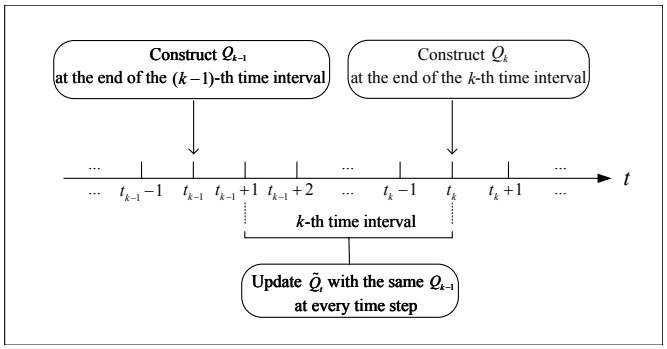

Fig. 1. A graphical illustration of the timeline for the construction of the sequence of Q-function approximators $\{\boldsymbol{Q_k}\}$. Note that $\boldsymbol{Q_k}$ remains unchanged between successive interpolation times for all $\boldsymbol{k}$.

Next, in order to reliably estimate the expectation and the reward R in (3), the point estimate $\tilde{Q}_t(s_t,a_t)$ is used in an asynchronous averaging procedure to adaptively update the Q-value estimates at all other state-action pairs that are considered to be sufficiently close to (s_t,a_t). Specifically, for each previously sampled state-action pair (s_l,a_l), if $a_t = a_l$ and $s_t \in B(s_l,r_t)$, then the Q-value estimate $\tilde{Q}_t(s_l,a_l)$ is updated by incorporating the new information $\tilde{Q}_t(s_t,a_t)$; otherwise, it remains unchanged. This leads to the following recursion:

$$\begin{aligned}\tilde{Q}_t(s_l,a_l) &= (1 - \alpha_t^k(s_l,a_l)I_t(s_l,a_l))\tilde{Q}_{t-1}(s_l,a_l) \\ &\quad + \alpha_t^k(s_l,a_l)I_t(s_l,a_l)\tilde{Q}_t(s_t,a_t),\end{aligned} \tag{5}$$

where $\alpha_t^k(s_l,a_l)$ is the learning rate at time t. We assume that the learning rate takes the form $\alpha_t^k(s_l,a_l) = f(N_t^k(s_l,a_l))$ for some real-valued function f, that is, it is viewed as a function of the number of times the neighborhoods of (s_l,a_l) have been visited since the most recent interpolation time prior to time t. Recursion (5) is essentially an asynchronous SA updating scheme, in which the usual deterministic step-size is replaced with a state-action pair-dependent random learning rate (see, e.g., Chapter 7 of [34]).

The detailed algorithmic steps and a pictorial illustration of the algorithm's general structure are presented in Algorithm 1 and Figure 1. The algorithm uses a separate function approximator $Q_k(s,a)$ for each action $a \in A$ to predict the Q-values at unvisited state-action pairs. We require the interpolation times to satisfy $t_k - t_{k-1} \to \infty$ as $k \to \infty$ so that there is an increasingly large number of iterations between successive updates of $Q_k(s,a)$; see Figure 1. Intuitively, as more state-action pairs are collected over $[t_{k-1}+1, t_k]$, the Q-value estimates obtained in (7) will become more accurate. This in turn allows the new function approximator $Q_k(s,a)$ to be constructed based on increasingly reliable data. Also, note that there is a trade-off involved in choosing the shrinking ball radius r_t. A large value of r_t helps to reduce the estimation noise (i.e., through averaging a large number of state-action pairs in (7)) but at the same time introduces a high estimation bias, and vice versa. The idea is thus to carefully control the decreasing speed of r_t so that both the noise and bias in the estimation can be eliminated by gradually sending r_t to

Algorithm 1: Relative Q-learning for Continuous-State Average-Reward MDPs

Input: Select a learning policy $\{\pi_t\}$, an initial state s_0, interpolation times $\{t_k\}$, learning rates $\{\alpha_t^k(s,a)\}$, shrinking ball radii $\{r_t\}$, and a reference state $s_{\mathbf{0}} \in S$.

Output: The function approximator Q_k.

$k \leftarrow 1$, $t \leftarrow 0$, $\Lambda_0 \leftarrow \emptyset$;

$Q_0(s,a) \leftarrow 0$, $\forall (s,a) \in S \times A$;

while *the stopping criterion is not satisfied* **do**

  Choose an action $a_t \sim \pi_t(\cdot|s_t)$, obtain $r(s_t,a_t,\omega_t)$, and observe the next state s_{t+1};

  $\Lambda_{t+1} \leftarrow \Lambda_t \cup \{(s_t,a_t)\}$;

  Estimate the Q-value at (s_t,a_t) as

$$\tilde{Q}_t(s_t,a_t) \leftarrow r(s_t,a_t,\omega_t) + \max_{b\in A} Q_{k-1}(s_{t+1},b) - \max_{b\in A} Q_{k-1}(s_{\mathbf{0}},b); \quad (6)$$

  foreach *previously sampled pair* $(s_l,a_l) \in \Lambda_t$ **do**

    Update the Q-value estimate as

$$\tilde{Q}_t(s_l,a_l) \leftarrow (1-\alpha_t^k(s_l,a_l)I_t(s_l,a_l))\tilde{Q}_{t-1}(s_l,a_l) + \alpha_t^k(s_l,a_l)I_t(s_l,a_l)\tilde{Q}_t(s_t,a_t); \quad (7)$$

  end

  if $t = t_k$ **then**

    foreach $a \in A$ **do**

      Construct $Q_k(s,a)$ by interpolating the data

$$\left\{\left((s',a'),\tilde{Q}_t(s',a')\right) : (s',a') \in \Lambda_t, a' = a\right\};$$

    end

    $k \leftarrow k+1$

  end

  $t \leftarrow t+1$

end

zero. We remark that in a finite-state space setting, each ball $B(s_l,r_t)$ will only contain the state s_l itself (assuming that the radius r_t is small enough). Thus, the update (7) will only be carried out when the same (s_l,a_l) pair is revisited at time t, in which case (6) and (7) together becomes identical to RVI Q-learning. Consequently, the algorithm can be viewed as a generalization of RVI Q-learning to continuous-state spaces.

III. Convergence Analysis

The convergence analysis is based on that of [25] with appropriate modifications tailored to the average-reward setting. We begin by introducing some notations. Define $\mathscr{F}_t = \sigma\{s_0,a_0,\omega_0,s_1,a_1,\omega_1,\ldots,s_t,a_t\}$. Let $\Lambda_t(a)$ be the set of sampled states contained in Λ_t at which action a is taken. The Euclidean distance between two states $s,s' \in S$ is denoted by $d(s,s')$, and for a set of states $C \subset S$, the distance between s and C is $d(s,C) := \inf_{s'\in C} d(s,s')$. The volume of a d-dimensional ball $B \subset S$ is denoted as $\mathrm{Vol}(B)$. For any two sequences of positive real numbers $\{a_t\}$ and $\{b_t\}$, we write $a_t = \Omega(b_t)$ if $\liminf_{t\to\infty} a_t/b_t > 0$. Denote by $\|\cdot\|_{\mathsf{TV}}$ the total variation norm for finite signed measures. For a bounded real-valued function $g(z)$ over a set Z, define the span seminorm of g as $\|g(z)\|_Z := \sup_{z\in Z} g(z) - \inf_{z\in Z} g(z)$. Note that $\|g(z)\|_Z = 0$ whenever $g(z)$ is a constant function on Z.

We make the following assumptions on the MDP model and algorithm parameters:

Assumptions:

A1. $R_{max} := \sup_{s,a,\omega} |r(s,a,\omega)| < \infty$. $R(\cdot,a)$ is Lipschitz continuous uniformly in a, i.e., there exists a constant K_R such that $|R(s,a) - R(s',a)| \le K_R d(s,s')$, $\forall s,s' \in S$, $\forall a \in A$.

A2. (i) There exists a constant $\beta \in (0,1)$ such that

$$\sup_{(s,a),(s',a')\in S\times A} \|q(\cdot|s,a) - q(\cdot|s',a')\|_{\mathsf{TV}} \le 2\beta,$$

where q is the one-step transition kernel of the underlying Markov chain.

(ii) There exists a constant K_p such that $\int |p(z|s,a) - p(z|s',a)|\,dz \le K_p d(s,s')$ for all $a \in A$ and $s,s' \in S$.

A3. For every $a \in A$, there exists an $L(a) < \infty$ such that the function approximator $Q_k(\cdot,a)$ is Lipschitz continuous uniformly in k with its Lipschitz constant bounded by $L(a)$ w.p.1.

A4. The learning rate function satisfies the following conditions: $f(i) \in (0,1)$ $\forall i$, $\sum_{i=1}^\infty f(i) = \infty$, and $\sum_{i=1}^\infty f^2(i) < \infty$.

A5. There exist constants γ, $\vartheta \in (0,1)$ such that

(i) $(t_k - t_{k-1})^{\frac{1}{2}} t_k^{-\gamma d} = \Omega(k^\epsilon)$ for an arbitrarily small constant $\epsilon > 0$;

(ii) The sequence of shrinking ball radii $\{r_t\}$ is non-increasing satisfying $r_t \to 0$ and $r_t = \Omega(t^{-\gamma})$;

(iii) The learning policy satisfies $\pi_t(a|s_t) \ge \vartheta$ for all $a \in A$, $s_t \in S$, and $t \ge 0$ w.p.1.

A6. There exist a (deterministic) stationary policy $\mu : S \to A$ and $\delta \in (0,1)$ such that for any d-dimensional ball $B \subset S$ with $\mathrm{Vol}(B) \le 1$, $\mathsf{P}_\mu(B) \ge \delta \mathrm{Vol}(B)$, where P_μ is the invariant probability measure of the state process under μ.

Assumption A1 has been previously adopted in, e.g., [23], [35], [36], to deal with computational issues for continuous-state MDP models. A2 involves regularity conditions on the transition dynamics of the underlying Markov chain. In particular, A2(i) implies that for any deterministic stationary policy μ', its associated t-step transition probability $\mathsf{P}^t_{\mu'}(\cdot|s)$ converges geometrically to its unique invariant probability measure $\mathsf{P}_{\mu'}$ in the sense that

$$\|\mathsf{P}^t_{\mu'}(\cdot|s) - \mathsf{P}_{\mu'}(\cdot)\|_{\mathsf{TV}} \le 2\beta^t \quad (8)$$

uniformly in s; see Lemma 3.3 on pp. 57 of [33]. Under A1 and A2, there exist a constant J^* and a bounded function V^* satisfying the AROE (1) (Corollary 3.6 in Chapter 3 of [33]). This further indicates the existence of an optimal stationary policy by the measurable selection theorem. For more general sufficient conditions that guarantee the existence of stationary optimal policies for average-reward MDPs, we refer the reader to, e.g., [6] and [37]. A3 requires the function approximator to be sufficiently smooth to quantify the prediction error at a given unvisited state-action pair based on information at

already sampled pairs. We remark that one potential limitation of A3 is that it is a condition that depends on the algorithm trajectory. Nevertheless, given that no prior knowledge about the Lipschitz constants is assumed, this smoothness requirement could be expected to hold by several interpolation methods such as barycentric interpolation, spline, and kernel-based approaches; see, e.g., [38]. A4 is the standard condition on the learning rate (step size) used in the SA literature. A5 includes some technical assumptions on algorithm input parameters. In particular, A5(i) imposes the condition on the growth rate of interpolation times t_k. A5(ii) characterizes the decreasing rate of the shrinking ball radius r_t. A5(iii) requires the learning policy to be constantly exploratory and is satisfied by widely-adopted ϵ-greedy learning policies [39]. In view of the ergodicity condition A2(i), A6 further requires the existence of a stationary policy μ, so that the Markov chain of the state process under μ is uniformly ergodic with an invariant probability measure that is bounded away from zero.

