Finite-Sample Analysis of Off-Policy TD-Learning via Generalized Bellman Operators

Zaiwei Chen
Georgia Institute of Technology

Siva Theja Maguluri
Georgia Institute of Technology

Sanjay Shakkottai
The University of Texas at Austin

Karthikeyan Shanmugam
IBM Research NY

Abstract

In TD-learning, off-policy sampling is known to be more practical than on-policy sampling, and by decoupling learning from data collection, it enables data reuse. It is known that policy evaluation has the interpretation of solving a generalized Bellman equation. In this paper, we derive finite-sample bounds for any general off-policy TD-like stochastic approximation algorithm that solves for the fixed-point of this generalized Bellman operator. Our key step is to show that the generalized Bellman operator is simultaneously a contraction mapping with respect to a weighted ℓ_p-norm for each p in $[1, \infty)$, with a common contraction factor. Off-policy TD-learning is known to suffer from high variance due to the product of importance sampling ratios. A number of algorithms (e.g. $Q^\pi(\lambda)$, Tree-Backup(λ), Retrace(λ), and Q-trace) have been proposed in the literature to address this issue. Our results immediately imply finite-sample bounds of these algorithms. In particular, we provide first-known finite-sample guarantees for $Q^\pi(\lambda)$, Tree-Backup(λ), and Retrace(λ), and improve the best known bounds of Q-trace in [19]. Moreover, we show the bias-variance trade-offs in each of these algorithms.

1 Introduction

Reinforcement learning (RL) demonstrated its success in learning effective policies for a variety of decision making problems such as autonomous driving [25, 26], recommender systems [1, 41], and game-related problems [23, 27, 39]. In RL, there is an important sub-problem – called the policy evaluation problem – of estimating the expected long term reward of a given policy. Solving the policy evaluation problem is usually an itermediate step in many existing RL algorithms to ultimately find an optimal policy, such as approximate policy iteration and actor-critic framework.

The policy evaluation problem is usually solved with the TD-learning method [30]. A key ingredient in TD-learning is the policy used to collect samples (called the behavior policy). Ideally, we want to generate samples from the target policy whose value function we want to estimate, and this is called on-policy sampling. However, in many cases such on-policy sampling is not possible due to practical reasons [16, 40], and hence we need to work with historical data that is generated by a possibly different policy (i.e., off-policy sampling). Although off-policy sampling is more practical than on-policy sampling, it is more challenging to analyze and is known to have high variance [15], which is a fundamental difficulty in off-policy learning. To overcome this difficulty, many variants of off-policy TD-learning algorithms have been proposed in the literature, such as $Q^\pi(\lambda)$ [17], Tree-Backup(λ) (henceforth denoted by TB(λ)) [24], Retrace(λ) [22], and Q-trace [19], etc.

1.1 Main Contributions

In this work, we establish finite-sample bounds of a general n-step off-policy TD-learning algorithm that also subsumes several algorithms presented in the literature. The key step is to show that such

algorithm can be modeled as a Markovian stochastic approximation (SA) algorithm for solving a generalized Bellman equation. We present sufficient conditions under which the generalized Bellman operator is contractive with respect to a weighted ℓ_p-norm for every $p \in [1, \infty)$, with a uniform contraction factor for all p. Our result shows that the sample complexity scales as $\tilde{\mathcal{O}}(\epsilon^{-2})$, where ϵ is the required accuracy. It also involves a factor that depends on the problem parameters, in particular, the generalized importance sampling ratios, and explicitly demonstrates the bias-variance trade-off.

Our result immediately gives finite-sample guarantees for variants of multi-step off-policy TD-learning algorithms including $Q^\pi(\lambda)$, TB(λ), Retrace(λ), and Q-trace. For $Q^\pi(\lambda)$, TB(λ), and Retrace(λ), we establish the first-known results in the literature, while for Q-trace, we improve the best known results in [19] in terms of the dependency on the size of the state-action space. The weighted ℓ_p-norm contraction property with a uniform contraction factor for all $p \in [1, \infty)$ is crucial for us to establish the improved sample complexity. Based on the finite-sample bounds, we show that all four algorithms overcome the high variance issue in Vanilla off-policy TD-learning, but their convergence rates are all affected to varying degrees.

1.2 Generalized Bellman Operator and Stochastic Approximation

In this section, we illustrate the interpretation of off-policy multi-step TD-learning as an SA algorithm for solving a generalized Bellman equation. Consider the policy evaluation problem where the goal is to estimate the state-action value function Q^π of a given policy π. In the simplest setting where TD(0) with on-policy sampling is employed, it is well known that the algorithm is an SA algorithm for solving the Bellman equation $Q = \mathcal{H}_\pi(Q)$, where $\mathcal{H}_\pi(\cdot)$ is the Bellman operator. The generalized Bellman operator $\mathcal{B}(\cdot)$ we consider in this paper is defined by:

$$\mathcal{B}(Q) = \mathcal{T}(\mathcal{H}(Q) - Q) + Q, \tag{1}$$

where $\mathcal{T}(\cdot)$ and $\mathcal{H}(\cdot)$ are two auxiliary operators. In the special case where $\mathcal{T}(\cdot) = I(\cdot)$ and $\mathcal{H}(\cdot) = \mathcal{H}_\pi(\cdot)$, the generalized Bellman operator $\mathcal{B}(\cdot)$ reduces to the regular Bellman operator $\mathcal{H}_\pi(\cdot)$. Note that any fixed point of $\mathcal{H}(\cdot)$ is also a fixed point of $\mathcal{B}(\cdot)$, as long as $\mathcal{T}(\cdot)$ is such that $\mathcal{T}(\mathbf{0}) = \mathbf{0}$. Thus, the operator $\mathcal{H}(\cdot)$ controls the fixed-point of the generalized Bellman operator $\mathcal{B}(\cdot)$, and as we will see later, the operator $\mathcal{T}(\cdot)$ can be used to control its contraction properties.

To further understand the operator $\mathcal{B}(\cdot)$, we demonstrate in the following that both on-policy n-step TD and TD(λ) can be viewed as SA algorithms for solving the generalized Bellman equation $\mathcal{B}(Q) = Q$, with different auxiliary operators $\mathcal{T}(\cdot)$ and $\mathcal{H}(\cdot)$. On-policy n-step TD is designed to solve the n-step Bellman equation $Q = \mathcal{H}_\pi^n(Q)$, which can be explicitly written as $Q = \sum_{i=0}^{n-1} (\gamma P_\pi)^i R + (\gamma P_\pi)^n Q$. Here R is the reward vector, γ is the discount factor, and P_π is the transition probability matrix under policy π. By reverse telescoping, the n-step Bellman equation is equivalent to $Q = \sum_{i=0}^{n-1} (\gamma P_\pi)^i (R + \gamma P_\pi Q - Q) + Q = \mathcal{T}(\mathcal{H}_\pi(Q) - Q) + Q$, where $\mathcal{T}(Q) = \sum_{i=0}^{n-1} (\gamma P_\pi)^i Q$. Similarly, one can formulate the TD(λ) Bellman equation in the form of $\mathcal{B}(Q) = Q$, where $\mathcal{T}(Q) = (1-\lambda) \sum_{i=0}^{\infty} \lambda^i \sum_{j=0}^{i-1} (\gamma P_\pi)^i Q$ and $\mathcal{H}(\cdot) = \mathcal{H}_\pi(\cdot)$.

In these examples, the operator $\mathcal{T}(\cdot)$ determines the contraction factor of $\mathcal{B}(\cdot)$ by controlling the degree of bootstrapping. In this work, we show that in addition to on-policy TD-learning, variants of off-policy TD-learning with multi-step bootstrapping and generalized importance sampling ratios can also be interpreted as SA algorithms for solving the generalized Bellman equation. Moreover, under some mild conditions, we show that the generalized Bellman operator $\mathcal{B}(\cdot)$ is a contraction mapping with respect to some weighted ℓ_p-norm for any $p \in [1, \infty)$, with a common contraction factor. This enables us to establish finite-sample bounds of general multi-step off-policy TD-like algorithms.

1.3 Related Literature

The TD-learning method was first proposed in [30] for solving the policy evaluation problem. Since then, there is an increasing interest in theoretically understanding TD-learning and its variants.

On-Policy TD-Learning. The most basic TD-learning method is the TD(0) algorithm [30]. Later it was extended to using multi-step bootstrapping (i.e., the n-step TD-learning algorithm [11, 37, 38]), and using eligibility trace (i.e., the TD(λ) algorithm [28, 30]). The asymptotic convergence of TD-learning was established in [13, 18, 35]. As for finite-sample analysis, a unified Lyapunov approach is presented in [10]. To overcome the curse of dimensionality in RL, TD-learning is usually incorporated with function approximation in practice. In the basic setting where a linear parametric architecture is

used, the asymptotic convergence of TD-learning was established in [36], and finite-sample bounds in [5, 12, 29, 34]. Very recently, the convergence and finite-sample guarantee of TD-learning with neural network approximation were studied in [7, 8].

Off-Policy TD-Learning. In the off-policy setting, since the samples are not necessarily generated by the target policy, usually importance sampling ratios (or "generalized" importance sampling ratios) are introduced in the TD-learning algorithm. The resulting algorithms are $Q^\pi(\lambda)$ [24], TB(λ) [17], Retrace(λ) [22], and Q-trace [19] (which is an extension of V-trace [14]), etc. The asymptotic convergence of these algorithms has been established in the papers in which they were proposed. To the best of our knowledge, finite-sample guarantees are established only for Q-trace and V-trace [9, 10, 19]. In the function approximation setting, TD-learning with off-policy sampling and function approximation is a typical example of the deadly triad [31], and can be unstable [2, 31]. To achieve convergence, one needs to significantly modify the original TD-learning algorithm, resulting in two time-scale algorithms such as GTD [21], TDC [32], and emphatic TD [33], etc.

1.4 Preliminaries

In this section, we cover the background of RL and the TD-learning method for solving the policy evaluation problem. The RL problem is usually modeled as a Markov decision process (MDP). In this work, we consider an MDP with a finite set of states $\mathcal{S}$, a finite set of actions $\mathcal{A}$, a set of unknown action dependent transition probability matrices $\mathcal{P} = \{P_a \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|} \mid a \in \mathcal{A}\}$, an unknown reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$, and a discount factor $\gamma \in (0, 1)$. In order for an MDP to progress, we must specify the policy of selecting actions based on the state of the environment. Specifically, a policy π is a mapping from the state-space to probability distributions supported on the action space, i.e., $\pi : \mathcal{S} \mapsto \Delta^{|\mathcal{A}|}$. The state-action value function Q^π associated with a policy π is defined by $Q^\pi(s, a) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k \mathcal{R}(S_k, A_k) \mid S_0 = s, A_0 = a]$ for all (s, a). The goal in policy evaluation is to estimate the state-action value function Q^π for a given policy π.

Since the transition probabilities as well as the reward function are unknown, such state-action value function cannot be directly computed. The TD-learning algorithm is designed to estimate Q^π using the SA method. Specifically, in TD-learning, we first collect a sequence of samples $\{(S_k, A_k)\}$ from the model using some behavior policy π_b. Then the value function Q^π is iteratively estimated using the samples $\{(S_k, A_k)\}$. When $\pi_b = \pi$, the algorithm is called on-policy TD-learning, otherwise the algorithm is referred to as off-policy TD-learning.

2 Finite-Sample Analysis of General Off-Policy TD-Learning

In this section, we present our unified framework for finite-sample analysis of off-policy TD-learning algorithms using generalized importance sampling ratios and multi-step bootstrapping. The proofs of all technical results presented in this paper are provided in the Appendix.

2.1 A Generic Model for Multi-Step Off-Policy TD-Learning

Algorithm 1 presents our generic algorithm model. Due to off-policy sampling, the two functions $c, \rho : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}_+$ are introduced in Algorithm 1 to serve as generalized importance sampling ratios in order to account for the discrepancy between the target policy π and the behavior policy π_b. We denote $c_{\max} = \max_{s,a} c(s, a)$ and $\rho_{\max} = \max_{s,a} \rho(s, a)$. We next show how Algorithm 1 captures variants of off-policy TD-learning algorithms in the literature by using different generalized importance sampling ratios $c(\cdot, \cdot)$ and $\rho(\cdot, \cdot)$.

Vanilla IS. When $c(s, a) = \rho(s, a) = \pi(a|s)/\pi_b(a|s)$ for all (s, a), Algorithm 1 is the standard off-policy TD-learning with importance sampling [24]. We will refer to this algorithm as Vanilla IS. Although Vanilla IS was shown to converge to Q^π [24], since the product of importance sampling ratios $\prod_{j=k+1}^{i} \frac{\pi(A_j|S_j)}{\pi_b(A_j|S_j)}$ is not controlled in any way, it suffers the most from high variance.