Our main result, as stated in Theorem 1 below, indicates that as the number of interpolations increases, the sequence of function approximators $\{Q_k\}$ will converge uniformly to the optimal Q-function Q^* w.p.1, modulo a constant value.

Theorem 1 Suppose all conditions A1−A6 are satisfied. As $k \to \infty$,

$$\|Q_k(s,a) - Q^*(s,a)\|_{S\times A} \to 0 \quad \text{w.p.1.}$$

Note that from (2), the optimal action a^* at any state s is obtained by $a^* = \arg\max_a Q^*(s,a)$, so adding a constant to Q^* will not have an effect on the choice of the optimal action a^*. Thus, when the algorithm terminates, we can use the last function approximator Q_k to approximately determine the optimal strategy.

The proof of Theorem 1 relies on a series of intermediate results (Lemmas 1−8 below). We begin with a preliminary result that shows the Lipschitz continuity of the optimal Q-function.

Lemma 1 If A1 and A2 hold, then for every $a \in A$, the optimal Q-function $Q^*(s,a)$ is Lipschitz continuous with Lipschitz constant $L_Q := (K_R + cK_p)$, where c is some positive constant.

Proof: Note that under A1 and A2, there exists a constant c such that $|V^r(s)| \le c$, $\forall s \in S$ (see Section 3.2 and 3.3 of [33]). Therefore, for each $a \in A$, we have for any $s, s' \in S$,

$$\begin{aligned}
&|Q^*(s,a) - Q^*(s',a)| \\
&= \Big|[R(s,a) + \int V^r(z)p(z|s,a)dz] \\
&\quad - [R(s',a) + \int V^r(z)p(z|s',a)dz]\Big| \\
&\le \big|R(s,a) - R(s',a)\big| + \int \big|V^r(z)[p(z|s,a) - p(z|s',a)]\big|dz \\
&\le K_R d(s,s') + cK_p d(s,s') \\
&= L_Q d(s,s'),
\end{aligned}$$

and the Lipschitz continuity of $Q^*(s,a)$ follows. ∎

Lemma 2 shows that the neighborhoods of each sampled state-action pair will be visited infinitely often (i.o.) from time $t_{k-1}+1$ to time t_k as $k \to \infty$. Therefore, the Q-value estimate is updated increasingly frequently during the time interval $[t_{k-1}+1, t_k]$ as k becomes large.

Lemma 2 If A2, A5, and A6 hold, then for each state-action pair (s_l, a_l) sampled at time l,

$$\mathsf{P}\big(\lim_{k\to\infty}(N_{t_k}(s_l,a_l) - N_{t_{k-1}}(s_l,a_l)) = \infty\big) = 1.$$

Proof: For the policy μ given in A6, we have from (8) and the properties of the total variation norm that $|\mathsf{P}^t_\mu(B|s) - \mathsf{P}_\mu(B)| \le \beta^t$ for all $s \in S$, $t \ge 0$, and any d-dimensional ball $B \subset S$ with $\mathrm{Vol}(B) \le 1$. It follows that $\mathsf{P}^t_\mu(B|s) \ge \mathsf{P}_\mu(B) - \beta^t \ge \delta\mathrm{Vol}(B) - \beta^t$. For a sufficiently large t, consider the ball $B(s_l, r_t)$ with $c_B t^{-\gamma d} \le \mathrm{Vol}(B(s_l,r_t)) \le 1$, where c_B is some positive constant (A5(ii)). Since $\beta^t \to 0$ at a geometric rate, there exists a positive integer m such that for all $s \in S$ and $t \ge m-1$, $\mathsf{P}^t_\mu(B(s_l,r_t)|s) \ge \delta\mathrm{Vol}(B(s_l,r_t)) - \frac{1}{2}\delta\mathrm{Vol}(B(s_l,r_t)) = \frac{1}{2}\delta\mathrm{Vol}(B(s_l,r_t)) > 0$. Hence, for any state s_{t-m+1} encountered at time $t-m+1$, under policy μ, we have

$$\mathsf{P}(s_t \in B(s_l,r_t)|s_{t-m+1}) \ge \frac{1}{2}\delta\mathrm{Vol}(B(s_l,r_t)).$$

Now for a fixed s_{t-m+1}, consider any sequence of actions $\{\hat{a}_{t-m+1}, \hat{a}_{t-m+2}, \ldots, \hat{a}_{t-1}\} \in A^{m-1}$ generated under μ. By A5(iii), the learning policy $\{\pi_t\}$ will take the same sequence of actions w.p. at least ϑ^{m-1}. Thus, under $\{\pi_t\}$, we have $\mathsf{P}(s_t \in B(s_l,r_t)|s_{t-m+1}) \ge \vartheta^{m-1}(\delta/2)\mathrm{Vol}(B(s_l,r_t))$. This in turn implies that for all $t \ge m$,

$$\begin{aligned}
&\mathsf{P}(s_t \in B(s_l,r_t)|\mathscr{F}_{t-m}) \\
&= \mathsf{P}(s_t \in B(s_l,r_t)|s_{t-m}, a_{t-m}) \\
&= \int_S \mathsf{P}(s_t \in B(s_l,r_t)|s_{t-m+1})q(ds_{t-m+1}|s_{t-m},a_{t-m}) \\
&\ge \frac{1}{2}\vartheta^{m-1}\delta\mathrm{Vol}(B(s_l,r_t)) \\
&\ge \frac{1}{2}\vartheta^{m-1}\delta c_B t^{-\gamma d}.
\end{aligned}$$

For notational brevity, define $\delta' := \frac{1}{2}\vartheta^m\delta$. Further let $\chi_k = \lfloor\frac{t_k - t_{k-1}-1}{m}\rfloor$ and $\rho_k = \delta' c_B(\chi_k+1)t_k^{-\gamma d} - 2\ln k - 2\sqrt{\mathcal{A}_k \ln k + \ln^2 k}$, where $\mathcal{A}_k := (\chi_k+1)(1-\delta' c_B t_k^{-\gamma d})$. We consider the following probability for a sufficiently large k (thus t is also large enough):

$$\begin{aligned}
&\mathsf{P}(N_{t_k}(s_l,a_l) - N_{t_{k-1}}(s_l,a_l) \le \rho_k) \\
&\le \mathsf{P}(\sum_{i=0}^{\chi_k} \mathbb{1}\{s_{t_k-im} \in B(s_l, r_{t_k-im}) \cap a_{t_k-im} = a_l\} \le \rho_k) \\
&= \mathsf{P}(\sum_{i=0}^{\chi_k} \mathbb{1}\{s_{t_k-im} \notin B(s_l, r_{t_k-im}) \cup a_{t_k-im} \ne a_l\} \\
&\qquad \ge \chi_k + 1 - \rho_k) \\
&\le \frac{\mathsf{E}[e^{\lambda\sum_{i=0}^{\chi_k}\mathbb{1}\{s_{t_k-im}\notin B(s_l,r_{t_k-im})\cup a_{t_k-im}\ne a_l\}}]}{e^{\lambda(\chi_k+1-\rho_k)}} \\
&\le \frac{\mathsf{E}[e^{\lambda\sum_{i=0}^{\chi_k}\mathbb{1}\{s_{t_k-im}\notin B(s_l,r_{t_k})\cup a_{t_k-im}\ne a_l\}}]}{e^{\lambda(\chi_k+1-\rho_k)}}
\end{aligned} \tag{9}$$

for any given constant $\lambda > 0$, where the second inequality follows from Markov's inequality and the third inequality is because the shrinking ball radius is non-increasing. Note that

$$\begin{aligned}
&\mathsf{E}[e^{\lambda \mathbb{1}\{s_{t_k} \notin B(s_l, r_{t_k}) \cup a_{t_k} \neq a_l\}} | \mathscr{F}_{t_k - m}] \\
&= (e^\lambda - 1)\mathsf{P}(s_{t_k} \notin B(s_l, r_{t_k}) \cup a_{t_k} \neq a_l | \mathscr{F}_{t_k - m}) + 1 \\
&= (e^\lambda - 1)[1 - \mathsf{P}(s_{t_k} \in B(s_l, r_{t_k}) | \mathscr{F}_{t_k - m}) \\
&\qquad \times \mathsf{P}(a_{t_k} = a_l | s_{t_k} \in B(s_l, r_{t_k}), \mathscr{F}_{t_k - m})] + 1 \\
&\leq (e^\lambda - 1)(1 - \delta' c_B t_k^{-\gamma d}) + 1.
\end{aligned}$$