The $Q^\pi(\lambda)$ Algorithm. When $c(s, a) = \lambda$ and $\rho(s, a) = \pi(a|s)/\pi_b(a|s)$, Algorithm 1 is the $Q^\pi(\lambda)$ algorithm [17]. The $Q^\pi(\lambda)$ algorithm overcomes the high variance issue in Vanilla IS by introducing the parameter λ. However, the algorithm converges only when λ is sufficiently small [22].

The TB(λ) Algorithm. When $c(s, a) = \lambda\pi(a|s)$ and $\rho(s, a) = \pi(a|s)/\pi_b(a|s)$, we have the TB($\lambda$) algorithm [24]. The TB(λ) algorithm also overcomes the high variance issue in Vanilla IS and is

---

Algorithm 1 A Generic Algorithm for Multi-Step Off-Policy TD-Learning

---
1: **Input:** K, $\{\alpha_k\}$, Q_0, π, π_b, generalized importance sampling ratios $c, \rho : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}_+$, and sample trajectory $\{(S_k, A_k)\}_{0 \leq k \leq K+n}$ collected under the behavior policy π_b.
2: **for** $k = 0, 1, \cdots, K-1$ **do**
3:    $\alpha_k(s, a) = \alpha_k \mathbb{I}\{(s, a) = (S_k, A_k)\}$ for all (s, a)
4:    $\Delta(S_i, A_i, S_{i+1}, A_{i+1}, Q_k) = \mathcal{R}(S_i, A_i) + \gamma\rho(S_{i+1}, A_{i+1})Q_k(S_{i+1}, A_{i+1}) - Q_k(S_i, A_i)$ for all $i \in \{k, k+1, ..., k+n-1\}$.
5:    $Q_{k+1}(s, a) = Q_k(s, a) + \alpha_k(s, a)\sum_{i=k}^{k+n-1}\gamma^{i-k}\prod_{j=k+1}^{i}c(S_j, A_j)\Delta(S_i, A_i, S_{i+1}, A_{i+1}, Q_k)$ for all (s, a)
6: **end for**
7: **Output:** Q_K

---

guaranteed to converge to Q^π without needing any strong assumptions. However, as discussed in [22], the TB(λ) algorithm lacks sample efficiency as it does not effectively use the multi-step return.

The Retrace(λ) Algorithm. When $c(s, a) = \lambda\min(1, \pi(a|s)/\pi_b(a|s))$ and $\rho(s, a) = \pi(a|s)/\pi_b(a|s)$, we have the Retrace($\lambda$) algorithm, which overcomes the high variance and converges to Q^π. The convergence rate of Retrace(λ) is empirically observed to be better than TB(λ) in [22].

The Q-trace Algorithm. When we choose $c(s, a) = \min(\bar{c}, \pi(a|s)/\pi_b(a|s))$ and $\rho(s, a) = \min(\bar{\rho}, \pi(a|s)/\pi_b(a|s))$, where $\bar{\rho} \geq \bar{c}$, Algorithm 1 is the Q-trace algorithm [19]. The Q-trace algorithm is an analog of the V-trace algorithm [14] in that Q-trace estimates the Q-function instead of the V-function. The two truncation levels $\bar{c}$ and $\bar{\rho}$ in these algorithms separately control the variance and the asymptotic bias in the algorithm respectively. Note that due to the truncation level $\bar{\rho}$, the algorithm no longer converges to Q^π, but to a biased limit point, denoted by $Q^{\pi,\rho}$ [19].

From now on, we focus on studying Algorithm 1. We make the following assumption about the behavior policy π_b, which is fairly standard in off-policy TD-learning.

Assumption 2.1. The behavior policy π_b satisfies $\pi_b(a|s) > 0$ for all (s, a). In addition, the Markov chain $\{S_k\}$ induced by the behavior policy π_b is irreducible and aperiodic.

Irreducibility and aperiodicity together imply that the Markov chain $\{S_k\}$ has a unique stationary distribution, which we denote by $\kappa_S \in \Delta^{|\mathcal{S}|}$. Moreover, the Markov chain $\{S_k\}$ mixes geometrically fast in that there exist $C > 0$ and $\sigma \in (0, 1)$ such that $\max_{s \in \mathcal{S}} \|P^k(s, \cdot) - \kappa_S(\cdot)\|_{\text{TV}} \leq C\sigma^k$ for all $k \geq 0$, where $\|\cdot\|_{\text{TV}}$ is the total variation distance [20]. Let $\kappa_{SA} \in \Delta^{|\mathcal{S}||\mathcal{A}|}$ be such that $\kappa_{SA}(s, a) = \kappa_S(s)\pi_b(a|s)$ for all (s, a). Note that $\kappa_{SA} \in \Delta^{|\mathcal{S}||\mathcal{A}|}$ is the stationary distribution of the Markov chain $\{(S_k, A_k)\}$ under the behavior policy π_b. Let $\mathcal{K}_S = \text{diag}(\kappa_S) \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$, and let $\mathcal{K}_{SA} = \text{diag}(\kappa_{SA}) \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}||\mathcal{A}|}$. Denote the minimal (maximal) diagonal entries of $\mathcal{K}_S$ and $\mathcal{K}_{SA}$ by $\mathcal{K}_{S,\min}$ ($\mathcal{K}_{S,\max}$) and $\mathcal{K}_{SA,\min}$ ($\mathcal{K}_{S,\max}$) respectively.

2.2 Identifying the Generalized Bellman Operator

In this section, we identify the generalized Bellman equation which Algorithm 1 is trying to solve, and also the corresponding generalized Bellman operator and its asynchronous variant. Let $\mathcal{T}_c, \mathcal{H}_\rho : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \mapsto \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ be two operators defined by

$$[\mathcal{T}_c(Q)](s, a) = \sum_{i=0}^{n-1}\gamma^i\mathbb{E}_{\pi_b}[\prod_{j=1}^{i}c(S_j, A_j)Q(S_i, A_i) \mid S_0 = s, A_0 = a], \quad \text{and}$$

$$[\mathcal{H}_\rho(Q)](s, a) = \mathcal{R}(s, a) + \gamma\mathbb{E}_{\pi_b}[\rho(S_{k+1}, A_{k+1})Q(S_{k+1}, A_{k+1}) \mid S_k = s, A_k = a]$$

for all (s, a). Note that the operator $\mathcal{T}_c(\cdot)$ depends on the generalized importance sampling ratio $c(\cdot, \cdot)$, while the operator $\mathcal{H}_\rho(\cdot)$ depends on the generalized importance sampling ratio $\rho(\cdot, \cdot)$.

With $\mathcal{T}_c(\cdot)$ and $\mathcal{H}_\rho(\cdot)$ defined above, Algorithm 1 can be viewed as an asynchronous SA algorithm for solving the generalized Bellman equation $\mathcal{B}_{c,\rho}(Q) = Q$, where the generalized Bellman operator $\mathcal{B}_{c,\rho}(\cdot)$ is defined by $\mathcal{B}_{c,\rho}(Q) = \mathcal{T}_c(\mathcal{H}_\rho(Q) - Q) + Q$. Since Algorithm 1 performs asynchronous update, using the terminology in [10], we further define the asynchronous variant $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ of the generalized Bellman operator $\mathcal{B}_{c,\rho}(\cdot)$ by

$$\tilde{\mathcal{B}}_{c,\rho}(Q) := \mathcal{K}_{SA}\mathcal{B}_{c,\rho}(Q) + (I - \mathcal{K}_{SA})Q = \mathcal{K}_{SA}\mathcal{T}_c(\mathcal{H}_\rho(Q) - Q) + Q. \tag{2}$$

Each component of the asynchronous generalized Bellman operator $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ can be thought of as a convex combination with identity, where the weights are the stationary probabilities of visiting state-action pairs. This captures the fact that when performing asynchronous update, the corresponding component is updated only when the state-action pair is visited. It is clear from its definition that $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ has the same fixed-points as $\mathcal{B}_{c,\rho}(\cdot)$ (provided that they exist). See [10] for a more detailed explanation about asynchronous Bellman operators.

Under some mild conditions on the generalized importance sampling ratios $c(\cdot, \cdot)$ and $\rho(\cdot, \cdot)$, we will show in the next section that both the asynchronous generalized Bellman operator $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ and the operator $\mathcal{H}_\rho(\cdot)$ are contraction mappings. Therefore, since $\mathcal{T}_c(\mathbf{0}) = \mathbf{0}$, the operators $\mathcal{H}_\rho(\cdot)$, $\mathcal{B}_{c,\rho}(\cdot)$, $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ all share the same unique fixed-point. Since the fixed-point of the operator $\mathcal{H}_\rho(\cdot)$ depends only on the generalized importance sampling ratio $\rho(\cdot, \cdot)$, but not on $c(\cdot, \cdot)$, we can flexibly choose $c(\cdot, \cdot)$ to control the variance while maintaining the fixed-point of the operator $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$. As we will see later, this is the key property used in designing variants of variance reduced n-step off-policy TD-learning algorithms such as $Q^\pi(\lambda)$, TB(λ), and Retrace(λ).

2.3 Establishing the Contraction Property

In this section, we study the fixed-point and the contraction property of the asynchronous generalized Bellman operator $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$. We begin by introducing some notation. Let $D_c, D_\rho \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}||\mathcal{A}|}$ be two diagonal matrices such that $D_c((s,a),(s,a)) = \sum_{a' \in \mathcal{A}} \pi_b(a'|s) c(s,a')$ and $D_\rho((s,a),(s,a)) = \sum_{a' \in \mathcal{A}} \pi_b(a'|s) \rho(s,a')$ for all (s,a). We denote $D_{c,\min}$ ($D_{c,\max}$) and $D_{\rho,\min}$ ($D_{\rho,\max}$) as the minimal (maximal) diagonal entries of the matrices D_c and D_ρ respectively.

In view of the definition of $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ in Eq. (2), any fixed-point of $\mathcal{H}_\rho(\cdot)$ must also be a fixed-point of $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$. We first study the fixed point of $\mathcal{H}_\rho(\cdot)$ by establishing its contraction property.

Proposition 2.1. *Suppose that $D_{\rho,\max} < 1/\gamma$. Then the operator $\mathcal{H}_\rho(\cdot)$ is a contraction mapping with respect to the ℓ_∞-norm, with contraction factor $\gamma D_{\rho,\max}$. In this case, the fixed-point $Q^{\pi,\rho}$ of $\mathcal{H}_\rho(\cdot)$ satisfies the following two inequalities: (1) $\|Q^\pi - Q^{\pi,\rho}\|_\infty \leq \frac{\gamma \max_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} |\pi(a|s) - \pi_b(a|s)\rho(s,a)|}{(1-\gamma)(1-\gamma D_{\rho,\max})}$, and (2) $\|Q^{\pi,\rho}\|_\infty \leq \frac{1}{1-\gamma D_{\rho,\max}}$.*

Observe from Proposition 2.1 (1) that when $\rho(s,a) = \pi(a|s)/\pi_b(a|s)$, which is the case for $Q^\pi(\lambda)$, TB(λ), and Retrace(λ), the unique fixed-point $Q^{\pi,\rho}$ is exactly the target value function Q^π. This agrees with the definition of the operator $\mathcal{H}_\rho(\cdot)$ in that it reduces to the regular Bellman operator $\mathcal{H}_\pi(\cdot)$ when $\rho(s,a) = \pi(a|s)/\pi_b(a|s)$ for all (s,a). If $\rho(s,a) \neq \pi(a|s)/\pi_b(a|s)$ for some (s,a), then in general the fixed-point of $\mathcal{H}_\rho(\cdot)$ is different from Q^π. See Appendix A.2 for more details. In that case, Proposition 2.1 provides an error bound on the difference between the potentially biased limit $Q^{\pi,\rho}$ and Q^π. Such error bound will be useful for us to study the Q-trace algorithm in Section 3. Proposition 2.1 (2) can be viewed as an analog to the inequality that $\|Q^\pi\|_\infty \leq 1/(1-\gamma)$ for any policy π. Since $\mathcal{H}_\rho(\cdot)$ is no longer the Bellman operator $\mathcal{H}_\pi(\cdot)$, the corresponding upper bound on the size of its fixed-point $Q^{\pi,\rho}$ also changes.

Note that Proposition 2.1 guarantees the existence and uniqueness of the fixed-point of the operator $\mathcal{H}_\rho(\cdot)$, hence also ensures the existence of fixed-points of the asynchronous generalized Bellman operator $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$. To further guarantee the uniqueness of the fixed-point of $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$, we establish its contraction property. We begin with the following definition.