A bound on the numerator of (9) can then be derived as follows:

$$\begin{aligned}
&\mathsf{E}[e^{\lambda \sum_{i=0}^{\chi_k} \mathbb{1}\{s_{t_k - im} \notin B(s_l, r_{t_k}) \cup a_{t_k - im} \neq a_l\}}] \\
&= \mathsf{E}\big[e^{\lambda \sum_{i=1}^{\chi_k} \mathbb{1}\{s_{t_k - im} \notin B(s_l, r_{t_k}) \cup a_{t_k - im} \neq a_l\}} \\
&\qquad \times \mathsf{E}[e^{\lambda \mathbb{1}\{s_{t_k} \notin B(s_l, r_{t_k}) \cup a_{t_k} \neq a_l\}} | \mathscr{F}_{t_k - m}]\big] \\
&\leq \big((e^\lambda - 1)(1 - \delta' c_B t_k^{-\gamma d}) + 1\big) \\
&\qquad \times \mathsf{E}[e^{\lambda \sum_{i=1}^{\chi_k} \mathbb{1}\{s_{t_k - im} \notin B(s_l, r_{t_k}) \cup a_{t_k - im} \neq a_l\}}] \\
&\leq \prod_{i=0}^{\chi_k} [(e^\lambda - 1)(1 - \delta' c_B t_k^{-\gamma d}) + 1] \\
&= \exp\big(\sum_{i=0}^{\chi_k} \ln[(e^\lambda - 1)(1 - \delta' c_B t_k^{-\gamma d}) + 1]\big) \\
&\leq \exp((\chi_k + 1)(e^\lambda - 1)(1 - \delta' c_B t_k^{-\gamma d})),
\end{aligned}$$

where the last inequality is due to the fact that $\ln(x+1) \leq x$ for $x \geq 0$. Plugging the above into (9) and optimizing the bound with respect to λ, we have

$$\mathsf{P}(N_{t_k}(s_l, a_l) - N_{t_{k-1}}(s_l, a_l) \leq \rho_k) \leq e^{\mathcal{B}_k (1 - \frac{\mathcal{A}_k}{\mathcal{B}_k} + \ln \frac{\mathcal{A}_k}{\mathcal{B}_k})},$$

where $\mathcal{B}_k := \chi_k + 1 - \rho_k$. Since $\rho_k \leq \delta' c_B (\chi_k + 1) t_k^{-\gamma d}$, we have $0 < \frac{\mathcal{A}_k}{\mathcal{B}_k} \leq 1$. Applying the inequality that $\ln x \leq (x-1) - \frac{1}{2}(x-1)^2$ for $x \in (0, 1]$, we obtain

$$\mathsf{P}(N_{t_k}(s_l, a_l) - N_{t_{k-1}}(s_l, a_l) \leq \rho_k) \leq e^{-\frac{(\mathcal{B}_k - \mathcal{A}_k)^2}{2\mathcal{B}_k}} = \frac{1}{k^2}.$$

It follows that

$$\sum_{k=1}^{\infty} \mathsf{P}(N_{t_k}(s_l, a_l) - N_{t_{k-1}}(s_l, a_l) \leq \rho_k) < \infty,$$

which implies that $\mathsf{P}(N_{t_k}(s_l, a_l) - N_{t_{k-1}}(s_l, a_l) \leq \rho_k, \text{ i.o.}) = 0$ by the Borel-Cantelli lemma (see, e.g., [40]). Finally, by A5(i), it can be observed that $\rho_k \to \infty$ as $k \to \infty$. This completes the proof. ∎

Lemma 3 indicates that for every action $a \in A$, the collection of states visited up to time t_k, i.e., $\Lambda_{t_k}(a)$, will become dense in S as $k \to \infty$.

Lemma 3 If A2, A5, and A6 hold, then for every $a \in A$, we have

$$\mathsf{P}\big(\lim_{k \to \infty} \sup_{s \in S} d(s, \Lambda_{t_k}(a)) = 0\big) = 1.$$

Proof: Let ϵ be small enough such that $\mathrm{Vol}(B(v, \epsilon/2)) \leq 1$ whenever $v \in S$. Since S is compact, we can find a finite collection of states $\{v_1, \ldots, v_n\}$ such that $S \subseteq \cup_{j=1}^n B(v_j, \epsilon/2)$. As in the proof of Lemma 2, there exists a constant T such that $\mathsf{P}(s_t \in B(v_j, \epsilon/2) | \mathscr{F}_{t-m}) \geq \frac{1}{2} \vartheta^{m-1} \delta c_B (\epsilon/2)^d$ and $\min_a \pi_t(a|s_t) \geq \vartheta$ for all $t \geq T$. Let $\ell_k = \lfloor \frac{t_k - 1 - T}{m} \rfloor$. For a sufficiently large k (thus $t \geq T$), we have

$$\begin{aligned}
&\mathsf{P}(\sup_{s \in S} d(s, \Lambda_{t_k}(a)) > \epsilon) = \mathsf{P}(\exists s' \in S,\, d(s', \Lambda_{t_k}(a)) > \epsilon) \\
&= \mathsf{P}(\exists s' \in S,\, B(s', \epsilon) \cap \Lambda_{t_k}(a) = \emptyset) \\
&\leq \mathsf{P}\big(\cup_{j=1}^n (B(v_j, \epsilon/2) \cap \Lambda_{t_k}(a) = \emptyset)\big) \\
&\leq \sum_{j=1}^n \mathsf{P}(B(v_j, \epsilon/2) \cap \Lambda_{t_k}(a) = \emptyset) \\
&= \sum_{j=1}^n \mathsf{P}\big((s_0 \notin B(v_j, \epsilon/2) \cup a_0 \neq a) \cap \ldots \cap \\
&\qquad (s_{t_k - 1} \notin B(v_j, \epsilon/2) \cup a_{t_k - 1} \neq a)\big) \\
&\leq \sum_{j=1}^n \mathsf{P}\big((s_{t_k - 1} \notin B(v_j, \epsilon/2) \cup a_{t_k - 1} \neq a) \\
&\qquad \cap (s_{t_k - 1 - m} \notin B(v_j, \epsilon/2) \cup a_{t_k - 1 - m} \neq a) \\
&\qquad \cap (s_{t_k - 1 - 2m} \notin B(v_j, \epsilon/2) \cup a_{t_k - 1 - 2m} \neq a) \\
&\qquad \cap \ldots \cap (s_{t_k - 1 - \ell_k m} \notin B(v_j, \epsilon/2) \cup a_{t_k - 1 - \ell_k m} \neq a)\big) \\
&= \sum_{j=1}^n \big[1 - \mathsf{P}\big(s_{t_k - 1} \in B(v_j, \epsilon/2) \cap a_{t_k - 1} = a | \\
&\qquad (s_{t_k - 1 - m} \notin B(v_j, \epsilon/2) \cup a_{t_k - 1 - m} \neq a) \cap \ldots \cap \\
&\qquad (s_{t_k - 1 - \ell_k m} \notin B(v_j, \epsilon/2) \cup a_{t_k - 1 - \ell_k m} \neq a)\big)\big] \\
&\qquad \times \mathsf{P}\big((s_{t_k - 1 - m} \notin B(v_j, \epsilon/2) \cup a_{t_k - 1 - m} \neq a) \cap \ldots \cap \\
&\qquad (s_{t_k - 1 - \ell_k m} \notin B(v_j, \epsilon/2) \cup a_{t_k - 1 - \ell_k m} \neq a)\big) \\
&\leq \sum_{j=1}^n [1 - \delta' c_B (\epsilon/2)^d] \\
&\qquad \times \mathsf{P}\big((s_{t_k - 1 - m} \notin B(v_j, \epsilon/2) \cup a_{t_k - 1 - m} \neq a) \cap \ldots \cap \\
&\qquad (s_{t_k - 1 - \ell_k m} \notin B(v_j, \epsilon/2) \cup a_{t_k - 1 - \ell_k m} \neq a)\big) \\
&\leq \sum_{j=1}^n \prod_{i=0}^{\ell_k} [1 - \delta' c_B (\epsilon/2)^d] \\
&\leq \sum_{j=1}^n \exp\big(-\sum_{i=0}^{\ell_k} \delta' c_B (\epsilon/2)^d\big) \\
&= n \exp\big(-\delta' c_B (\epsilon/2)^d (\ell_k + 1)\big).
\end{aligned}$$

It is easy to see that $\sum_{k=0}^{\infty} \mathsf{P}(\sup_{s \in S} d(s, \Lambda_{t_k}(a)) > \epsilon) < \infty$. Thus, the Borel-Cantelli lemma implies that $\mathsf{P}(\sup_{s \in S} d(s, \Lambda_{t_k}(a)) > \epsilon, \text{ i.o.}) = 0$. Finally, the result is proved because ϵ can be arbitrarily small. ∎

Next, we show that both the point estimate $\tilde{Q}_t(s, a)$ and the Q-function approximator $Q_k(s, a)$ constructed by the algorithm remain bounded at all times.

Lemma 4 If A1 and A3 hold, then $\max_{s \in \Lambda_t} \max_a |\tilde{Q}_t(s, a)|$ and $\sup_{s \in S} \max_a |Q_k(s, a)|$ are bounded for all t and k w.p.1.

Proof: For notational convenience, let $D_t = \max_{s \in \Lambda_t} \max_a |\tilde{Q}_t(s, a)|$, and denote by D the diameter of S, i.e., $D := \sup_{s, s' \in S} d(s, s')$. By A3, for any $s, s' \in S$ and $a \in A$, w.p.1

$$|Q_k(s, a)| \leq |Q_k(s', a)| + L d(s, s')$$

$$\leq |Q_k(s',a)| + LD,$$

where $L := \max_{a\in A} L(a) < \infty$. Furthermore, since $Q_k(s,a)$ is constructed by interpolating $\left\{\left((s',a'), \tilde{Q}_{t_k}(s',a')\right) : (s',a') \in \Lambda_{t_k}, a' = a\right\}$, we have $\sup_{s\in S}\max_a|Q_k(s,a)| \leq D_{t_k} + LD$. From (6), the point estimate obtained at any time $t > 0$ satisfies

$$\begin{aligned}
&|\tilde{Q}_t(s_t,a_t)| \\
&\leq R_{max} + |\max_b Q_{k-1}(s_{t+1},b) - \max_b Q_{k-1}(s_{\mathbf{0}},b)| \\
&\leq R_{max} + \max_b |Q_{k-1}(s_{t+1},b) - Q_{k-1}(s_{\mathbf{0}},b)| \\
&\leq R_{max} + LD.
\end{aligned}$$

This, together with (7), shows that $|\tilde{Q}_t(s_l,a_l)| \leq \max\{D_{t-1}, R_{max} + LD\}$ for every $(s_l,a_l) \in \Lambda_t$, and thus

$$D_t \leq \max\left\{D_{t-1}, R_{max} + LD\right\}.$$

Note that by construction, $Q_0(s,a) = 0$ for all $(s,a) \in S \times A$. Clearly, $D_0 \leq R_{max} + LD$, and a simple induction shows that $D_t \leq R_{max} + LD$ for all t. It follows that $\sup_{s\in S}\max_b|Q_k(s,a)| \leq D_{t_k} + LD \leq R_{max} + 2LD$ for all k. This completes the proof. ∎

Our main result is to show that the sequence of function approximators $\{Q_k\}$ converges to the optimal Q-function Q^* under span semi-norm. Since Q_k is constructed using point estimates $\tilde{Q}_{t_k}$, we proceed by investigating the convergence properties of the iterates generated by (6). For each $(s_l,a_l) \in \Lambda_{t_{k-1}}$ and $k \in \{1,2,\ldots\}$, we consider the error term

$$\varepsilon_t(s_l,a_l) := \tilde{Q}_t(s_l,a_l) - Q^*(s_l,a_l), \quad \forall t \in [t_{k-1}+1, t_k].$$