Definition 2.1. *Let $\{\mu_i\}_{1 \leq i \leq d}$ be such that $\mu_i > 0$ for all i. Then for any $x \in \mathbb{R}^d$, the weighted ℓ_p-norm ($p \in [1, \infty)$) of x with weights $\{\mu_i\}$ is defined by $\|x\|_{\mu,p} = (\sum_i \mu_i |x_i|^p)^{1/p}$.*

We next establish the contraction property of the operator $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ in the following theorem. Let $\omega = \mathcal{K}_{SA,\min} f(\gamma D_{c,\min})(1 - \gamma D_{\rho,\max})$, where the function $f : \mathbb{R} \mapsto \mathbb{R}$ is defined by $f(x) = n$ when $x = 1$, and $f(x) = \frac{1-x^n}{1-x}$ when $x \neq 1$.

Theorem 2.1. *Suppose $c(s,a) \leq \rho(s,a)$ for all (s,a) and $D_{\rho,\max} < 1/\gamma$. Then we have the following results: (1) For any $\theta \in (0,1)$, there exists a weight vector $\mu \in \Delta^{|\mathcal{S}||\mathcal{A}|}$ satisfying $\mu(s,a) \geq \frac{\omega(1-\theta)}{(1-\theta\omega)|\mathcal{S}||\mathcal{A}|}$ for all (s,a) such that the operator $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ is a contraction mapping with respect to $\|\cdot\|_{\mu,p}$ for any $p \in [1, \infty)$, with contraction factor $\gamma_c = (1-\omega)^{1-1/p}(1-\theta\omega)^{1/p}$, (2) The operator $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ is a contraction mapping with respect to $\|\cdot\|_\infty$, with contraction factor $\gamma_c = 1 - \omega$.*

Consider Theorem 2.1 (1). Observe that we can further upper bound $\gamma_c = (1-\omega)^{1-1/p}(1-\theta\omega)^{1/p}$ by $1-\theta\omega$, which is independent of p and is the uniform contraction factor we are going to use. Theorem 2.1 (2) can be viewed as an extension of Theorem 2.1 because $\lim_{p\to\infty}\|x\|_{\mu,p} = \|x\|_\infty$ for any $x \in \mathbb{R}^d$ and weight vector μ, and $\lim_{p\to\infty}(1-\omega)^{1-1/p}(1-\theta\omega)^{1/p} = 1-\omega$.

Theorem 2.1 is the key result for our finite-sample analysis, and we present its proof in the next section. The weighted ℓ_p-norm (especially the weighted ℓ_2-norm) contraction property we established for the operator $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ has a far-reaching impact even beyond the finite-sample analysis of tabular RL in this paper. Specifically, recall that the key property used for establishing the convergence and finite-sample bound of on-policy TD-learning with *linear function approximation* in the seminal work [36] is that the corresponding Bellman operator is a contraction mapping not only with respect to the ℓ_∞-norm, but also with respect to a weighted ℓ_2-norm. We establish the same property in the off-policy setting, and hence lay down the foundation for extending our results to the function approximation setting. This is an immediate future research direction.

2.4 Proof of Theorem 2.1

We begin by explicitly computing the asynchronous generalized Bellman operator $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$. Let π_c and π_ρ be two policies defined by $\pi_c(a|s) = \frac{\pi_b(a|s)c(s,a)}{D_c((s,a),(s,a))}$ and $\pi_\rho(a|s) = \frac{\pi_b(a|s)\rho(s,a)}{D_\rho((s,a),(s,a))}$ for all (s,a). Let $R \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ be the reward vector defined by $R(s,a) = \mathcal{R}(s,a)$ for all (s,a). For any policy π', let $P_{\pi'}$ be the transition probability matrix of the Markov chain $\{(S_k, A_k)\}$ under π', i.e., $P_{\pi'}((s,a),(s',a')) = P_a(s,s')\pi'(a'|s')$ for all state-action pairs (s,a) and (s',a').

Proposition 2.2. *The operator $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ is explicitly given by $\tilde{\mathcal{B}}_{c,\rho}(Q) = AQ + b$, where $A = I - \mathcal{K}_{SA}\sum_{i=0}^{n-1}(\gamma P_{\pi_c}D_c)^i(I - \gamma P_{\pi_\rho}D_\rho)$ and $b = \mathcal{K}_{SA}\sum_{i=0}^{n-1}(\gamma P_{\pi_c}D_c)^i R$.*

In light of Proposition 2.2, to prove Theorem 2.1, it is enough to study the matrix A. To proceed, we require the following definition.

Definition 2.2. *Given $\beta \in [0,1]$, a matrix $M \in \mathbb{R}^{d\times d}$ is called a substochastic matrix with modulus β if and only if $M_{ij} \geq 0$ for all i,j and $\sum_j M_{ij} \leq 1-\beta$ for all i.*

Remark. Note that for any non-negative matrix M, we have $\|M\|_\infty = \max_i \sum_j M_{ij}$. Therefore, a matrix M being a substochastic matrix with modulus β automatically implies that $\|M\|_\infty \leq 1-\beta$.

We next show in the following two propositions that (1) the matrix A given in Proposition 2.2 is a substochastic matrix with modulus ω, and (2) for any substochastic matrix M with a positive modulus, there exist weights $\{\mu_i\}$ such that the induced matrix norm $\|M\|_{\mu,p}$ is strictly less than 1. These two results together immediately imply Theorem 2.1.

Proposition 2.3. *Suppose that $c(s,a) \leq \rho(s,a)$ for all (s,a) and $D_{\rho,\max} < 1/\gamma$. Then the matrix A given in Proposition 2.2 is a substochastic matrix with modulus ω.*

The condition $c(s,a) \leq \rho(s,a)$ ensures that the matrix A is non-negative, and the condition $D_{\rho,\max} < 1/\gamma$ ensures that the each row of the matrix A sums up to at most $1-\omega$. Together they imply the substochasticity of A. The modulus ω is an important parameter for our finite-sample analysis. In view of Theorem 2.1, we see that large modulus gives smaller (or better) contraction factor of $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$.

Proposition 2.4. *For any substochastic matrix $M \in \mathbb{R}^{d\times d}$ with a positive modulus $\beta \in (0,1)$, for any $\theta \in (0,1)$, there exists a weight vector $\mu \in \Delta^d$ satisfying $\mu_i \geq \frac{\beta(1-\theta)}{(1-\theta\beta)d}$ for all i such that $\|M\|_{\mu,p} \leq (1-\beta)^{1-1/p}(1-\theta\beta)^{1/p}$ for any $p \in [1,\infty)$. Furthermore, if M is irreducible [1], then we can choose $\theta = 1$.*

The result of Proposition 2.4 further implies $\|M\|_{\mu,p} \leq 1-\theta\beta$, which is independent of the choice of p. This implies that $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ is a uniform contraction mapping with respect to $\|\cdot\|_{\mu,p}$ for all $p \geq 1$. In general, for different p and p', an operator being a $\|\cdot\|_p$-norm contraction does not imply being a $\|\cdot\|_{p'}$-norm contraction. The reason that we have such a strong uniform contractive result is that the operator $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ has a linear structure, and involves a substochastic matrix.

Note that Proposition 2.4 introduces the tunable parameter θ. It is clear that large θ gives better contraction factor of $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ but worse lower bound on the entries of the weight vector μ. In

---

[1] A non-negative matrix is irreducible if and only if its associated graph is strongly connected [4].

general, when M is not irreducible, we cannot hope to choose a weight vector $\mu \in \Delta^d$ with positive components and obtain $\|M\|_{\mu,p} \leq 1 - \omega$. To see this, consider the example where $M = (1 - \omega)[\mathbf{0}, \mathbf{0}, \cdots, \mathbf{1}]$, which is clearly a substochastic matrix with modulus ω, but is not an irreducible matrix. For any weight vector $\mu \in \Delta^d$, we have $\|M\|_{\mu,p} = (1 - \omega) \max_{x \in \mathbb{R}^d : \|x\|_{\mu,p} = 1} |x_d| = (1 - \omega)/\mu_d^{1/p} > 1 - \omega$. However, by choosing μ_d close to unity, we can get $\|M\|_{\mu,p}$ arbitrarily close to $1 - \omega$. This is analogous to choosing θ close to one in Proposition 2.4. Since Proposition 2.4 is the major result for proving Theorem 2.1, we provide its proof sketch in Section 4.

2.5 Finite-Sample Convergence Guarantees

In light of Theorem 2.1, Algorithm 1 is a Markovian SA algorithm for solving a fixed-point equation $\tilde{\mathcal{B}}_{c,\rho}(Q) = Q$, where the fixed-point operator $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ is a contraction mapping. Therefore, to establish the finite-sample bounds, we use a Lyapunov drift argument where we choose $W(Q) = \|Q - Q^{\pi,\rho}\|_{\mu,p}^2$ as the Lyapunov function. This leads to a finite-sample bound on $\mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_{\mu,p}^2]$. However, since μ is unknown, to make the finite-sample bound independent of μ, we use the lower bound on the components of μ provided in Theorem 2.1, and also tune the parameters p and θ to obtain a finite-sample bound on $\mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_{\infty}^2]$. The fact that we have a uniform contraction factor $1 - \theta\omega$ (cf. Theorem 2.1) plays an important role in such tuning process.

To present the results, we need to introduce more notation. For any $\delta > 0$, define $t_\delta(\mathcal{MC}_S)$ as the mixing time of the Markov chain $\{S_k\}$ (induced by π_b) with precision δ, i.e., $t_\delta(\mathcal{MC}_S) = \min\{k \geq 0 : \max_{s \in \mathcal{S}} \|P^k(s, \cdot) - \kappa_S(\cdot)\|_{\text{TV}} \leq \delta\}$. Under Assumption 2.1, one can easily verify that $t_\delta(\mathcal{MC}_S) \leq L(\log(1/\delta) + 1)$ for some constant $L > 0$, which depends only on C and δ. Let $\tau_{\delta,n} = t_\delta(\mathcal{MC}_S) + n + 1$. The parameters c_1, c_2 and c_3 used in stating the following theorem are numerical constants, and will be explicitly given in the Appendix. For ease of exposition, we here only present the finite-sample bound for using constant stepsize.

Theorem 2.2. *Consider $\{Q_k\}$ of Algorithm 1. Suppose that: (1) Assumptions 2.1 is satisfied, (2) $c(s,a) \leq \rho(s,a)$ for all (s,a) and $D_{\rho,\max} < 1/\gamma$, and (3) the constant stepsize α is chosen such that $\alpha\tau_{\alpha,n} \leq \frac{c_1\omega}{\log(2|\mathcal{S}||\mathcal{A}|/\omega)f(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2}$. Then we have for all $k \geq \tau_{\alpha,n}$:*

$$\mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_{\infty}^2] \leq \zeta_1 \left(1 - \frac{\omega\alpha}{2}\right)^{k-\tau_{\alpha,n}} + \zeta_2 \frac{f(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2 \log(2|\mathcal{S}||\mathcal{A}|/\omega)}{\omega}\alpha\tau_{\alpha,n}, \quad (3)$$

where $\zeta_1 = c_2(\|Q_0 - Q^{\pi,\rho}\|_\infty + \|Q_0\|_\infty + 1)^2$, and $\zeta_2 = c_3(3\|Q^{\pi,\rho}\|_\infty + 1)^2$.

Theorem 2.2 enables one to design a wide class of off-policy TD variants with provable finite-sample guarantees by choosing appropriate generalized importance sampling ratios $c(\cdot,\cdot)$ and $\rho(\cdot,\cdot)$. The first term on the RHS of Eq. (3) is usually called the bias in SA literature [6], and it goes to zero at a geometric rate. The second term on the RHS of Eq. (3) stands for the variance in the iterates, and it is a constant proportional to $\alpha\tau_{\alpha,n}$. To see more explicitly the bias-variance trade-off, we derive the sample complexity of Algorithm 1 in the following.

Corollary 2.2.1. *For an accuracy $\epsilon > 0$, to obtain $\mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_\infty] \leq \epsilon$, the sample complexity is*

$$\underbrace{\mathcal{O}\left(\frac{\log^2(1/\epsilon)}{\epsilon^2}\right)}_{T_1} \underbrace{\tilde{\mathcal{O}}\left(\frac{1}{\omega^2}\right)}_{T_2} \underbrace{\tilde{\mathcal{O}}\left(\frac{nf(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2}{(1-\gamma D_{\rho,\max})^2}\right)}_{T_3}. \quad (4)$$

In Corollary 2.2.1, the $\tilde{\mathcal{O}}(\epsilon^{-2})$ dependence on the accuracy is the same as n-step TD-learning in the on-policy setting [10], and is in general not improvable. The term T_2 can be equivalently written as $\tilde{\mathcal{O}}(1/(1 - \text{Contraction factor})^2)$, hence capturing the impact from the contraction factor. This agrees with our intuition that smaller contraction factor leads to better sample complexity. The term T_3 arises because of the variance term on the RHS of Eq. (3). The linear dependence on n is due to using n-step bootstrapping. By optimizing the sample complexity in terms of n, we have $n_{\text{optimal}} \sim 1/\log(1/(\gamma D_{c,\min}))$. This is analogous to the optimal n in the on-policy setting, which is $1/\log(1/\gamma)$ [10]. The additional $D_{c,\min}$ factor arises because of using off-policy learning. The rest of parameters in T_3 are determined by the choice of the generalized importance sampling ratios $c(\cdot,\cdot)$ and $\rho(\cdot,\cdot)$. It is clear that smaller $c_{\max}$ and $\rho_{\max}$ lead to smaller variance. As we will see later, this is the reason for the variance reduction of various off-policy TD-learning algorithms in the literature. In light of the previous analysis, the bias-variance trade-off in general off-policy multi-step TD-learning algorithm 1 is intuitively of the form $\tilde{\mathcal{O}}\left(\frac{\text{Variance}}{(1-\text{Contraction factor})^2}\right)$.

3 Application to Various Off-Policy TD-Learning Algorithms

In this section, we apply Theorem 2.2 to various off-policy n-step TD-learning algorithms in the literature. We begin by introducing some notation. Let $\pi_{\max}$ ($\pi_{\min}$) and $\pi_{b,\max}$ ($\pi_{b,\min}$) be the maximal (minimal) entry of the target policy π and the behavior policy π_b respectively. Let $r_{\max} = \max_{s,a}(\pi(a|s)/\pi_b(a|s))$ ($r_{\min} = \min_{s,a}(\pi(a|s)/\pi_b(a|s))$) be the maximum (minimum) ratio between π and π_b. We will overload the notation of ζ_1 and ζ_2 from Theorem 2.2. Note that $Q^{\pi,\rho} = Q^\pi$ in $Q^\pi(\lambda)$, TB(λ), and Retrace(λ), but $Q^{\pi,\rho} \neq Q^\pi$ in Q-trace.

3.1 Finite-Sample Analysis of Vanilla IS

We first present the sample complexity bound of the Vanilla IS algorithm, where $c(s,a) = \rho(s,a) = \pi(a|s)/\pi_b(a|s)$ for all (s,a).

Theorem 3.1. *Consider Algorithm 1 with Vanilla IS update, where we note that $c_{\max} = \rho_{\max} = r_{\max}$, $D_c = D_\rho = I$, and $\omega = \mathcal{K}_{SA,\min}(1-\gamma^n)$. Suppose that Assumption 2.1 is satisfied. Then, to achieve ϵ-accuracy, the sample complexity is $O\left(\frac{\log^2(1/\epsilon)}{\epsilon^2}\right)\tilde{O}\left(\frac{1}{\omega^2}\right)\tilde{O}\left(\frac{n((\gamma r_{\max})^n+1)^2}{(1-\gamma)^2}\right)$.*

In the special case where $\pi = \pi_b$ (i.e., on-policy n-step TD), the sample complexity bound reduces to $\tilde{O}\left(\frac{n\log^2(1/\epsilon)}{\epsilon^2 \mathcal{K}_{SA,\min}^2(1-\gamma^n)^2(1-\gamma)^2}\right)$, which is comparable to the results in [10]. See Appendix C.2 for more details. In the off-policy setting, note that the factor $((\gamma r_{\max})^n + 1)^2$ appears in the sample complexity. When $\gamma r_{\max} > 1$ (which can usually happen), the sample complexity bound involves an exponential factor $(\gamma r_{\max})^n$. The reason is that the product of importance sampling ratios $c(\cdot, \cdot)$ are not at all controlled by any means in Vanilla IS. Therefore, the variance can be very large. On the other hand, since the importance sampling ratios are not modified, Vanilla IS effectively uses the full n-step return. As a result, the parameter $\omega = \mathcal{K}_{SA,\min}(1-\gamma^n)$ within Vanilla IS is the largest (best) among all the algorithms we study.

3.1.1 Finite-Sample Analysis of $Q^\pi(\lambda)$

In this section, we present the sample complexity of the $Q^\pi(\lambda)$ algorithm, where $c(s,a) = \lambda$ and $\rho(s,a) = \pi(a|s)/\pi_b(a|s)$ for all (s,a).

Theorem 3.2. *Consider Algorithm 1 with $Q^\pi(\lambda)$ update, where we note that $c_{\max} = \lambda$, $\rho_{\max} = r_{\max}$, $D_c = \lambda I$, $D_\rho = I$, and $w = \mathcal{K}_{SA,\min}f(\gamma\lambda)(1-\gamma)$. Suppose that Assumption 2.1 is satisfied, and $\lambda \leq r_{\min}$. Then, to achieve ϵ-accuracy, the sample complexity is $O\left(\frac{\log^2(1/\epsilon)}{\epsilon^2}\right)\tilde{O}\left(\frac{1}{\omega^2}\right)\tilde{O}\left(\frac{nf(\gamma\lambda)^2(\gamma r_{\max}+1)^2}{(1-\gamma)^2}\right)$.*

To see how $Q^\pi(\lambda)$ overcomes the high variance issue in Vanilla IS, observe that since $\gamma\lambda \leq \gamma r_{\min} \leq \gamma < 1$, we have $f^2(\gamma\lambda) \leq 1/(1-\gamma\lambda)^2$. Therefore, by replacing $c(s,a) = \pi(a|s)/\pi_b(a|s)$ in Vanilla IS with a properly chosen constant λ, $Q^\pi(\lambda)$ algorithm successfully avoids an exponential large factor in the sample complexity. However, choosing a small λ to control the variance has a side effect on the contraction factor. Intuitively, when λ is small, $Q^\pi(\lambda)$ does not effectively use the n-step return. Hence the parameter ω in $Q^\pi(\lambda)$ is less (worse) than the one in Vanilla IS.

3.1.2 Finite-Sample Analysis of TB(λ)

In this section, we present the sample complexity of the TB(λ) algorithm, where $c(s,a) = \lambda\pi(a|s)$ and $\rho(s,a) = \pi(a|s)/\pi_b(a|s)$ for all (s,a).

Theorem 3.3. *Consider Algorithm 1 with TB(λ) update, where we note that $c_{\max} = \lambda\pi_{\max}$, $\rho_{\max} = r_{\max}$, $D_c(s,a) = \lambda\sum_a \pi_b(a|s)\pi(a|s)$, $D_\rho(s,a) = 1$, and $\omega = \mathcal{K}_{SA,\min}f(\gamma D_{c,\min})(1-\gamma)$. Suppose that Assumption 2.1 is satisfied, and $\lambda \leq 1/\pi_{b,\max}$. Then, to achieve ϵ-accuracy, the sample complexity is $O\left(\frac{\log^2(1/\epsilon)}{\epsilon^2}\right)\tilde{O}\left(\frac{1}{\omega^2}\right)\tilde{O}\left(\frac{nf(\gamma\lambda\pi_{\max})^2(\gamma r_{\max}+1)^2}{(1-\gamma)^2}\right)$.*

Suppose we further choose $\lambda < 1/(\gamma\pi_{\max})$, the TB($\lambda$) algorithm also overcomes the high variance issue in Vanilla IS because $f(\gamma\lambda\pi_{\max}) \leq 1/(1-\gamma\lambda\pi_{\max})$, which does not involve any exponential large factor. When compared to $Q^\pi(\lambda)$, an advantage of TB(λ) is that the constraint on λ is much relaxed. However, the same side effect on the contraction factor is also present here. To see this, since $D_{c,\min} = \lambda\min_{s,a}\sum_a \pi_b(a|s)\pi(a|s) \leq 1$, the TB($\lambda$) algorithm does not effectively use the n-step return, hence the parameter ω in TB(λ) is less (worse) than the one in Vanilla IS.

3.1.3 Finite-Sample Analysis of Retrace(λ)

In this section, we present the sample complexity of the Retrace(λ) algorithm, where $c(s, a) = \lambda \min(1, \pi(a|s)/\pi_b(a|s))$ and $\rho(s, a) = \pi(a|s)/\pi_b(a|s)$ for all (s, a).

Theorem 3.4. *Consider Algorithm 1 with Retrace(λ) update, where we note that $c_{\max} = \lambda$, $\rho_{\max} = r_{\max}$, $D_c(s, a) = \lambda \sum_a \min(\pi_b(a|s), \pi(a|s))$, $D_\rho(s, a) = 1$, and $\omega = \mathcal{K}_{SA,\min} f(\gamma D_{c,\min})(1 - \gamma)$. Suppose that Assumption 2.1 is satisfied, and $\lambda \leq 1$. Then, to achieve ϵ-accuracy, the sample complexity is $O\left(\frac{\log^2(1/\epsilon)}{\epsilon^2}\right) \tilde{\mathcal{O}}\left(\frac{1}{\omega^2}\right) \tilde{\mathcal{O}}\left(\frac{nf(\gamma\lambda)^2(\gamma r_{\max}+1)^2}{(1-\gamma)^2}\right)$.*

The Retrace(λ) algorithm overcomes the high variance issue in Vanilla IS by truncating the importance sampling ratio at 1, which prevents an exponential large factor in the variance term. In addition, it does not require choosing λ to be extremely small as required in $Q^\pi(\lambda)$. As for the compromise in the contraction factor, note that $\min(1, \pi(a|s)/\pi_b(a|s)) \geq \pi(a|s)$, which implies that $D_c(s, a)$ (and hence $D_{c,\min}$) is larger in the Retrace(λ) algorithm than the TB(λ) algorithm. As a result, Retrace(λ) does not truncate the n-step return as heavy as TB(λ), and hence the parameter ω is larger (better) in Retrace(λ) than in TB(λ).

3.1.4 Finite-Sample Analysis of Q-Trace

Lastly, we present the sample complexity of the Q-trace algorithm, where $c(s, a) = \min(\bar{c}, \pi(a|s)/\pi_b(a|s))$ and $\rho(s, a) = \min(\bar{\rho}, \pi(a|s)/\pi_b(a|s))$ for all (s, a).

Theorem 3.5. *Consider Algorithm 1 with Q-trace update, where we note that $c_{\max} = \bar{c}$, $\rho_{\max} = \bar{\rho}$, $D_c(s, a) = \sum_a \min(\bar{c}\pi_b(a|s), \pi(a|s))$, $D_\rho(s, a) = \sum_a \min(\bar{\rho}\pi_b(a|s), \pi(a|s))$, and $\omega = \mathcal{K}_{SA,\min} f(\gamma D_{c,\min})(1 - \gamma D_{\rho,\max})$. Suppose that Assumption 2.1 is satisfied, and $\bar{c} \leq \bar{\rho}$. Then, to achieve ϵ-accuracy, the sample complexity is $O\left(\frac{\log^2(1/\epsilon)}{\epsilon^2}\right) \tilde{\mathcal{O}}\left(\frac{1}{\omega^2}\right) \tilde{\mathcal{O}}\left(\frac{nf(\gamma\bar{c})^2(\gamma\bar{\rho}+1)^2}{(1-\gamma D_{\rho,\max})^2}\right)$.*

To avoid an exponential large variance, in view of the term $f(\gamma\bar{c})$ in our bound, we need to choose $\bar{c} \leq 1/\gamma$. The major difference between Q-trace and Retrace(λ) is that the importance sampling ratio $\rho(\cdot, \cdot)$ inside the temporal difference (line 4 of Algorithm 1) also involves a truncation. As shown in Section 2.3, due to introducing the truncation level $\bar{\rho}$, the algorithm converges to a biased limit $Q^{\pi,\rho}$ instead of Q^π. Such truncation bias can be controlled using Proposition 2.1. These observations agree with the results [19], where the finite-sample bounds of Q-trace were first established.

Compared to [19], we have an improved sample complexity. Specifically, the result in [19] implies a sample complexity of $\tilde{\mathcal{O}}(\frac{\log^2(1/\epsilon)nf(\gamma\bar{c})^2(\gamma\bar{\rho}+1)^2}{\epsilon^2\omega^3(1-\gamma D_{\rho,\max})^2})$, which has an additional factor of ω^{-1}. Since $\omega^{-1} \propto \mathcal{K}_{SA,\min}^{-1} \geq |\mathcal{S}||\mathcal{A}|$, our result improves the dependency on the size of the state-action space by a factor of at least $|\mathcal{S}||\mathcal{A}|$ compared to [19]. Similarly, since the V-trace algorithm [14] is an analog of the Q-trace algorithm, we can also improve the sample complexity for V-trace in [10].