Also let $\eta_t(s_l,a_l) = \alpha_t^k(s_l,a_l)I_t(s_l,a_l)$ for notational convenience. Hence by subtracting both sides of (7) by $Q^*(s_l,a_l)$, we obtain the following recursion:

$$\begin{aligned}
\varepsilon_t(s_l,a_l) &= (1-\eta_t(s_l,a_l))\varepsilon_{t-1}(s_l,a_l) + \eta_t(s_l,a_l) \\
&\quad \times [r(s_t,a_t,\omega_t) + \max_b Q_{k-1}(s_{t+1},b) \\
&\quad - \max_b Q_{k-1}(s_{\mathbf{0}},b) - Q^*(s_l,a_l)] \\
&= (1-\eta_t(s_l,a_l))\varepsilon_{t-1}(s_l,a_l) + \eta_t(s_l,a_l) \\
&\quad \times (B_t(s_l,a_l) + W_t(s_t,a_t) + H_t(s_t,a_t)),
\end{aligned}$$

where $B_t(s_l,a_l) := Q^*(s_t,a_t) - Q^*(s_l,a_l)$ is the bias caused by using the shrinking ball strategy, $W_t(s_t,a_t) := r(s_t,a_t,\omega_t) + \max_b Q_{k-1}(s_{t+1},b) - \mathsf{E}[r(s_t,a_t,\omega_t) + \max_b Q_{k-1}(y,b)]$ is a noise term where $y \sim p(\cdot|s_t,a_t)$, and

$$\begin{aligned}
&H_t(s_t,a_t) \\
&:= \mathsf{E}[r(s_t,a_t,\omega_t) + \max_b Q_{k-1}(y,b)] - \max_b Q_{k-1}(s_{\mathbf{0}},b) \\
&\quad - (\mathsf{E}[r(s_t,a_t,\omega_t) + \max_b Q^*(y,b)] - \max_b Q^*(s_{\mathbf{0}},b)) \\
&= \int [\max_b Q_{k-1}(y,b) - \max_b Q^*(y,b)] q(dy|s_t,a_t) \\
&\quad - \max_b Q_{k-1}(s_{\mathbf{0}},b) + \max_b Q^*(s_{\mathbf{0}},b)
\end{aligned}$$

is the approximation error caused by replacing Q^* with Q_{k-1}. To see how this error propagates between successive interpolation times, we expand the recursion for $\varepsilon_t(s_l,a_l)$ starting from time $t_{k-1}+1$. This yields

$$\begin{aligned}
\varepsilon_{t_k}(s_l,a_l) &= U_\varepsilon(t_k : t_{k-1}) + U_B(t_k : t_{k-1}) \\
&\quad + U_W(t_k : t_{k-1}) + U_H(t_k : t_{k-1}),
\end{aligned}$$

where we have defined

$$\begin{aligned}
&U_\varepsilon(t_k : t_{k-1}) := [\prod_{i=t_{k-1}+1}^{t_k} (1-\eta_i(s_l,a_l))]\varepsilon_{t_{k-1}}(s_l,a_l), \\
&U_B(t_k : t_{k-1}) \\
&:= \sum_{i=t_{k-1}+1}^{t_k} [\prod_{j=i+1}^{t_k} (1-\eta_j(s_l,a_l))]\eta_i(s_l,a_l)B_i(s_l,a_l), \\
&U_W(t_k : t_{k-1}) \\
&:= \sum_{i=t_{k-1}+1}^{t_k} [\prod_{j=i+1}^{t_k} (1-\eta_j(s_l,a_l))]\eta_i(s_l,a_l)W_i(s_i,a_i), \\
&U_H(t_k : t_{k-1}) \\
&:= \sum_{i=t_{k-1}+1}^{t_k} [\prod_{j=i+1}^{t_k} (1-\eta_j(s_l,a_l))]\eta_i(s_l,a_l)H_i(s_i,a_i).
\end{aligned}$$

The convergence properties of the terms $U_\epsilon(t_k : t_{k-1})$, $U_B(t_k : t_{k-1})$, and $U_W(t_k : t_{k-1})$ are presented and analyzed in Lemmas 5, 7, and 8 below.

Lemma 5 If A1−A6 hold, then for each state-action pair $(s_l,a_l) \in \Lambda_{t_{k-1}}$, $U_\varepsilon(t_k : t_{k-1}) \to 0$ as $k \to \infty$ w.p.1.

Proof: By Lemma 4 and Lemma 1, we have $|\varepsilon_{t_{k-1}}(s_l,a_l)| \leq 2R_{max} + LD + c$. Thus,

$$\begin{aligned}
&|U_\varepsilon(t_k : t_{k-1})| \\
&= [\prod_{i=t_{k-1}+1}^{t_k} (1-\eta_i(s_l,a_l))]|\varepsilon_{t_{k-1}}(s_l,a_l)| \\
&\leq \exp(-\sum_{i=t_{k-1}+1}^{t_k} \eta_i(s_l,a_l))(2R_{max} + LD + c).
\end{aligned}$$

Notice that

$$\begin{aligned}
\sum_{i=t_{k-1}+1}^{t_k} \eta_i(s_l,a_l) &= \sum_{i=t_{k-1}+1}^{t_k} f(N_i^k(s_l,a_l))I_i(s_l,a_l) \\
&= \sum_{j=1}^{N_{t_k}(s_l,a_l) - N_{t_{k-1}}(s_l,a_l)} f(j).
\end{aligned}$$

By Lemma 2 and the condition $\sum_{i=1}^\infty f(i) = \infty$ (A4), we obtain $|U_\varepsilon(t_k : t_{k-1})| \to 0$ as $k \to \infty$ w.p.1 . ∎

The analysis of the remaining results relies on the following intermediate result.

Lemma 6 Let A4 hold. Then for each state-action pair (s,a) sampled by the algorithm and any positive integer l, $\sum_{i=l+1}^t [\prod_{j=i+1}^t (1-\eta_j(s,a))]\eta_i(s,a) \leq 1$ for all $t > l$. Further, if A2, A5, and A6 hold, then $\sum_{i=t_{k-1}+1}^{t_k} \left[\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))\right]\eta_i(s,a) \to 1$ as $k \to \infty$ w.p.1.

Proof: Let $Y_t = \sum_{i=l+1}^{t}[\prod_{j=i+1}^{t}(1-\eta_j(s,a))]\eta_i(s,a)$ for notational convenience. First, we show $Y_t \leq 1$ for all $t > l$ by induction. When $t = l+1$, it is clear that $Y_t \leq 1$. Suppose $Y_t \leq 1$ for some $t > l$, we obtain

$$\begin{aligned} Y_{t+1} &= \sum_{i=l+1}^{t+1}[\prod_{j=i+1}^{t+1}(1-\eta_j(s,a))]\eta_i(s,a) \\ &= \eta_{t+1}(s,a) + (1-\eta_{t+1}(s,a)) \\ &\quad \times \sum_{i=l+1}^{t}[\prod_{j=i+1}^{t}(1-\eta_j(s,a))]\eta_i(s,a) \\ &= \eta_{t+1}(s,a) + (1-\eta_{t+1}(s,a))Y_t \\ &\leq 1, \end{aligned}$$

where the inequality follows from the induction hypothesis. Thus we have $Y_t \leq 1$ for all $t > l$.

Next, let $X_t = \sum_{i=t_{k-1}+1}^{t}\left[\prod_{j=i+1}^{t}(1-\eta_j(s,a))\right]\eta_i(s,a)$ and $\Delta_t = |X_t - 1|$ where $t \in [t_{k-1}+1, t_k]$. Using a similar argument as above, we can easily obtain that $X_{t_k} = \eta_{t_k}(s,a) + (1-\eta_{t_k}(s,a))X_{t_k-1}$. It follows that

$$\begin{aligned} &\Delta_{t_k} = \Delta_{t_k-1}(1-\eta_{t_k}(s,a)) \\ &= \Delta_{t_{k-1}+1}\prod_{i=t_{k-1}+2}^{t_k}(1-\eta_i(s,a)) \\ &\leq \Delta_{t_{k-1}+1}\exp(-\sum_{i=t_{k-1}+2}^{t_k}\eta_i(s,a)) \\ &= \Delta_{t_{k-1}+1}\exp(-\sum_{i=t_{k-1}+1}^{t_k}\eta_i(s,a) + \eta_{t_{k-1}+1}(s,a)) \\ &= \Delta_{t_{k-1}+1}\exp(-\sum_{j=1}^{N_{t_k}(s,a)-N_{t_{k-1}}(s,a)}f(j) + \eta_{t_{k-1}+1}(s,a)) \\ &\leq \exp(-\sum_{j=1}^{N_{t_k}(s,a)-N_{t_{k-1}}(s,a)}f(j) + 1), \end{aligned}$$

which tends to zero as $k \to \infty$ w.p.1 by Lemma 2. This completes our proof. ∎

Lemma 7 If A1−A6 hold, then for each state-action pair $(s_l, a_l) \in \Lambda_{t_{k-1}}$, $U_B(t_k : t_{k-1}) \to 0$ as $k \to \infty$ w.p.1.

Proof: From the definition of $B_i(s_l,a_l)$ and Lemma 1, we have

$$I_i(s_l,a_l)|B_i(s_l,a_l)| \leq I_i(s_l,a_l)L_Q d(s_i,s_l) \leq L_Q r_i,$$

which tends to zero as $i \to \infty$ (due to $k \to \infty$) by A5(ii). Hence for any $\epsilon > 0$, there exist constants $N > 0$ and $M > 0$ such that $I_i(s_l,a_l)|B_i(s_l,a_l)| \leq \epsilon/2$ for all $k \geq N$ and $i \in (t_{k-1}+M, t_k]$. Thus we obtain that for a sufficiently large k w.p.1,

$$\begin{aligned} &|U_B(t_k : t_{k-1})| \\ &= \sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s_l,a_l))]\eta_i(s_l,a_l)|B_i(s_l,a_l)| \\ &\leq \sum_{i=t_{k-1}+1}^{t_{k-1}+M}[\prod_{j=i+1}^{t_k}(1-\eta_j(s_l,a_l))]\eta_i(s_l,a_l)|B_i(s_l,a_l)| \\ &\quad + \frac{\epsilon}{2}\sum_{i=t_{k-1}+M+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s_l,a_l))]\eta_i(s_l,a_l) \\ &\leq \sum_{i=t_{k-1}+1}^{t_{k-1}+M}[\prod_{j=i+1}^{t_k}(1-\eta_j(s_l,a_l))]\eta_i(s_l,a_l)|B_i(s_l,a_l)| + \frac{\epsilon}{2} \\ &\leq \sum_{i=t_{k-1}+1}^{t_{k-1}+M}\exp(-\sum_{j=i+1}^{t_k}\eta_j(s_l,a_l))|B_i(s_l,a_l)| + \frac{\epsilon}{2} \\ &\leq M\exp(-\sum_{j=t_{k-1}+M+1}^{t_k}\eta_j(s_l,a_l))(2R_{max}+2c) + \frac{\epsilon}{2} \\ &= 2M(R_{max}+c)\exp\big(-\sum_{j=t_{k-1}+1}^{t_k}f(N_j^k(s_l,a_l))I_j(s_l,a_l) \\ &\qquad + \sum_{j=t_{k-1}+1}^{t_{k-1}+M}f(N_j^k(s_l,a_l))I_j(s_l,a_l)\big) + \frac{\epsilon}{2} \\ &\leq 2M(R_{max}+c) \\ &\qquad \times \exp\big(-\sum_{j=1}^{N_{t_k}(s_l,a_l)-N_{t_{k-1}}(s_l,a_l)}f(j) + M\big) + \frac{\epsilon}{2} \\ &\leq \epsilon, \end{aligned}$$

where the second inequality comes from Lemma 6, the second last inequality is due to $f(N_j^k(s_l,a_l))I_j(s_l,a_l) \leq 1$, and the last inequality comes from the assumption A4 and Lemma 2 that implies $2M(R_{max}+c)\exp(-\sum_{j=1}^{N_{t_k}(s_l,a_l)-N_{t_{k-1}}(s_l,a_l)}f(j) + M) \leq \epsilon/2$ holds for k sufficiently large. Finally, since ϵ is arbitrary, we have $|U_B(t_k : t_{k-1})| \to 0$ as $k \to \infty$ w.p.1. ∎

Now, we show the convergence of the noise term $U_W(t_k : t_{k-1})$.