In addition to analyzing existing algorithms, observe that our results, especially Theorem 2.2, provide sufficient conditions under which Algorithm 1 has provable finite-sample guarantees, and hence enable us to design new algorithms. As an example, in light of the Retrace(λ) algorithm and the Q-trace algorithm, one can take advantage of both algorithms to let $c(s, a) = \lambda_c \min(\bar{c}, \pi(a|s)/\pi_b(a|s))$ and $\rho(s, a) = \lambda_\rho \min(\bar{\rho}, \pi(a|s)/\pi_b(a|s))$, where λ_c, λ_ρ, $\bar{c}$, and $\bar{\rho}$ are tunable parameters. As long as $\lambda_c\bar{c} \leq \lambda_\rho\bar{\rho} < 1/\gamma$, Theorem 2.2 is applicable and hence finite-sample convergence is guaranteed. To avoid an exponentially large variance, we choose $\lambda_c\bar{c} \leq 1/\gamma$ so that there are no exponentially large terms in the term T_3 of sample complexity bound. After that, we can tune the rest of the parameters to further optimize the performance of the algorithm.

Sample Complexity Comparison. Now that we have derived the sample complexity bounds of various off-policy n-step TD-learning algorithms, we summarize them in the following table. For ease of exposition, we omit the common factor $\log^2(1/\epsilon)/(\epsilon^2\mathcal{K}_{SA,\min}^2)$ when presenting the sample complexity, and use $a \wedge b$ ($a \vee b$) to denote the minimum (maximum) of two real numbers a and b.

In view of Table 1, when $r_{\max} < 1/\gamma$, which indicates that the target policy π and the behavior policy π_b are relatively close to each other, Vanilla IS has the best performance since it has the best contraction factor, and the cumulative product of the generalized importance sampling ratios does not result in exponentially large variance. When $r_{\max} > 1/\gamma$, then Vanilla IS can potentially have exponentially large variance, while other four algorithms do not. In this case, among $Q^\pi(\lambda)$, TB(λ), and Retrace(λ), $Q^\pi(\lambda)$ has the best sample complexity bound. However, we need to point out that

Table 1: Summary of the Sample Complexity Bounds

Algorithm	$c(s,a)$	$\rho(s,a)$	Requirements	Sample Complexity
Vanilla IS	$\frac{\pi(a\mid s)}{\pi_b(a\mid s)}$	$\frac{\pi(a\mid s)}{\pi_b(a\mid s)}$	None	$\tilde{\mathcal{O}}\left(\frac{(\gamma r_{\max})^n+1)^2}{(1-\gamma^n)^2(1-\gamma)^2}\right)$
$Q^\pi(\lambda)$	λ	$\frac{\pi(a\mid s)}{\pi_b(a\mid s)}$	$\lambda \leq r_{\min}$	$\tilde{\mathcal{O}}\left(\frac{(\gamma r_{\max}+1)^2}{(1-\gamma)^4}\right)$
TB(λ)	$\lambda\pi(a\mid s)$	$\frac{\pi(a\mid s)}{\pi_b(a\mid s)}$	$\lambda < \frac{1}{(\pi_{b,\max}\vee\gamma\pi_{\max})}$	$\tilde{\mathcal{O}}\left(\frac{f(\gamma\lambda\pi_{\max})^2(\gamma r_{\max}+1)^2}{f(\gamma D_{c,\min})^2(1-\gamma)^4}\right)$
Retrace(λ)	$\lambda[1\wedge\frac{\pi(a\mid s)}{\pi_b(a\mid s)}]$	$\frac{\pi(a\mid s)}{\pi_b(a\mid s)}$	$\lambda \leq 1$	$\tilde{\mathcal{O}}\left(\frac{f(\gamma\lambda)^2(\gamma r_{\max}+1)^2}{f(\gamma D_{c,\min})^2(1-\gamma)^4}\right)$
Q-trace	$\bar{c}\wedge\frac{\pi(a\mid s)}{\pi_b(a\mid s)}$	$\bar{\rho}\wedge\frac{\pi(a\mid s)}{\pi_b(a\mid s)}$	$\bar{c}\leq\bar{\rho},\ \bar{c}<\frac{1}{\gamma}$	$\tilde{\mathcal{O}}\left(\frac{f(\gamma\bar{c})^2(\gamma\bar{\rho}+1)^2}{f(\gamma D_{c,\min})^2(1-\gamma D_{\rho,\max})^4}\right)$

the requirement $\lambda \leq r_{\min}$ for $Q^\pi(\lambda)$ is most restrictive, and the algorithm can easily diverge when this requirement is not satisfied, as evidenced by the numerical experiments presented in [22]. As for the Q-trace algorithm, although rigorously speaking it is not directly comparable with the other algorithms as it converges to a biased limit point, it is clear that using truncated importance sampling ratio for $\rho(\cdot,\cdot)$ can further reduce the sample complexity.

We want to mention that our comparison is based on the upper bounds we derived for the sample complexity, which may not be tight. To complete the story, one should also derive lower bounds on the sample complexity, which is an interesting future direction. Nevertheless, our comparison provides insight into the behavior of off-policy n-step TD-learning algorithms,

4 Proof sketch of Proposition 2.4

The idea is to construct a stochastic matrix M'' such that: (1) M'' dominates M in the sense that $M''_{ij} \geq M_{ij}$ for all i,j, and (2) the Markov chain associated with M'' is irreducible, hence admits a unique stationary distribution $\mu > 0$. Using μ as weights and we have the desired result. The detailed analysis is presented in Appendix A.5. We here present the construction of the stochastic matrix M''.

First of all, consider the special case where M itself is irreducible. Then we first scale up M by a factor of $1/(1-\omega)$ to obtain $M' = \frac{M}{1-\omega}$, which is clearly a substochastic matrix, with modulus zero. Hence there exists a stochastic matrix M'' that dominates M' (and also M). Moreover, since M'' is also irreducible, its associated Markov chain has a unique stationary distribution μ. This is equivalent to choosing $\theta = 1$ in Proposition 2.4. In fact, the matrix M being irreducible is only a sufficient condition to choose $\theta = 1$. What we need is the existence of a strictly positive stationary distribution of the stochastic matrix M'', which is guaranteed when M'' does not have transient states.

Now consider the general case where M is not necessarily irreducible. We construct the intermediate matrix M' by performing a convex combination of the matrix $\frac{M}{1-\omega}$ and the uniform stochastic matrix $\frac{E}{d}$, where E is the all one matrix, with weight $\frac{1-\omega}{1-\theta\omega}$. Specifically, for any $\theta \in (0,1)$, we define $M' = \left(\frac{1-\omega}{1-\theta\omega}\right)\frac{M}{1-\omega} + \left(1 - \frac{1-\omega}{1-\theta\omega}\right)\frac{E}{d}$. Note that M' is a non-negative matrix. In addition, since $M'\mathbf{1} \leq \frac{1-\omega}{1-\theta\omega}\mathbf{1} + \left(1 - \frac{1-\omega}{1-\theta\omega}\right)\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is the all one vector, the matrix M' is a substochastic matrix with modulus zero, and is also irreducible because all its entries are strictly positive. Therefore, there exists a stochastic matrix M'' such that $M'' \geq M'$. In addition, since M'' also has strictly positive entries, the Markov chain associated with M'' is irreducible, hence admits a unique stationary distribution $\mu \in \Delta^d$. By our construction, we can show a lower bound on the components of the stationary distribution μ.

5 Conclusion

In this work, we establish finite-sample guarantees of general n-step off-policy TD-learning algorithms. The key in our approach is to identify a generalized Bellman operator and establish its contraction property with respect to a weighted ℓ_p-norm for each $p \in [1,\infty)$, with a uniform contraction factor. Our results are used to derive finite-sample guarantees of variants of n-step off-policy TD-learning algorithms in the literature. Specifically, for $Q^\pi(\lambda)$, TB(λ), and Retrace(λ), we provide the first-known results, and for Q-trace, we improve the result in [19]. The finite-sample bounds we establish also provide insights about the trade-offs between the bias and the variance.

Acknowledgements

References

[1] Aggarwal, C. C. et al. (2016). *Recommender systems*, volume 1. Springer.

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

[3] Beck, A. (2017). *First-order methods in optimization*, volume 25. SIAM.

[4] Berman, A. and Plemmons, R. J. (1994). *Nonnegative matrices in the mathematical sciences*. SIAM.

[5] Bhandari, J., Russo, D., and Singal, R. (2018). A finite time analysis of temporal difference learning with linear function approximation. In *Conference On Learning Theory*, pages 1691–1692.

[6] Bottou, L., Curtis, F. E., and Nocedal, J. (2018). Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311.

[7] Cai, Q., Yang, Z., Lee, J. D., and Wang, Z. (2019). Neural temporal-difference learning converges to global optima. *Advances in Neural Information Processing Systems*, 32.

[8] Cayci, S., Satpathi, S., He, N., and Srikant, R. (2021). Sample Complexity and Overparameterization Bounds for Projection-Free Neural TD Learning. *Preprint arXiv:2103.01391*.

[9] Chen, Z., Maguluri, S. T., Shakkottai, S., and Shanmugam, K. (2020). Finite-Sample Analysis of Contractive Stochastic Approximation Using Smooth Convex Envelopes. *Advances in Neural Information Processing Systems*, 33.

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

[11] Cichosz, P. and Mulawka, J. J. (1995). Fast and efficient reinforcement learning with truncated temporal differences. In *Machine Learning Proceedings 1995*, pages 99–107. Elsevier.

[12] Dalal, G., Szörényi, B., Thoppe, G., and Mannor, S. (2018). Finite Sample Analyses for TD (0) With Function Approximation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[13] Dayan, P. (1992). The convergence of TD(λ) for general λ. *Machine learning*, 8(3-4):341–362.

[14] Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al. (2018). IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures. In *International Conference on Machine Learning*, pages 1407–1416.

[15] Glynn, P. W. and Iglehart, D. L. (1989). Importance sampling for stochastic simulations. *Management science*, 35(11):1367–1392.

[16] Gottesman, O., Futoma, J., Liu, Y., Parbhoo, S., Celi, L., Brunskill, E., and Doshi-Velez, F. (2020). Interpretable off-policy evaluation in reinforcement learning by highlighting influential transitions. In *International Conference on Machine Learning*, pages 3658–3667. PMLR.

[17] Harutyunyan, A., Bellemare, M. G., Stepleton, T., and Munos, R. (2016). Q(λ) with Off-Policy Corrections. In *International Conference on Algorithmic Learning Theory*, pages 305–320. Springer.

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

[19] Khodadadian, S., Chen, Z., and Maguluri, S. T. (2021). Finite-Sample Analysis of Off-Policy Natural Actor-Critic Algorithm. *The 38th International Conference on Machine Learning*.

[20] Levin, D. A. and Peres, Y. (2017). *Markov chains and mixing times*, volume 107. American Mathematical Soc.

[21] Maei, H. R. (2011). *Gradient temporal-difference learning algorithms*. University of Alberta.

[22] Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. G. (2016). Safe and efficient off-policy reinforcement learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1054–1062.

[23] Nowé, A., Vrancx, P., and De Hauwere, Y.-M. (2012). Game theory and multi-agent reinforcement learning. In *Reinforcement Learning*, pages 441–470. Springer.

[24] Precup, D., Sutton, R. S., and Singh, S. P. (2000). Eligibility Traces for Off-Policy Policy Evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 759–766.

[25] Sallab, A. E., Abdou, M., Perot, E., and Yogamani, S. (2017). Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19):70–76.

[26] Shalev-Shwartz, S., Shammah, S., and Shashua, A. (2016). Safe, multi-agent, reinforcement learning for autonomous driving. *Preprint arXiv:1610.03295*.

[27] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354.

[28] Singh, S. P., Jaakkola, T., and Jordan, M. I. (1994). Learning without state-estimation in partially observable Markovian decision processes. In *Machine Learning Proceedings 1994*, pages 284–292. Elsevier.

[29] Srikant, R. and Ying, L. (2019). Finite-time error bounds for linear stochastic approximation and TD learning. In *Conference on Learning Theory*, pages 2803–2830.

[30] Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44.

[31] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[32] Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, C., and Wiewiora, E. (2009). Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 993–1000.

[33] Sutton, R. S., Mahmood, A. R., and White, M. (2016). An emphatic approach to the problem of off-policy temporal-difference learning. *The Journal of Machine Learning Research*, 17(1):2603–2631.

[34] Thoppe, G. and Borkar, V. (2019). A concentration bound for stochastic approximation via Alekseev's formula. *Stochastic Systems*, 9(1):1–26.

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

[36] Tsitsiklis, J. N. and Van Roy, B. (1997). Analysis of temporal-difference learning with function approximation. In *Advances in neural information processing systems*, pages 1075–1081.

[37] Van Seijen, H., Mahmood, A. R., Pilarski, P. M., Machado, M. C., and Sutton, R. S. (2016). True online temporal-difference learning. *The Journal of Machine Learning Research*, 17(1):5057–5096.

[38] Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. King's College, Cambridge United Kingdom.

[39] Zhang, Q., Guan, Y., and Tsiotras, P. (2020). Learning Nash Equilibria in Zero-Sum Stochastic Games via Entropy-Regularized Policy Approximation. *Preprint arXiv:2009.00162*.

[40] Zhao, Y., Zeng, D., Socinski, M. A., and Kosorok, M. R. (2011). Reinforcement learning strategies for clinical trials in nonsmall cell lung cancer. *Biometrics*, 67(4):1422–1433.

[41] Zou, L., Xia, L., Ding, Z., Song, J., Liu, W., and Yin, D. (2019). Reinforcement learning to optimize long-term user engagement in recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2810–2818.

Checklist

1. For all authors...

    (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
    (b) Did you describe the limitations of your work? [Yes] See Section 5.
    (c) Did you discuss any potential negative societal impacts of your work? [N/A]
    (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

    (a) Did you state the full set of assumptions of all theoretical results? [Yes]
    (b) Did you include complete proofs of all theoretical results? [Yes] All proofs are presented in the Appendix based on the order they appear in the paper.

3. If you ran experiments...

    (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
    (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
    (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
    (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

    (a) If your work uses existing assets, did you cite the creators? [N/A]
    (b) Did you mention the license of the assets? [N/A]
    (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
    (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
    (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

    (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
    (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
    (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendices

A Technical Details in Section 2

A.1 Proof of Proposition 2.1

For any $Q_1, Q_2 \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, and state-action pairs (s, a), using the definition of $\mathcal{H}_\rho(\cdot)$ and we have

$$
|[\mathcal{H}_\rho(Q_1)](s, a) - [\mathcal{H}_\rho(Q_2)](s, a)|
$$

$$
= \gamma \left| \sum_{s' \in \mathcal{A}} P_a(s, s') \sum_{a' \in \mathcal{A}} \pi_b(a'|s')\rho(s', a')(Q_1(s', a') - Q_2(s', a')) \right|
$$

$$
\leq \gamma \sum_{s' \in \mathcal{A}} P_a(s, s') \sum_{a' \in \mathcal{A}} \pi_b(a'|s')\rho(s', a')|Q_1(s', a') - Q_2(s', a')|
$$

$$
\leq \gamma \|Q_1 - Q_2\|_\infty \sum_{s' \in \mathcal{A}} P_a(s, s') \sum_{a' \in \mathcal{A}} \pi_b(a'|s')\rho(s', a')
$$

$$
\leq \gamma \sum_{s' \in \mathcal{A}} P_a(s, s') D_{\rho,\max} \|Q_1 - Q_2\|_\infty
$$

$$
= \gamma D_{\rho,\max} \|Q_1 - Q_2\|_\infty.
$$

It follows that $\|\mathcal{H}_\rho(Q_1) - \mathcal{H}_\rho(Q_2)\|_\infty \leq \gamma D_{\rho,\max} \|Q_1 - Q_2\|_\infty$. Since $D_{\rho,\max} < 1/\gamma$, the operator $\mathcal{H}_\rho(\cdot)$ is a contraction mapping with respect to $\|\cdot\|_\infty$, with contraction factor $\gamma D_{\rho,\max}$.

(1) We now derive the upper bound on $\|Q^\pi - Q^{\pi,\rho}\|_\infty$. Since $Q^\pi = \mathcal{H}_\pi(Q^\pi)$ and $Q^{\pi,\rho} = \mathcal{H}_\rho(Q^{\pi,\rho})$, we have

$$
|Q^\pi(s, a) - Q^{\pi,\rho}(s, a)|
$$
$$
= |[\mathcal{H}_\pi(Q^\pi)](s, a) - [\mathcal{H}_\rho(Q^{\pi,\rho})](s, a)|
$$
$$
= |[\mathcal{H}_\pi(Q^\pi)](s, a) - [\mathcal{H}_\rho(Q^\pi)](s, a) + [\mathcal{H}_\rho(Q^\pi)](s, a) - [\mathcal{H}_\rho(Q^{\pi,\rho})](s, a)|
$$
$$
\leq |[\mathcal{H}_\pi(Q^\pi)](s, a) - [\mathcal{H}_\rho(Q^\pi)](s, a)| + |[\mathcal{H}_\rho(Q^\pi)](s, a) - [\mathcal{H}_\rho(Q^{\pi,\rho})](s, a)|
$$
$$
= \gamma \left| \sum_{s' \in \mathcal{S}} P_a(s, s') \sum_{a' \in \mathcal{A}} (\pi(a'|s') - \pi_b(a'|s')\rho(s', a')) Q^\pi(s', a') \right| + \gamma D_{\rho,\max} \|Q^\pi - Q^{\pi,\rho}\|_\infty
$$
$$
\leq \frac{\gamma}{1 - \gamma} \sum_{s' \in \mathcal{S}} P_a(s, s') \sum_{a' \in \mathcal{A}} |\pi(a'|s') - \pi_b(a'|s')\rho(s', a')| + \gamma D_{\rho,\max} \|Q^\pi - Q^{\pi,\rho}\|_\infty \qquad (*)
$$
$$
\leq \frac{\gamma}{1 - \gamma} \max_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} |\pi(a|s) - \pi_b(a|s)\rho(s, a)| + \gamma D_{\rho,\max} \|Q^\pi - Q^{\pi,\rho}\|_\infty,
$$

where in Eq. $(*)$ we used the inequality $|Q^\pi(s, a)| \leq \sum_{k=0}^\infty \gamma^k = \frac{1}{1-\gamma}$ for all (s, a). Therefore, we have

$$
\|Q^\pi - Q^{\pi_\rho}\|_\infty \leq \frac{\gamma}{1 - \gamma} \max_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} |\pi(a|s) - \pi_b(a|s)\rho(s, a)| + \gamma D_{\rho,\max} \|Q^\pi - Q^{\pi,\rho}\|_\infty.
$$

Rearranging terms and we obtain the desired result.

(2) To prove the upper bound on $\|Q^{\pi,\rho}\|_\infty$, we begin with the fixed-point equation

$$
Q^{\pi,\rho} = \mathcal{H}_\rho(Q^{\pi,\rho}) = R + \gamma P_{\pi_\rho} D_\rho Q^{\pi,\rho}, \qquad (5)
$$

where we recall the definition of D_ρ and π_ρ in Section 2. Eq. (5) is equivalent to $Q^{\pi,\rho} = (I - \gamma P_{\pi_\rho} D_\rho)^{-1} R$. Therefore, we have

$$
\|Q^{\pi,\rho}\|_\infty = \|(I - \gamma P_{\pi_\rho} D_\rho)^{-1} R\|_\infty \leq \|(I - \gamma P_{\pi_\rho} D_\rho)^{-1}\|_\infty \|R\|_\infty \leq \frac{1}{1 - \gamma D_{\rho,\max}}.
$$

A.2 On the Fixed-Point of the Operator $\mathcal{H}_\rho(\cdot)$

Suppose that for the state-action pair (s_0, a_0), we have $\rho(s_0, a_0) \neq \pi(a_0|s_0)/\pi_b(a_0|s_0)$. Let an MDP be that the transition probability matrix is an identity matrix for each action, and the reward is zero for all state-action pairs except at (s_0, a_0), where it is equal to 1.

In this case, it is clear that for any policy π, we have $Q^\pi(s, a) = 0$ for all $(s, a) \neq (s_0, a_0)$, and $Q^\pi(s_0, a_0) = \frac{1}{1-\gamma}$. Suppose that $Q^\pi = Q^{\pi,\rho}$. then we have

$$
\begin{aligned}
0 &= Q^\pi(s_0, a_0) - Q^{\pi,\rho}(s_0, a_0) \\
&= \gamma \sum_{s' \in \mathcal{S}} P_{a_0}(s_0, s') \sum_{a' \in \mathcal{A}} \left(\pi(a'|s') - \pi_b(a'|s')\rho(s', a')\right) Q^\pi(s', a') \\
&= \left(\pi(a_0|s_0) - \pi_b(a_0|s_0)\rho(s_0, a_0)\right) Q^\pi(s_0, a_0) \\
&= \frac{1}{1-\gamma} \left(\pi(a_0|s_0) - \pi_b(a_0|s_0)\rho(s_0, a_0)\right).
\end{aligned}
$$

This contradicts to the fact that $\rho(s_0, a_0) \neq \pi(a_0|s_0)/\pi_b(a_0|s_0)$. Therefore, we have $Q^\pi \neq Q^{\pi,\rho}$.

A.3 Proof of Proposition 2.2

Recall the definition of $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ in Eq. (2):

$$
\tilde{\mathcal{B}}_{c,\rho}(Q) = \mathcal{K}_{SA}(\mathcal{B}_{c,\rho}(Q) - Q) + Q = \mathcal{K}_{SA}\mathcal{T}_c(\mathcal{H}_\rho(Q) - Q) + Q.
$$

We first explicitly compute the operators $\mathcal{T}_c(\cdot)$ and $\mathcal{H}_\rho(\cdot)$. For the operator $\mathcal{H}_\rho(\cdot)$, we have from its definition that

$$
\begin{aligned}
[\mathcal{H}_\rho(Q)](s, a) &= \mathcal{R}(s, a) + \gamma \mathbb{E}_{\pi_b}[\rho(S_{k+1}, A_{k+1})Q(S_{k+1}, A_{k+1}) \mid S_k = s, A_k = a] \\
&= \mathcal{R}(s, a) + \gamma \sum_{s'} P_a(s, s') \sum_{a'} \pi_b(a'|s')\rho(s', a')Q(s', a') \\
&= \mathcal{R}(s, a) + \gamma \sum_{s'} P_a(s, s') \sum_{a'} \frac{\pi_b(a'|s')\rho(s', a')}{D_\rho(s', a')} D_\rho(s', a')Q(s', a') \\
&= \mathcal{R}(s, a) + \gamma \sum_{s',a'} P_a(s, s')\pi_\rho(a'|s')D_\rho(s', a')Q(s', a') \\
&= [R + P_{\pi_\rho}D_\rho Q](s, a).
\end{aligned}
$$

Note that $P_{\pi_\rho} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}||\mathcal{A}|}$ here is the transition probability matrix of the Markov chain $\{(S_k, A_k)\}$ under π_ρ, i.e., $P_{\pi_\rho}((s, a), (s', a')) = P_a(s, s')\pi_\rho(a'|s')$ for any (s, a) and (s', a'). Hence we have

$$
\mathcal{H}_\rho(Q) = R + P_{\pi_\rho}D_\rho Q.
$$

As for the operator $\mathcal{T}_c(\cdot)$, similarly using the Markov property and the tower property of conditional expectation, we have $\mathcal{T}_c(Q) = \sum_{i=0}^{n-1}(\gamma P_{\pi_c}D_c)^i Q$. It follows that

$$
\begin{aligned}
\tilde{\mathcal{B}}_{c,\rho}(Q) &= \mathcal{K}_{SA}\mathcal{T}_c(\mathcal{H}_\rho(Q) - Q) + Q \\
&= \mathcal{K}_{SA} \sum_{i=0}^{n-1}(\gamma P_{\pi_c}D_c)^i(R + \gamma P_{\pi_\rho}D_\rho Q - Q) + Q \\
&= \underbrace{\left[I - \mathcal{K}_{SA} \sum_{i=0}^{n-1}(\gamma P_{\pi_c}D_c)^i(I - \gamma P_{\pi_\rho}D_\rho)\right]}_{A} Q + \underbrace{\mathcal{K}_{SA} \sum_{i=0}^{n-1}(\gamma P_{\pi_c}D_c)^i R}_{b}.
\end{aligned}
$$

A.4 Proof of Proposition 2.3

Consider the matrix A given in Proposition 2.2. To show that A is a substochastic matrix with a positive modulus, we first show that A is non-negative. Observe that

$$
A = I - \mathcal{K}_{SA} \sum_{i=0}^{n-1}(\gamma P_{\pi_c}D_c)^i + \mathcal{K}_{SA} \sum_{i=0}^{n-1}(\gamma P_{\pi_c}D_c)^i \gamma P_{\pi_\rho}D_\rho
$$

$$= (I - \mathcal{K}_{SA}) - \mathcal{K}_{SA} \sum_{i=1}^{n-1} (\gamma P_{\pi_c} D_c)^i + \mathcal{K}_{SA} \sum_{i=0}^{n-1} (\gamma P_{\pi_c} D_c)^i \gamma P_{\pi_\rho} D_\rho$$

$$= (I - \mathcal{K}_{SA}) - \mathcal{K}_{SA} \sum_{i=0}^{n-2} (\gamma P_{\pi_c} D_c)^{i+1} + \mathcal{K}_{SA} \sum_{i=0}^{n-1} (\gamma P_{\pi_c} D_c)^i \gamma P_{\pi_\rho} D_\rho$$

$$= (I - \mathcal{K}_{SA}) + \mathcal{K}_{SA} \sum_{i=0}^{n-2} (\gamma P_{\pi_c} D_c)^i \gamma (P_{\pi_\rho} D_\rho - P_{\pi_c} D_c) + \mathcal{K}_{SA} (\gamma P_{\pi_c} D_c)^{n-1} \gamma P_{\pi_\rho} D_\rho. \quad (6)$$