Lemma 8 If A1−A6 hold, then for each state-action pair $(s_l, a_l) \in \Lambda_{t_{k-1}}$, $U_W(t_k : t_{k-1}) \to 0$ as $k \to \infty$ w.p.1.

Proof: For any $k \geq 1$, consider the sequence $M_t := \sum_{i=t_{k-1}+1}^{t}\eta_i(s_l,a_l)W_i(s_i,a_i)$, $\forall t \in [t_{k-1}+1, t_k]$. Note that $\eta_t(s_l,a_l)$ is $\mathscr{F}_t$-measurable, we thus have

$$\begin{aligned} \mathsf{E}[M_t|\mathscr{F}_t] &= \sum_{i=t_{k-1}+1}^{t-1}\eta_i(s_l,a_l)W_i(s_i,a_i) + \eta_t(s_l,a_l) \\ &\quad \times \mathsf{E}\big[r(s_t,a_t,\omega_t) + \max_b Q_{k-1}(s_{t+1},b) \\ &\quad - \mathsf{E}[r(s_t,a_t,\omega_t) + \max_b Q_{k-1}(y,b)]|\mathscr{F}_t\big] \\ &= M_{t-1}. \end{aligned}$$

In addition,

$$\begin{aligned} \mathsf{E}[M_t^2] &= \mathsf{E}[(\sum_{i=t_{k-1}+1}^{t}\eta_i(s_l,a_l)W_i(s_l,a_l))^2] \\ &= \mathsf{E}[\sum_{i=t_{k-1}+1}^{t}\eta_i^2(s_l,a_l)W_i^2(s_l,a_l)], \end{aligned}$$

where the last step follows from the fact that for all $i < j$, the cross terms

$$\begin{aligned}&\mathsf{E}[\eta_i(s_l,a_l)\eta_j(s_l,a_l)W_i(s_l,a_l)W_j(s_l,a_l)]\\&=\mathsf{E}[\eta_i(s_l,a_l)\eta_j(s_l,a_l)W_i(s_l,a_l)\mathsf{E}[W_j(s_j,a_j)|\mathscr{F}_j]]\\&=0.\end{aligned}$$

By A1 and Lemma 4, we have $|W_t(s,a)| \le 4(R_{max}+LD)$ for all t and (s,a). Consequently, due to the condition $\sum_{j=1}^{\infty} f^2(j) < \infty$ (A4),

$$\begin{aligned}\mathsf{E}[M_t^2] &\le 16(R_{max}+LD)^2\mathsf{E}[\sum_{i=t_{k-1}+1}^{t}\eta_i^2(s_l,a_l)]\\&= 16(R_{max}+LD)^2\mathsf{E}[\sum_{j=1}^{N_t(s_l,a_l)-N_{t_{k-1}}(s_l,a_l)} f^2(j)]\\&< \infty.\end{aligned}$$

Hence $\{M_t\}$ is an L^2-bounded martingale.

Recall that $t_k - t_{k-1} \to \infty$ as $k \to \infty$. Thus, according to the martingale convergence theorem, for any $\epsilon > 0$, we can find constants K and T such that for all $k \ge K$, there exists a finite random variable M_∞ satisfying $|M_t - M_\infty| < \epsilon$, $\forall t \in [t_{k-1}+T, t_k]$ w.p.1. Thus for any $k \ge 1$, we have

$$\begin{aligned}&U_W(t_k : t_{k-1})\\&= \sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s_l,a_l))]\eta_i(s_l,a_l)W_i(s_i,a_i)\\&= \prod_{j=t_{k-1}+1}^{t_k}(1-\eta_j(s_l,a_l))\\&\quad\times\sum_{i=t_{k-1}+1}^{t_k}\frac{1}{\prod_{j=t_{k-1}+1}^{i}(1-\eta_j(s_l,a_l))}\eta_i(s_l,a_l)W_i(s_i,a_i)\\&= \frac{1}{b_{t_k}}\sum_{i=t_{k-1}+1}^{t_k} b_i\eta_i(s_l,a_l)W_i(s_i,a_i),\end{aligned}$$

where $b_i := \frac{1}{\prod_{j=t_{k-1}+1}^{i}(1-\eta_j(s_l,a_l))}$. It can be observed that $0 < b_i \le b_{i+1}$ and $b_i \to \infty$ as $t_k - t_{k-1} \to \infty$ (due to $k \to \infty$). Based on the fact that $\sum_{i=t_{k-1}+1}^{t_k}\frac{b_i\eta_i(s_l,a_l)W_i(s_i,a_i)}{b_i} = M_{t_k} < \infty$ for all k w.p.1 and applying the Kronecker's lemma (see, e.g., [40]) in a path-wise manner, we obtain

$$\frac{1}{b_{t_k}}\sum_{i=t_{k-1}+1}^{t_k} b_i\eta_i(s_l,a_l)W_i(s_i,a_i) \to 0$$

as $k \to \infty$ w.p.1, which completes our proof. ∎

Finally, we are ready to present the proof of the main convergence result Theorem 1.

Proof of Theorem 1 By the definition of $Q^*(s,a)$ and Lemma 4, we have $\sup_{s\in S}\max_a|Q^*(s,a)| \le R_{max}+c$ and $\sup_{s\in S}\max_a|Q_k(s,a)| \le R_{max}+2LD$, $\forall k > 0$. Hence

$$\begin{aligned}&\|Q_k(s,a)-Q^*(s,a)\|_{S\times A}\\&=\sup_{s\in S}\max_a(Q_k(s,a)-Q^*(s,a))\\&\quad-\inf_{s\in S}\min_a(Q_k(s,a)-Q^*(s,a))\qquad(10)\\&\le 2\sup_{s\in S}\max_a|Q_k(s,a)-Q^*(s,a)|\\&\le 2(2R_{max}+2LD+c).\end{aligned}$$

Next, we proceed by using an inductive argument and suppose that on each sample path ϖ, there exists a constant G and time $\tau_j > 0$ such that $\|Q_k(s,a)-Q^*(s,a)\|_{S\times A} \le G$ for all $k \ge \tau_j$. In what follows, we show that we can find another time $\tau_{j+1} > \tau_j$ and a constant $\zeta \in (0,1)$ satisfying $\|Q_k(s,a)-Q^*(s,a)\|_{S\times A} \le \zeta G$ for all $k \ge \tau_{j+1}$. Repeating this argument in turn shows the convergence of $\|Q_k(s,a)-Q^*(s,a)\|_{S\times A}$ to 0.

By A2(i), there exists a positive constant β' such that $\beta+\beta' < 1$. Let $r = \frac{\xi G}{4(L+L_Q)}$ where $\xi \in (0, 1-\beta-\beta')$ is a given constant. Define the event $\Omega_a = \{\lim_{k\to\infty}\sup_s d(s,\Lambda_{t_k}(a)) = 0\}$, $\forall a \in A$. For each sample path $\varpi \in \cap_{a\in A}\Omega_a$, there existis some τ' such that $S \subseteq \cup_{s\in\Lambda_{t_{\tau'}}(a)}B(s,r)$, $\forall a \in A$. Let $\tau = \max\{\tau', \tau_j\}$. Clearly, $S \subseteq \cup_{s\in\Lambda_{t_\tau}(a)}B(s,r)$ for any $a \in A$.

For any state-action pair $(s,a) \in \Lambda_{t_\tau}$ and for all $k \ge \tau+1$, consider the recursion

$$\begin{aligned}\varepsilon_{t_k}(s,a) &= (1-\eta_{t_k}(s,a))\varepsilon_{t_k-1}(s,a)+\eta_{t_k}(s,a)\\&\quad\times\big(B_{t_k}(s,a)+W_{t_k}(s_{t_k},a_{t_k})+H_{t_k}(s_{t_k},a_{t_k})\big)\\&= U_\varepsilon(t_k:t_{k-1})+U_B(t_k:t_{k-1})\\&\quad+U_W(t_k:t_{k-1})+U_H(t_k:t_{k-1}).\end{aligned}$$

Let $\Omega_\varepsilon = \{\lim_{k\to\infty}U_\varepsilon(t_k : t_{k-1}) = 0\}$, $\Omega_{sp} = \{\lim_{k\to\infty}\sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a) = 1\}$, $\Omega_B = \{\lim_{k\to\infty}U_B(t_k : t_{k-1}) = 0\}$ and $\Omega_W = \{\lim_{k\to\infty}U_W(t_k : t_{k-1}) = 0\}$. From Lemmas 5−8, for each sample path $\varpi \in \cap_{a\in A}\Omega_a\cap\Omega_\varepsilon\cap\Omega_{sp}\cap\Omega_B\cap\Omega_W$, there exists an interpolation time $\tau'_{j+1} \ge \tau$ such that for all $(s,a) \in \Lambda_{t_\tau}$ and $k \ge \tau'_{j+1}$,

$$\begin{aligned}&\varepsilon_{t_k}(s,a)\\&\le\frac{\xi G}{4}+\sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a)H_i(s_i,a_i)\end{aligned}$$

and

$$\begin{aligned}&\varepsilon_{t_k}(s,a)\\&\ge-\frac{\xi G}{4}+\sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a)H_i(s_i,a_i).\end{aligned}$$

Next for any $s \in S$ and $b \in A$, denote $s_b = \arg\min_{s'\in\Lambda_{t_\tau}(b)}d(s,s')$. Since $S \subseteq \cup_{s'\in\Lambda_{t_\tau}(b)}B(s',r)$, we have $d(s,s_b) \le r$. It follows that $Q_k(s,b)-Q^*(s,b) = Q_k(s,b)-Q_k(s_b,b)+Q_k(s_b,b)-Q^*(s_b,b)+Q^*(s_b,b)-Q^*(s,b) = Q_k(s,b)-Q_k(s_b,b)+\tilde{Q}_{t_k}(s_b,b)-Q^*(s_b,b)+Q^*(s_b,b)-Q^*(s,b)$ for all $k \ge \tau'_{j+1}+1$, where in the last step we have used the fact that $Q_k(s_b,b) = \tilde{Q}_{t_k}(s_b,b)$ due to the interpolation property of Q_k. Hence we have

$$\begin{aligned}&Q_k(s,b)-Q^*(s,b)\\&\le|Q_k(s,b)-Q_k(s_b,b)|+\tilde{Q}_{t_k}(s_b,b)-Q^*(s_b,b)\\&\quad+|Q^*(s_b,b)-Q^*(s,b)|\\&\le(L+L_Q)r+\frac{\xi G}{4}\end{aligned}$$

$$+\sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s_b,b))]\eta_i(s_b,b)H_i(s_i,a_i)$$

and

$$\begin{aligned}&Q_k(s,b)-Q^*(s,b)\\&\geq -|Q_k(s,b)-Q_k(s_b,b)|+\tilde{Q}_{t_k}(s_b,b)-Q^*(s_b,b)\\&\quad-|Q^*(s_b,b)-Q^*(s,b)|\\&\geq -(L+L_Q)r-\frac{\xi G}{4}\\&\quad+\sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s_b,b))]\eta_i(s_b,b)H_i(s_i,a_i).\end{aligned}$$

This suggests that

$$\begin{aligned}&\sup_{s\in S}\max_b(Q_k(s,b)-Q^*(s,b))\leq(L+L_Q)r+\frac{\xi G}{4}+\\&\max_{(s',a')\in\Lambda_{t_\tau}}\sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s',a'))]\eta_i(s',a')H_i(s_i,a_i)\end{aligned}\tag{11}$$

and

$$\begin{aligned}&\inf_{s\in S}\min_b(Q_k(s,b)-Q^*(s,b))\geq-(L+L_Q)r-\frac{\xi G}{4}+\\&\min_{(s',a')\in\Lambda_{t_\tau}}\sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s',a'))]\eta_i(s',a')H_i(s_i,a_i).\end{aligned}\tag{12}$$

Next we derive a bound for $\|\sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a)H_i(s_i,a_i)\|_{\Lambda_{t_\tau}}$. To this end, we note that for any two state-action pairs (s',a') and (s'',a'') in $S\times A$ and for any $k>0$ and $t\in[t_{k-1}+1,t_k]$,

$$\begin{aligned}&H_t(s',a')-H_t(s'',a'')\\&=\int[\max_b Q_{k-1}(y,b)-\max_b Q^*(y,b)]q(dy|s',a')\\&\quad-\int[\max_b Q_{k-1}(y,b)-\max_b Q^*(y,b)]q(dy|s'',a'')\\&=\int[\max_b Q_{k-1}(y,b)-\max_b Q^*(y,b)]\nu(dy),\end{aligned}$$

where ν is a finite signed measure on S defined by $\nu(\cdot):=q(\cdot|s',a')-q(\cdot|s'',a'')$. By the Hahn-Jordan decomposition theorem, there exist two disjoint measurable sets S^+ and S^- with $S^+\cup S^-=S$ such that

$$\|\nu\|_{\mathsf{TV}}=\nu(S^+)-\nu(S^-)\leq 2\beta,$$

where $\nu(S^+)>0$ and $\nu(S^-)<0$ (by A2(i)). On the other hand, since $\nu(S)=\nu(S^+)+\nu(S^-)=0$, we have that $\nu(S^+)\leq\beta$. It follows that

$$\begin{aligned}&\int[\max_b Q_{k-1}(y,b)-\max_b Q^*(y,b)]\nu(dy)\\&=\int_{S^+}[\max_b Q_{k-1}(y,b)-\max_b Q^*(y,b)]d\nu\\&\quad+\int_{S^-}[\max_b Q_{k-1}(y,b)-\max_b Q^*(y,b)]d\nu\\&\leq\int_{S^+}\sup_y\max_b[Q_{k-1}(y,b)-Q^*(y,b)]d\nu\\&\quad-\int_{S^+}\inf_y\min_b[Q_{k-1}(y,b)-Q^*(y,b)]d\nu\\&\quad+\int_{S^+}\inf_y\min_b[Q_{k-1}(y,b)-Q^*(y,b)]d\nu\\&\quad+\int_{S^-}\inf_y\min_b[Q_{k-1}(y,b)-Q^*(y,b)]d\nu\\&\leq\nu(S^+)\|Q_{k-1}(y,b)-Q^*(y,b)\|_{S\times A}\\&\quad+\inf_y\min_b[Q_{k-1}(y,b)-Q^*(y,b)]\cdot\nu(S)\\&\leq\beta G.\end{aligned}$$

Therefore,

$$\begin{aligned}&H_t(s',a')-H_t(s'',a'')\\&=\int[\max_b Q_{k-1}(y,b)-\max_b Q^*(y,b)]\nu(dy)\\&\leq\beta G.\end{aligned}$$

From the arbitrariness of (s',a') and (s'',a''), we know for all $t>0$,

$$\|H_t(s,a)\|_{S\times A}\leq\beta G.\tag{13}$$

We now use (13) to establish a bound for

$$\|\sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a)H_i(s_i,a_i)\|_{\Lambda_{t_\tau}}.$$

By Lemma 6, the limit of the sequence

$$\left\{\sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a)\right\}_{k=\tau'_{j+1}+1}^{\infty}$$

is 1 for all $(s,a)\in\Lambda_{t_\tau}$. Thus there exists an interpolation time $\tau_{j+1}>\tau'_{j+1}$ such that for all $k\geq\tau_{j+1}$ and any $(s,a)\in\Lambda_{t_\tau}$,

$$\begin{aligned}&|\sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a)-1|\\&\leq\frac{\beta' G}{4\times(2R_{max}+2LD+c)}.\end{aligned}$$

It follows that for any two state-action pairs (s',a') and (s'',a'') in Λ_{t_τ},

$$\begin{aligned}&|\sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s',a'))]\eta_i(s',a')\\&\quad-\sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s'',a''))]\eta_i(s'',a'')|\\&\leq|\sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s',a'))]\eta_i(s',a')-1|\\&\quad+|1-\sum_{i=t_{k-1}+1}^{t_k}[\prod_{j=i+1}^{t_k}(1-\eta_j(s'',a''))]\eta_i(s'',a'')|\\&\leq\frac{\beta' G}{2\times(2R_{max}+2LD+c)}.\end{aligned}$$

Note that since H_i is fixed during time interval $[t_{k-1}+1, t_k]$ (due to using the same function approximator), we denote it as $\bar{H}_k$. Therefore, when $k \geq \tau_{j+1}$, we have

$$
\begin{aligned}
&\| \sum_{i=t_{k-1}+1}^{t_k} [\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a)H_i(s_i,a_i)\|_{\Lambda_{t_\tau}} \\
&= \max_{(s,a)\in\Lambda_{t_\tau}} \sum_{i=t_{k-1}+1}^{t_k} [\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a)\bar{H}_k(s_i,a_i) \\
&\quad - \min_{(s,a)\in\Lambda_{t_\tau}} \sum_{i=t_{k-1}+1}^{t_k} [\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a)\bar{H}_k(s_i,a_i) \\
&\leq \max_{(s,a)\in\Lambda_{t_\tau}} \Big\{ \\
&\quad \sup_{(s',a')\in S\times A} \bar{H}_k(s',a') \sum_{i=t_{k-1}+1}^{t_k} [\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a)\Big\} \\
&\quad - \min_{(s,a)\in\Lambda_{t_\tau}} \Big\{ \\
&\quad \inf_{(s',a')\in S\times A} \bar{H}_k(s',a') \sum_{i=t_{k-1}+1}^{t_k} [\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a)\Big\} \\
&\leq \max_{(s,a)\in\Lambda_{t_\tau}} \Big\{ (\sup_{(s',a')\in S\times A} \bar{H}_k(s',a') - \inf_{(s',a')\in S\times A} \bar{H}_k(s',a')) \\
&\qquad \times \sum_{i=t_{k-1}+1}^{t_k} [\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a)\Big\} \\
&\quad + \Big\| \inf_{(s',a')\in S\times A} \bar{H}_k(s',a') \\
&\qquad \times \sum_{i=t_{k-1}+1}^{t_k} [\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a)\Big\|_{\Lambda_{t_\tau}} \\
&\leq \beta G + |\inf_{(s',a')\in S\times A} \bar{H}_k(s',a')| \\
&\qquad \times \| \sum_{i=t_{k-1}+1}^{t_k} [\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a)\|_{\Lambda_{t_\tau}} \\
&\leq \beta G \\
&\quad + 2\times(2R_{max}+2LD+c)\times \frac{\beta' G}{2\times(2R_{max}+2LD+c)} \\
&= (\beta+\beta')G. \qquad (14)
\end{aligned}
$$

By combining the result with (10), (11) and (12), we obtain that

$$
\begin{aligned}
&\|Q_k(s,b)-Q^*(s,b)\|_{S\times A} \\
&= \sup_{s\in S}\max_b (Q_k(s,b)-Q^*(s,b)) \\
&\quad - \inf_{s\in S}\min_b (Q_k(s,b)-Q^*(s,b)) \\
&\leq 2(L+L_Q)r + \frac{\xi G}{2} \\
&\quad + \| \sum_{i=t_{k-1}+1}^{t_k} [\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a)H_i(s_i,a_i)\|_{\Lambda_{t_\tau}} \\
&= \xi G + \| \sum_{i=t_{k-1}+1}^{t_k} [\prod_{j=i+1}^{t_k}(1-\eta_j(s,a))]\eta_i(s,a)H_i(s_i,a_i)\|_{\Lambda_{t_\tau}} \\
&\leq (\xi+\beta+\beta')G,
\end{aligned}
$$

where the last inequality comes from (14).