It remains to show that the matrix $P_{\pi_\rho} D_\rho - P_{\pi_c} D_c$ has non-negative entries. For any (s, a) and (s', a'), since $c(s', a') \le \rho(s', a')$ for all (s', a'), we have

$$[P_{\pi_\rho} D_\rho - P_{\pi_c} D_c]((s, a), (s', a')) = P_a(s, s') \pi_b(a'|s')(\rho(s', a') - c(s', a')) \ge 0.$$

We next show that $A\mathbf{1} \le (1 - \omega)\mathbf{1}$, where $\mathbf{1} \in \mathbb{R}^d$ is the all one vector. Since A is non-negative and $D_{\rho,\max} < 1/\gamma$ for all (s, a), we have

$$\mathcal{K}_{SA} \sum_{i=0}^{n-1} (\gamma P_{\pi_c} D_c)^i (I - \gamma P_{\pi_\rho} D_\rho)\mathbf{1} \ge \mathcal{K}_{SA} \sum_{i=0}^{n-1} (\gamma P_{\pi_c} D_c)^i (I - \gamma P_{\pi_\rho} D_{\rho,\max})\mathbf{1}$$

$$= (1 - \gamma D_{\rho,\max}) \mathcal{K}_{SA} \sum_{i=0}^{n-1} (\gamma P_{\pi_c} D_c)^i \mathbf{1}$$

$$\ge \mathcal{K}_{SA,\min} \sum_{i=0}^{n-1} (\gamma D_{c,\min})^i (1 - \gamma D_{\rho,\max})\mathbf{1}$$

$$= \mathcal{K}_{SA,\min} f(\gamma D_{c,\min})(1 - \gamma D_{\rho,\max})\mathbf{1}.$$

It follows that

$$A\mathbf{1} = \left[I - \mathcal{K}_{SA} \sum_{i=0}^{n-1} (\gamma P_{\pi_c} D_c)^i (I - \gamma P_{\pi_\rho} D_\rho) \right] \mathbf{1} \le [1 - \mathcal{K}_{SA,\min} f(\gamma D_{c,\min})(1 - \gamma D_{\rho,\max})]\mathbf{1}.$$

This implies that A is a substochastic matrix with modulus $\omega = \mathcal{K}_{SA,\min} f(\gamma D_{c,\min})(1 - \gamma D_{\rho,\max})$.

A.5 Proof of Proposition 2.4

Consider a substochastic matrix $M \in \mathbb{R}^{d \times d}$ with modulus $\beta \in (0, 1)$. For any $\theta \in (0, 1)$, let

$$M' = \frac{M}{1 - \theta\beta} + \frac{\beta(1 - \theta)}{1 - \theta\beta} \frac{E}{d},$$

where E is the all one matrix. It is clear that $M' > 0$. Moreover, since

$$M'\mathbf{1} \le \frac{1 - \beta}{1 - \theta\beta}\mathbf{1} + \frac{\beta(1 - \theta)}{1 - \theta\beta}\mathbf{1} = \mathbf{1},$$

the matrix M' is a substochastic matrix with modulus 0, there exists a stochastic matrix M'' such that $M'' \ge M' > 0$. Since M'' has strictly positive entries, the Markov chain associated with the stochastic matrix M'' is irreducible and aperiodic, hence admits a unique stationary distribution $\mu \in \Delta^d$. In the special case where M itself is irreducible, we are allowed to choose $\theta = 1$ in the preceding construction process, and the resulting stochastic matrix M'' is also guaranteed to be irreducible, and hence has a unique stationary distribution μ. Since $\mu^\top = \mu^\top M''$, we have

$$\mu^\top = \mu^\top M'' \ge \mu^\top M' \ge \mu^\top \frac{\beta(1 - \theta)}{1 - \theta\beta} \frac{E}{d} = \frac{\beta(1 - \theta)}{(1 - \theta\beta)d}\mathbf{1}^\top.$$

This proves the lower bound on the entries of μ.

Now using μ as the weight vector and we have for any $p \in [1, \infty)$ and $x \in \mathbb{R}^d$:

$$\|Mx\|_{\mu,p}^p = \sum_i \mu_i \left| \sum_j M_{ij} x_j \right|^p$$

16

$$= \sum_i \mu_i \left(\sum_\ell M_{i\ell} \right)^p \left| \sum_j \frac{M_{ij}}{\sum_\ell M_{i\ell}} x_j \right|^p$$

$$\leq \sum_i \mu_i \left(\sum_\ell M_{i\ell} \right)^{p-1} \sum_j M_{ij} |x_j|^p \qquad \text{(Jensen's inequality)}$$

$$\leq (1-\beta)^{p-1} \sum_i \mu_i \sum_j M_{ij} |x_j|^p$$

$$\leq (1-\beta)^{p-1}(1-\theta\beta) \sum_i \mu_i \sum_j M'_{ij} |x_j|^p \qquad \text{(definition of } M')$$

$$\leq (1-\beta)^{p-1}(1-\theta\beta) \sum_i \mu_i \sum_j M''_{ij} |x_j|^p \qquad \text{(definition of } M'')$$

$$= (1-\beta)^{p-1}(1-\theta\beta) \sum_j |x_j|^p \sum_i \mu_i M''_{ij} \qquad \text{(change of summation order)}$$

$$= (1-\beta)^{p-1}(1-\theta\beta) \sum_j \mu_j |x_j|^p \qquad (\mu^\top M'' = \mu^\top)$$

$$= (1-\beta)^{p-1}(1-\theta\beta)\|x\|_{\mu,p}^p.$$

It follows that $\|Mx\|_{\mu,p} \leq (1-\omega)^{1-1/p}(1-\theta\beta)^{1/p}\|x\|_{\mu,p}$ for any $x \in \mathbb{R}^d$ and $p \in [1,\infty)$. Using the definition of induced matrix norm immediately gives the result.

A.6 Proof of Theorem 2.2

We first state a more general result in the following, which implies Theorem 2.2.

Theorem A.1. *Consider the iterates $\{Q_k\}$ generated by Algorithm 1. Suppose that Assumption 2.1 is satisfied, and $c(s,a) \leq \rho(s,a)$ for all (s,a) and $D_{\rho,\max} < 1/\gamma$. Then for any $\theta \in (0,1)$, there exists a weighted ℓ_p-norm with weights $\mu \in \Delta^{|\mathcal{S}||\mathcal{A}|}$ satisfying $\mu_{\min} \geq \frac{\omega(1-\theta)}{(1-\theta\omega)|\mathcal{S}||\mathcal{A}|}$ such that the following inequality holds when the constant stepsize α is chosen such that $\alpha\tau_{\alpha,n} \leq \frac{\theta\mu_{\min}^{2/p}\omega}{2052pf(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2}$:*

$$\mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_{\mu,p}^2] \leq \tilde{\zeta}_1(1-\theta\omega\alpha)^{k-\tau_{\alpha,n}} + \tilde{\zeta}_2 \frac{pf(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2}{\mu_{\min}^{2/p}\omega}\alpha\tau_{\alpha,n},$$

where $\tilde{\zeta}_1 = (\|Q_0 - Q^{\pi,\rho}\|_{\mu,p} + \|Q_0\|_{\mu,p} + 1)^2$, and $\tilde{\zeta}_2 = 228(3\|Q^{\pi,\rho}\|_{\mu,p} + 1)^2$.

By using the inequality that $\mu_{\min}^{1/p}\|\cdot\|_p \leq \|\cdot\|_{\mu,p}$ (where $\|\cdot\|_p$ is the unweighted ℓ_p-norm), Theorem A.1 implies the following finite-sample bound on $\mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_p]$.

Corollary A.1.1. *Under same assumptions as Theorem 2.1, we have for all $k \geq \tau_{\alpha,n}$:*

$$\mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_p^2] \leq \frac{\tilde{\zeta}_1}{\mu_{\min}^{2/p}}(1-\theta\omega\alpha)^{k-\tau_{\alpha,n}} + \frac{\tilde{\zeta}_2}{\mu_{\min}^{2/p}} \frac{pf(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2}{\mu_{\min}^{2/p}\omega}\alpha\tau_{\alpha,n},$$

To proceed and prove Theorem 2.2, observe that for any $p \geq 1$ we have

$$\mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_\infty^2] \leq \mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_p^2]$$

$$\leq \frac{\tilde{\zeta}_1}{\mu_{\min}^{2/p}}(1-\theta\omega\alpha)^{k-\tau_{\alpha,n}} + \frac{\tilde{\zeta}_2 pf(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2}{\mu_{\min}^{4/p}\omega}\alpha\tau_{\alpha,n}.$$

Let $\theta = 1/2$ and $p = 4\log(1/\mu_{\min})$. Then we have

$$\frac{1}{\mu_{\min}^{2/p}} = \mu_{\min}^{-\frac{1}{2\log(1/\mu_{\min})}} = \mu_{\min}^{\frac{1}{2\log(\mu_{\min})}} = \sqrt{e} \leq 2, \quad \text{and}$$

$$\frac{p}{\mu_{\min}^{4/p}} \leq 4e\log(1/\mu_{\min}) \leq 4e\log\left(\frac{2|\mathcal{S}||\mathcal{A}|}{\omega}\right). \qquad \text{(Using the lower bound on } \mu_{\min})$$

It follows that when $\alpha\tau_{\alpha,n} \leq \frac{\omega}{32832\log(2|\mathcal{S}||\mathcal{A}|/\omega)f(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2}$, we have for all $k \geq \tau_{\alpha,n}$:

$$\mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_\infty^2] \leq 2\tilde{\zeta}_1\left(1 - \frac{\omega\alpha}{2}\right)^{k-\tau_{\alpha,n}} + 4e\tilde{\zeta}_2\frac{f(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2\log(2|\mathcal{S}||\mathcal{A}|/\omega)}{\omega}\alpha\tau_{\alpha,n}$$

$$= \zeta_1\left(1 - \frac{\omega\alpha}{2}\right)^{k-\tau_{\alpha,n}} + \zeta_2\frac{f(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2\log(2|\mathcal{S}||\mathcal{A}|/\omega)}{\omega}\alpha\tau_{\alpha,n},$$

where in the last line we used $2\tilde{\zeta}_1 \leq \zeta_1 = 2(\|Q_0 - Q^{\pi,\rho}\|_\infty + \|Q_0\|_\infty + 1)^2$, and $4e\tilde{\zeta}_2 \leq \zeta_2 = 912e(3\|Q^{\pi,\rho}\|_\infty + 1)^2$. This proves Theorem 2.2.

A.6.1 Proof of Theorem A.1

To prove Theorem A.1, we use a Lyapunov drift argument. We next present two approaches for proving Theorem A.1. One is by directly using $W(Q) = \frac{1}{2}\|Q\|_{\mu,p}^2$ as the Lyapunov function. Another one is by applying [10, Theorem 2.1], which studies general stochastic approximation under contraction assumption.

We begin by rewriting Algorithm 1 using simplified notation. Let $Y_k = (S_k, A_k, \cdots, S_{k+n}, A_{k+n})$ for all $k \geq 0$, which is clearly a Markov chain, with finite state-space denoted by $\mathcal{Y}$. Note that under Assumption 2.1 the Markov chain $\{Y_k\}$ has a unique stationary distribution $\kappa_Y \in \Delta^{|\mathcal{Y}|}$. Define an operator $F : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \times \mathcal{Y} \mapsto \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ by

$$[F(Q, y)](s, a) = [F(Q, s_0, a_0, ..., s_n, a_n)](s, a)$$

$$= \mathbb{I}_{\{(s_0, a_0) = (s, a)\}} \sum_{i=0}^{n-1} \gamma^i \prod_{j=1}^{i} c(s_j, a_j)(\mathcal{R}(s_i, a_i) + \gamma\rho(s_{i+1}, a_{i+1})Q(s_{i+1}, a_{i+1}) - Q(s_i, a_i)) + Q(s, a).$$

Then the update equation of Algorithm 1 can be equivalently written by $Q_{k+1} = Q_k + \alpha(F(Q_k, Y_k) - Q_k)$. We next establish in the following proposition the properties of the operators $F(\cdot, \cdot)$ and the Markov chain $\{Y_k\}$, which will be useful in both approaches we present later.