By Lemma 3, 5, 6, 7, and 8, we have $\mathsf{P}(\cap_{a\in A}\Omega_a \cap \Omega_\varepsilon \cap \Omega_{sp} \cap \Omega_B \cap \Omega_W) = 1$. Let $\zeta = \xi+\beta+\beta'$, then we have $\|Q_k(s,b)-Q^*(s,b)\|_{S\times A} \leq \zeta G$ for all $k \geq \tau_{j+1}$ w.p.1. This completes our proof. ■

IV. An Illustrative Example

We illustrate our algorithm by applying it to a machine replacement example. The original problem is frequently used as a testbed to evaluate the performance of algorithms for discounted MDPs (see, e.g., [41], [42], [43]) and is adapted to an average-reward setting in [32] and [23]. We consider a four-dimensional version of the problem, where the state variable $s = (s_1, s_2, s_3, s_4)$ measures the accumulated utilization of four independent machines. For each machine, there are two admissible actions: keep the current machine (**K**) or replace it with a new one (**R**). Thus the action set $A = \{(a_1,a_2,a_3,a_4) \in (\mathbf{K}\times\mathbf{R})^4\}$ contains 16 different actions. For each machine $i \in \{1,2,3,4\}$, the transition density is given by

$$
p(s_i'|s_i,a_i) = \begin{cases} \varphi_i \exp(-\varphi_i(s_i'-s_i)), & s_i' \geq s_i, a_i = \mathbf{K}; \\ \varphi_i \exp(-\varphi_i s_i'), & s_i' \geq 0, a_i = \mathbf{R}; \\ 0, & \text{otherwise.} \end{cases}
$$

The immediate reward produced by any machine i is given by $r(s,\mathbf{K}) = \kappa_i s_i$ and $r(s,\mathbf{R}) = \psi_i$. Thus, the AROE for machine i can be stated as follows:

$$
J_i^* + v_i^*(s) = \max\{T_{k,i}, T_{r,i}\},
$$

where we have defined

$$
\begin{aligned}
T_{k,i} &:= -\kappa_i s_i + \int_0^\infty \varphi_i \exp(-\varphi_i(s_i'-s_i)) v_i^*(s_i') ds_i', \\
T_{r,i} &:= -\psi_i + \int_0^\infty \varphi_i \exp(-\varphi_i s_i') v_i^*(s_i') ds_i'.
\end{aligned}
$$

The goal is to maximize the expectation of the long-run average reward. For comparison with the theoretical optimal solution, we note that the optimal value function v_i^* has a closed-form expression given by

$$
v_i^*(s) = \begin{cases} -\kappa_i(1+\varphi_i\bar{s}_i)s_i + \dfrac{\kappa_i\varphi_i}{2}s_i^2, & 0 \leq s_i \leq \bar{s}_i; \\ -\kappa_i\bar{s}_i - \dfrac{\kappa_i\varphi_i}{2}\bar{s}_i^2, & \text{otherwise,} \end{cases}
$$

where $\bar{s}_i$ is a unique threshold such that the optimal action is **K** whenever $s_i \in [0,\bar{s}_i]$ and **R** if $s_i > \bar{s}_i$; see, e.g., Section 5 of [23]. Therefore, the optimal value function v^* under our setting is simply given by the sum of the four v_i^*'s. In our computational experiment, we set the parameters as follows: $\varphi_1 = 2/3$, $\kappa_1 = 3$, $\psi_1 = 15$, $\varphi_2 = 4/5$, $\kappa_2 = 2$, $\psi_2 = 17$, $\varphi_3 = 3/4$, $\kappa_3 = 7$, $\psi_3 = 5$, $\varphi_4 = 3/2$, $\kappa_4 = 10$, $\psi_4 = 20$, which leads to the thresholds $\bar{s}_1 \approx 2.65$, $\bar{s}_2 \approx 3.53$, $\bar{s}_3 \approx 0.59$, $\bar{s}_4 \approx 1.10$.

To make the state space compact, we adopt the same approach used in [42] by setting an upper bound $s_{max} = 5$ on the state values. We assume that if the i-th element of the next state happens to be larger than s_{max} then the i-th machine is

replaced immediately, and a new state is drawn as if action $\mathbf{R}$ were taken for the i-th machine in the previous step. It can be seen that $\int_{s_{max}}^{\infty} p(s'|s,\mathbf{R})ds'$ is almost negligible under our parameter setting and thus the optimal value function of the modified problem closely matches that of the original problem.

The proposed relative Q-learning (RQ) algorithm is implemented with the following parameter values: learning rate $\alpha_t(s,a) = N_t^k(s,a)^{-0.501}$, interpolation times $t_k = \sum_{i=1}^{k} i^2$, reference state $s_{\mathbf{0}} = (2.5, 2.5, 2.5, 2.5)$. We examine the algorithm with two different shrinking ball radii: a logarithmic decaying radius $r_t = C_1/\log(1+t)$ and a polynomially decaying radius $r_t = C_2(t+1)^{-\gamma}$, where C_1 and C_2 are positive constants. We denote the algorithms corresponding to these two choices as RQ-log and RQ-poly, respectively. It is easy to verify that in the latter case, Assumptions A5(i) and A5(ii) are satisfied with $0 < \gamma < 1/12$. So our implementation of RQ-poly is based on setting $\gamma = 0.083 \approx 1/12$. Regarding C_1 and C_2, we recommend choosing their values to make the radius approximately 10% of the diameter of the state space when the algorithm terminates. The numerical results reported here are based on the choice $C_1 = 13$ and $C_2 = 3$. The learning policy of RQ is taken to be an ϵ-greedy policy with $\epsilon = 0.1$, that is, choosing the greedy action with respect to Q_{k-1} with probability $1-\epsilon$ and selecting a random action with probability ϵ at every iteration step. The function approximator is constructed by using the stochastic kriging method (see, e.g., [44], [45]). As suggested in, e.g., [46], [47], we use the Matérn kernel as the covariance function in the kriging model. The initial state is set to $(4,4,4,4)$.

In addition to RQ, we have also applied three other methods: a discretization-based heuristic variant of RVI Q-learning, the ERVL algorithm proposed in [32], and the ARVL method proposed in [23]. In the first method, we combine the soft-state aggregation method of [30] with RVI Q-learning to construct an asynchronous online algorithm called RVIQ-SSA. It uses the transition samples generated from a learning policy to iteratively estimate the Q-function values at a given set of clusters (aggregate states). In the experiments, those clusters are obtained by discretizing the state space using a grid size of 1.0 along each dimension, and each encountered state s belongs to the jth cluster with probability $P(j|s) = \frac{\exp(-\|s-j\|^2/0.01)}{\sum_{j'}\exp(-\|s-j'\|^2/0.01)}$. The Q-function estimator is then constructed in the form of a weighted sum $\sum_j P(j|s)\hat{Q}(j,a)$ for all (s,a), where $\hat{Q}(j,a)$ is an estimate of the Q-value at each cluster-action pair. In ERVL and ARVL, the iterates are (estimated) value functions. At each iteration, both algorithms sample N states uniformly over the state space. For each sampled state action combination, M next states are obtained by simulating the transition dynamics. In particular, since there are 16 actions in this example, each state-action combination is repeatedly simulated $16M$ times. These samples are used to approximate the expectation involved in the AROE (assuming the immediate rewards are deterministic) through either direct sample average approximation (in ERVL) or kernel density estimation (in ARVL). A synchronous approximate value iteration step is then carried out, and an estimated value function is subsequently constructed based on the nearest neighbor averaging technique. In our implementation, we have used $N = 100$, $M = 5$, and the total number of algorithm iterations is set to $K = 50$. Other hyper-parameters, including the bandwidth used in Gaussian kernel density estimation (used in ARVL) and the number of nearest neighbors, are taken to be the same as in [32] and [23]. To allow for a fair comparison with ERVL and ARVL, the numbers of iterations of RQ and RVIQ-SSA are set to $N \times 16M \times K = 100 \times 80 \times 50 = 400000$, which corresponds to the total number of transition samples consumed by ERVL and ARVL.

Since all comparison algorithms are randomized, we perform ten independent replications for each algorithm and denote by $\tilde{v}_{alg}^i$ the estimated value function obtained in the i-th run of an algorithm, where $i = 1, 2, \ldots, 10$ and $alg \in \{$RQ-poly, RQ-log, RVIQ-SSA, ERVL, ARVL$\}$. Table I shows the spans of the differences between the optimal value function v^* and the estimated value functions obtained by different comparison algorithms upon termination. Note that the results are averaged over ten replications, i.e., $\frac{1}{10}\sum_{i=1}^{10}\|\tilde{v}_{alg}^i(s) - v^*(s)\|_D$, where D is a set of 1024 low-discrepancy states selected by using the Sobol sequence on the four-dimensional state space (cf., e.g., Chapter 5 of [48]). Fig. 2 illustrates the convergence behavior of the five algorithms by plotting the averaged span semi-norm values with respect to the number of samples used.

TABLE I

SPANS OF VALUE FUNCTION APPROXIMATES OBTAINED BY RQ-POLY, RQ-LOG, RVIQ-SSA, ERVL, AND ARVL (MEANS AND STAND ERRORS BASED ON 10 INDEPENDENT REPLICATIONS).

RQ-poly	RQ-log	RVIQ-SSA	ERVL	ARVL
16.21(0.43)	22.57(0.83)	32.52(0.64)	46.24(1.05)	47.95(0.34)

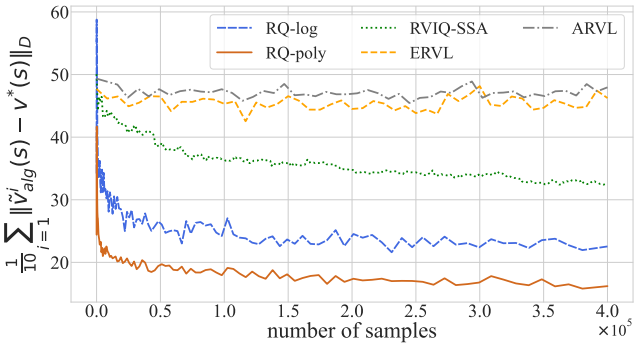

Fig. 2. Performance of comparison algorithms in averaged span semi-norm.

We see that the proposed algorithm outperforms RVIQ-SSA, ERVL, and ARVL and yields the smallest span semi-norm values as data accumulate. Test results indicate comparable performance of ERVL and ARVL. Since both algorithms use a large number of transition samples at each step, whereas RQ works with a single sample trajectory, they show a faster initial improvement than RQ. However, both ERVL and ARVL stop making improvements during early iterations. We conjecture that this is mainly due to the discarding of past sampling information in these algorithms, so that the

constant number of transition samples used at each step (i.e., constant values of N and M) may result in an estimation error in density estimation/sample average approximation that cannot be eliminated across the iterations. RQ, on the other hand, is an online algorithm that fully retains past learning data. Compared to RQ, ERVL, and ARVL, the advantage of RVIQ-SSA lies in its computational and memory efficiencies because the algorithm uses a constant number of aggregate states and does not require storing historical transition data. However, the use of the weighted average in the Q-function approximator could lead to substantial bias in its estimation. The performance of ERVL and ARVL could be improved by increasing the per-iteration sample size; however, that would result in a reduced number of algorithm iterations under a given computing budget.

V. Conclusion

In this paper, motivated by RVI Q-learning, we have proposed a relative Q-learning algorithm for solving average-reward MDPs with continuous state spaces in a model-free online manner. In particular, to achieve the transition from the commonly studied discrete-state setting to a continuous-state domain, the algorithm integrates an asynchronous online averaging procedure with interpolation-based function approximation. The online averaging procedure allows the estimation error at a visited state-action pair to be eliminated by averaging Q-value estimates at all pairs that are within its neighborhood; whereas the function approximator offers the flexibility in approximating the Q-function over the entire domain by interpolating historical data collected during the learning process. Under appropriate conditions, we have shown the almost sure (uniform) convergence of the sequence of function approximators to the optimal Q-function, modulo a constant value that does not affect the determination of the optimal policy. To our knowledge, this is the first online Q-learning based algorithm for solving continuous-state average reward problems with a strong convergence guarantee. A simple benchmark example has also been presented to illustrate the algorithm, indicating its promising performance compared to some of the existing methods.

References

[1] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Cambridge, 1989.

[2] H. S. Chang, J. Hu, M. C. Fu, and S. I. Marcus, *Simulation-Based Algorithms for Markov Decision Processes*, 2nd ed. London, UK: Springer, 2013.

[3] L. Busoniu, R. Babuska, B. D. Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*, 1st ed. Boca Raton, USA: CRC Press, 2010.

[4] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. Hoboken, USA: John Wiley & Sons, 2014.

[5] J. G. Dai and M. Gluzman, "Queueing network controls via deep reinforcement learning," *Stochastic Systems*, vol. 12, no. 1, pp. 30–67, 2022.

[6] E. A. Feinberg and Y. Liang, "On the optimality equation for average cost Markov decision processes and its validity for inventory control," *Annals of Operations Research*, vol. 317, no. 2, pp. 569–586, October 2022.

[7] P. Tadepalli and D. Ok, "Model-based average reward reinforcement learning," *Artificial Intelligence*, vol. 100, no. 1, pp. 177–224, 1998.

[8] G. Qu, Y. Lin, A. Wierman, and N. Li, "Scalable multi-agent reinforcement learning for networked systems with average reward," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 2074–2086.

[9] A. Arapostathis, V. S. Borkar, E. Fernández-Gaucherand, M. K. Ghosh, and S. I. Marcus, "Discrete-time controlled markov processes with average cost criterion: A survey," *SIAM Journal on Control and Optimization*, vol. 31, no. 2, pp. 282–344, 1993.

[10] C.-Y. Wei, M. Jafarnia-Jahromi, H. Luo, H. Sharma, and R. Jain, "Model-free reinforcement learning in infinite-horizon average-reward markov decision processes," in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML'20. JMLR.org, 2020.

[11] A. Schwartz, "A reinforcement learning method for maximizing undiscounted rewards," in *Proceedings of the Tenth International Conference on International Conference on Machine Learning*, ser. ICML'93. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993, p. 298–305.

[12] S. P. Singh, "Reinforcement learning algorithms for average-payoff markovian decision processes," in *Proceedings of the Twelfth AAAI National Conference on Artificial Intelligence*, ser. AAAI'94. AAAI Press, 1994, p. 700–705.

[13] J. Abounadi, D. Bertsekas, and V. S. Borkar, "Learning algorithms for markov decision processes with average cost," *SIAM Journal on Control and Optimization*, vol. 40, no. 3, pp. 681–698, 2001.

[14] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. II*, 3rd ed. Nashua, USA: Athena Scientific, 2007.

[15] S. Yang, Y. Gao, B. An, H. Wang, and X. Chen, "Efficient average reward reinforcement learning using constant shifting values," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[16] K. E. Avrachenkov and V. S. Borkar, "Whittle index based q-learning for restless bandits with average reward," *Automatica*, vol. 139, p. 110186, 2022.

[17] Y. Wan, A. Naik, and R. S. Sutton, "Learning and planning in average-reward markov decision processes," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 10 653–10 662.

[18] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds., vol. 12. MIT Press, 1999.

[19] P. Marbach and J. Tsitsiklis, "Simulation-based optimization of markov reward processes," *IEEE Transactions on Automatic Control*, vol. 46, no. 2, pp. 191–209, Feb 2001.

[20] S. Mahadevan, "Average reward reinforcement learning: Foundations, algorithms, and empirical results," *Machine Learning*, vol. 22, no. 1-3, pp. 159–195, 1996.

[21] V. Dewanto, G. Dunn, A. Eshragh, M. Gallagher, and F. Roosta, "Average-reward model-free reinforcement learning: a systematic review and literature mapping," 2021.

[22] S. Baumert and R. L. Smith, "Pure random search for noisy objective functions," University of Michigan, Tech. Rep., 2002.

[23] H. Sharma, M. Jafarnia-Jahromi, and R. Jain, "Approximate relative value learning for average-reward continuous state mdps," in *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, ser. Proceedings of Machine Learning Research, R. P. Adams and V. Gogate, Eds., vol. 115. Tel Aviv, Israel: PMLR, 22–25 Jul 2020, pp. 956–964.

[24] C.-Y. Wei, M. Jafarnia Jahromi, H. Luo, and R. Jain, "Learning infinite-horizon average-reward mdps with linear function approximation," in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Banerjee and K. Fukumizu, Eds., vol. 130. PMLR, 13–15 Apr 2021, pp. 3007–3015.

[25] J. Hu, X. Yang, J.-Q. Hu, and Y. Peng, "A q-learning algorithm for markov decision processes with continuous state spaces," *submitted to Systems & Control Letters*, 2022.

[26] A. M. Devraj and S. P. Meyn, "Q-learning with uniformly bounded variance," *IEEE Transactions on Automatic Control*, vol. 67, no. 11, pp. 5948–5963, 2022.

[27] R. Ortner, "Pseudometrics for state aggregation in average reward markov decision processes," in *Algorithmic Learning Theory*, M. Hutter, R. A. Servedio, and E. Takimoto, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 373–387.

[28] ——, "Adaptive aggregation for reinforcement learning in average reward Markov decision processes," *Annals of Operations Research*, vol. 208, no. 1, pp. 321–336, September 2013.
[29] D. Ormoneit and P. Glynn, "Kernel-based reinforcement learning in average-cost problems," *IEEE Transactions on Automatic Control*, vol. 47, no. 10, pp. 1624–1636, Oct 2002.
[30] S. Singh, T. Jaakkola, and M. Jordan, "Reinforcement learning with soft state aggregation," *Advances in neural information processing systems*, vol. 7, 1994.
[31] D. Shah and Q. Xie, "Q-learning with nearest neighbors," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018, pp. 3111–3121.
[32] H. Sharma, R. Jain, and A. Gupta, "An empirical relative value learning algorithm for non-parametric mdps with continuous state space," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 1368–1373.
[33] O. Hernández-Lerma, *Adaptive Markov Control Processes*, 1st ed. New York, USA: Springer, 1989.
[34] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*, 1st ed., ser. Texts and Readings in Mathematics ; 48. Gurgaon: Hindustan Book Agency, 2008.
[35] R. Ortner and D. Ryabko, "Online regret bounds for undiscounted continuous reinforcement learning," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.
[36] J. Qian, R. Fruit, M. Pirotta, and A. Lazaric, "Exploration bonus for regret minimization in discrete and continuous average reward mdps," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019, pp. 4890–4899.
[37] E. A. Feinberg, P. O. Kasyanov, and N. V. Zadoianchuk, "Average cost markov decision processes with weakly continuous transition probabilities," *Mathematics of Operations Research*, vol. 37, no. 4, pp. 591–607, 2012.
[38] C. Szepesvári and W. D. Smart, "Interpolation-based q-learning," in *Proceedings of the 21st International Conference on Machine Learning*, 2004, pp. 791–798.
[39] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári, "Convergence results for single-step on-policy reinforcement-learning algorithms," *Machine learning*, vol. 38, no. 3, pp. 287–308, 2000.
[40] R. Durrett, *Probability: theory and examples*, 5th ed. Cambridge, UK: Cambridge university press, 2019, vol. 49.
[41] J. Rust, "Chapter 14 numerical dynamic programming in economics," ser. Handbook of Computational Economics. Elsevier, 1996, vol. 1, pp. 619–729.
[42] R. Munos and C. Szepesvári, "Finite-time bounds for fitted value iteration," *Journal of Machine Learning Research*, vol. 9, p. 815–857, Jun 2008.
[43] W. B. Haskell, R. Jain, H. Sharma, and P. Yu, "A universal empirical dynamic programming algorithm for continuous state mdps," *IEEE Transactions on Automatic Control*, vol. 65, no. 1, pp. 115–129, Jan 2020.
[44] B. Ankenman, B. L. Nelson, and J. Staum, "Stochastic kriging for simulation metamodeling," *Operations Research*, vol. 58, no. 2, pp. 371–382, 2010.
[45] B. Wang and J. Hu, "Some monotonicity results for stochastic kriging metamodels in sequential settings," *INFORMS Journal on Computing*, vol. 30, no. 2, pp. 278–294, 2018.
[46] M. L. Stein, *Interpolation of spatial data: some theory for kriging*, 1st ed. New York, NY: Springer, 1999.
[47] S. Petit, J. Bect, P. Feliot, and E. Vazquez, "Gaussian process interpolation: the choice of the family of models is more important than that of the selection criterion," Jul. 2021, working paper or preprint.
[48] P. Glasserman, *Monte Carlo methods in financial engineering*. Springer, 2004, vol. 53.

Xiangyu Yang holds a bachelor's degree in engineering management, minoring in financial mathematics from Shandong University, China, and a doctoral degree in management science from Fudan University, China. He is now a post-doctoral fellow with the School of Management, Shandong University. His research interests include optimization and simulation-based MDPs and financial statistics.

Jiaqiao Hu received the B.E. degree in automation from Shanghai Jiao Tong University, the M.S. degree in applied mathematics from the University of Maryland, Baltimore County, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park. Since 2006, he has been with the Department of Applied Mathematics and Statistics at the State University of New York, Stony Brook, where he is currently an Associate Professor. Dr. Hu's research interests include Markov Decision Processes, simulation optimization, and stochastic modeling and analysis. His research has been supported by the National Science Foundation, Air Force Office of Scientific Research, and the Department of Energy. Dr. Hu currently serves on the editorial boards of *IISE Transactions* and *Operations Research*.

Jian-Qiang Hu is the Distinguished Professor of Fudan University and the Hongyi Professor of Management Science in School of Management, Fudan University. He received his B.S. degree in applied mathematics from Fudan University, China, and M.S. and Ph.D. degrees in applied mathematics from Harvard University. His research interests include discrete-event stochastic systems, simulation, stochastic optimization, with applications in supply chain management, financial engineering, and healthcare. He has published over 100 research papers and is a co-author of the book, *Conditional Monte Carlo: Gradient Estimation and Optimization Applications* (Kluwer Academic Publishers, 1997). He won the Outstanding Simulation Publication Award from INFORMS Simulation Society twice (1998, 2019) and the Outstanding Research Award from Operations Research Society of China in 2020. He has been on editorial board of *Automatica*, *Operation Research*, *IIE Transaction on Design and Manufacturing*, and *Journal of the Operations Research Society of China*.