Proposition A.1. *The following statements hold.*

(1) The operator $F(\cdot)$ satisfies for any Q_1, Q_2 and y:

(a) $\|F(Q_1, y) - F(Q_2, y)\|_{\mu,p} \leq \frac{2}{\mu_{\min}^{1/p}}f(\gamma c_{\max})(\gamma\rho_{\max}+1)\|Q_1 - Q_2\|_{\mu,p}$,

(b) $\|F(\mathbf{0}, y)\|_{\mu,p} \leq f(\gamma c_{\max})$.

(2) For any $k \geq 0$ and $n \geq 0$, we have $\max_{y\in\mathcal{Y}}\|P^{k+n+1}(y, \cdot) - \kappa_Y(\cdot)\|_{TV} \leq C\sigma^k$.

(3) For any Q, we have $\mathbb{E}_{Y\sim\kappa_Y}[F(Q, Y)] = \tilde{\mathcal{B}}_{c,\rho}(Q)$.

We now present our first approach of proving Theorem A.1, where we directly use $W(Q) = \frac{1}{2}\|Q\|_{\mu,p}^2$ as the Lyapunov function.

First Approach: Note that the function $W(Q) = \frac{1}{2}\|Q\|_{\mu,p}^2$ is a $(p-1)$-smooth function with respect to $\|\cdot\|_{\mu,p}$ [3], i.e., $W(Q_2) \leq W(Q_1) + \langle\nabla W(Q_1), Q_2 - Q_1\rangle + \frac{p-1}{2}\|Q_1 - Q_2\|_{\mu,p}^2$ for any $Q_1, Q_2 \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$. Therefore, using the update equation of Algorithm 1, we have for any $k \geq 0$:

$$\mathbb{E}[W(Q_{k+1} - Q^{\pi,\rho})] \leq \mathbb{E}[W(Q_k - Q^{\pi,\rho})] + \alpha_k\mathbb{E}[\langle\nabla W(Q_k - Q^{\pi,\rho}), Q_{k+1} - Q_k\rangle]$$

$$+ \frac{(p-1)\alpha_k^2}{2}\mathbb{E}[\|Q_{k+1} - Q_k\|_p^2].$$

The rest of the proof is identical to that of [10, Theorem 2.1] (where Proposition A.1 plays an important role), and is omitted. Here, we can directly use $W(Q)$ as a Lyapunov function because it is smooth. In contrast, [9, 10] study the more general settings when it is not smooth. In that case, the Lyapunov function is obtained by using a smoothing technique involving generalized Moreau envelop and infimal convolution to obtain a smooth approximation of $W(Q)$. One can of course, directly apply the result in [9, 10], which we present as a second approach.

18

Second Approach: We next present how to apply [10, Theorem 2.1] to obtain the results. We begin by restating Theorem 2.1 of [10] in the case of weighted ℓ_p-norm contraction with weights $\{\mu_i\}_{1 \leq i \leq d}$. Using the notation of [10], we choose the smoothing norm $\|\cdot\|_s$ to be the same norm as the contraction norm: $\|\cdot\|_{\mu,p}$.

Theorem A.2 (Theorem 2.1 in [10]). *Consider the SA algorithm*

$$x_{k+1} = x_k + \alpha(F(x_k, Y_k) - x_k). \tag{7}$$

Suppose that

(1) The random process $\{Y_k\}$ is a Markov chain (denoted by $\mathcal{MC}_Y$) with finite state-space $\mathcal{Y}$. In addition, $\{Y_k\}$ has a unique stationary distribution κ_Y, and there exist $C_1 > 0$ and $\sigma_1 \in (0,1)$ such that $\max_{y \in \mathcal{Y}} \|P^k(y, \cdot) - \kappa_Y(\cdot)\|_{TV} \leq C_1 \sigma_1^k$ for all $k \geq 0$.

(2) The operator $F : \mathbb{R}^d \times \mathcal{Y} \mapsto \mathbb{R}^d$ satisfies for any $x_1, x_2 \in \mathbb{R}^d$ and $y \in \mathcal{Y}$

    (a) $\|F(x_1, y) - F(x_2, y)\|_{\mu,p} \leq a_1 \|x_1 - x_2\|_{\mu,p}$, where $a_1 > 0$ is a constant,

    (b) $\|F(\mathbf{0}, y)\|_{\mu,p} \leq b_1$, where $b_1 > 0$ is a constant.

(3) The expected operator $\bar{F} : \mathbb{R}^d \mapsto \mathbb{R}^d$ defined by $\bar{F}(x) = \mathbb{E}_{Y \sim \kappa_Y}[F(x, Y)]$ satisfies $\bar{F}(x^*) = x^*$, and is a contraction mapping with respect to $\|\cdot\|_{\mu,p}$, with contraction factor $\gamma_c \in (0,1)$.

(4) The constant stepsize α is chosen such that $\alpha t_\alpha(\mathcal{MC}_Y) \leq \frac{1-\gamma_c}{228p(a_1+1)^2}$.

Then we have for all $k \geq t_\alpha(\mathcal{MC}_Y)$ that

$$\mathbb{E}[\|x_k - x^*\|_{\mu,p}^2] \leq \tilde{c}_1(1 - (1-\gamma_c)\alpha)^{k - t_\alpha(\mathcal{MC}_Y)} + \frac{228p\tilde{c}_2}{(1-\gamma_c)}\alpha t_\alpha(\mathcal{MC}_Y),$$

where $\tilde{c}_1 = (\|x_0 - x^*\|_{\mu,p} + \|x_0\|_{\mu,p} + b_1/(a_1+1))^2$ and $\tilde{c}_2 = ((a_1+1)\|x^*\|_{\mu,p} + b_1)^2$.

Proposition A.1 in conjunction with Theorem 2.1 imply that the requirements for applying Theorem A.2 are satisfied. For any $\theta \in (0,1)$, when the constant stepsize α is chosen such that $\alpha\tau_{\alpha,n} \leq \frac{\theta\mu_{\min}^{2/p}\omega}{2052pf(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2}$, we have for any $k \geq \tau_{\alpha,n}$:

$$\mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_{\mu,p}^2] \leq \tilde{\zeta}_1(1 - \theta\omega\alpha)^{k-\tau_{\alpha,n}} + \tilde{\zeta}_2\frac{pf(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2}{\mu_{\min}^{2/p}\omega}\alpha\tau_{\alpha,n},$$

where $\tilde{\zeta}_1 = (\|Q_0 - Q^{\pi,\rho}\|_{\mu,p} + \|Q_0\|_{\mu,p} + 1)^2$, and $\tilde{\zeta}_2 = 228(3\|Q^{\pi,\rho}\|_{\mu,p} + 1)^2$.

A.6.2 Proof of Proposition A.1

(1) For any $Q_1, Q_2 \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and $y = (s_0, a_0, \cdots, s_n, a_n) \in \mathcal{Y}$, we have

$$\|F(Q_1, s_0, a_0, ..., s_n, a_n) - F(Q_2, s_0, a_0, ..., s_n, a_n)\|_{\mu,p}$$

$$\leq \left[\sum_{s,a} \mu(s,a)\left(\mathbb{I}_{\{(s,a)=(s_0,a_0)\}}\sum_{i=0}^{n-1}(\gamma c_{\max})^i(\gamma\rho_{\max}+1)\|Q_1 - Q_2\|_\infty\right)^p\right]^{1/p}$$

$$+ \|Q_1 - Q_2\|_{\mu,p} \qquad\qquad\qquad\qquad\qquad \text{(Triangle inequality)}$$

$$= f(\gamma c_{\max})(\gamma\rho_{\max}+1)\|Q_1 - Q_2\|_\infty + \|Q_1 - Q_2\|_{\mu,p}.$$

$$\leq \frac{2}{\mu_{\min}^{1/p}}f(\gamma c_{\max})(\gamma\rho_{\max}+1)\|Q_1 - Q_2\|_{\mu,p}.$$

Similarly, for any $y = (s_0, a_0, \cdots, s_n, a_n) \in \mathcal{Y}$, we have

$$\|F(\mathbf{0}, s_0, a_0, ..., s_n, a_n)\|_{\mu,p} \leq \left[\sum_{s,a} \mu(s,a)\mathbb{I}_{\{(s,a)=(s_0,a_0)\}}\left(\sum_{i=0}^{n-1}(\gamma c_{\max})^i\right)^p\right]^{1/p} \leq f(\gamma c_{\max}).$$

(2) Under Assumption 2.1, it is clear that $\{Y_k\}$ has a unique stationary distribution, which we have denoted by κ_Y, and is given by

$$\kappa_Y(s_0, a_0, ..., s_n, a_n) = \kappa_S(s_0) \left(\prod_{i=0}^{n-1} \pi(a_i|s_i) P_{a_i}(s_i, s_{i+1}) \right) \pi(a_n|s_n).$$

Now use the definition of total variation distance, and we have for any $y = (s_0, a_0, ..., s_n, a_n)$ and $k \geq 0$:

$$\|P^{k+n+1}((s_0, a_0, ..., s_n, a_n), \cdot) - \kappa_Y(\cdot)\|_{\text{TV}}$$

$$= \frac{1}{2} \sum_{s_0', a_0', ..., s_n', a_n'} \left| \sum_s P_{a_n}(s_n, s) P_{\pi_b}^k(s, s_0') - \kappa_S(s_0') \right| \left(\prod_{i=0}^{n-1} \pi(a_i'|s_i') P_{a_i'}(s_i', s_{i+1}') \right) \pi(a_n'|s_n')$$

$$= \frac{1}{2} \sum_{s_0'} \left| \sum_s P_{a_n}(s_n, s) P_{\pi_b}^k(s, s_0') - \kappa_S(s_0') \right|$$

$$\leq \frac{1}{2} \sum_{s_0'} \sum_s P_{a_n}(s_n, s) \left| P_{\pi_b}^k(s, s_0') - \kappa_S(s_0') \right|$$

$$= \frac{1}{2} \sum_s P_{a_n}(s_n, s) \sum_{s_0'} \left| P_{\pi_b}^k(s, s_0') - \kappa_S(s_0') \right|$$

$$\leq \frac{1}{2} \sum_s P_{a_n}(s_n, s) \max_{s'} \sum_{s_0'} \left| P_{\pi_b}^k(s', s_0') - \kappa_S(s_0') \right|$$

$$= \max_{s \in \mathcal{S}} \|P_{\pi_b}^k(s, \cdot) - \kappa_S(\cdot)\|_{\text{TV}}$$

$$\leq C\sigma^k.$$

(3) It is clear that $\mathbb{E}_{Y \sim \mathcal{K}_Y}[F(Q, Y)] = \mathcal{K}_{SA} \mathcal{T}_c(\mathcal{H}_\rho(Q) - Q) + Q$, which by definition is equal to $\tilde{\mathcal{B}}_{c,\rho}(Q)$.

B Connection to Linear SA Involving a Hurwitz Matrix

In view of Proposition 2.2, Algorithm 1 can be alternatively interpreted as a linear SA algorithm for solving the equation $(A - I)Q + b = 0$. In the case where the matrix A is substochastic, our results imply finite-sample bounds for such linear SA algorithm, which is stated in the following.

Let $\{Y_k\}$ be a Markov chain with finite state-space $\mathcal{Y}$ and unique stationary distribution κ_Y. Let $\tilde{A} : \mathcal{Y} \mapsto \mathbb{R}^{d \times d}$ be a matrix valued function and let $\tilde{b} : \mathcal{Y} \mapsto \mathbb{R}^d$ be a vector valued function. Let $\bar{A} = \mathbb{E}_{Y \sim \kappa_Y}[\tilde{A}(Y)]$ and $\bar{b} = \mathbb{E}_{Y \sim \kappa_Y}[\tilde{b}(Y)]$. Consider the following linear SA algorithm:

$$x_{k+1} = x_k + \alpha((\tilde{A}(Y_k) - I)x_k + \tilde{b}(Y_k)), \tag{8}$$

where α is the constant stepsize. Then, we have the following result.

Theorem B.1. *Consider $\{x_k\}$ generated by Algorithm* (8). *Suppose that*

(1) The Markov chain $\{Y_k\}$ has a unique stationary distribution κ_Y, and $\max_{y \in \mathcal{Y}} \|P^k(y, \cdot) - \kappa_Y(\cdot)\|_{TV} \leq C'\sigma'^k$ for all $k \geq 0$, where $C' > 0$ and $\sigma' \in (0, 1)$ are constants.

(2) There exist $A_{\max}, b_{\max} > 0$ such that $\|\tilde{A}(y)\|_\infty \leq A_{\max}$ and $\|\tilde{b}(y)\|_\infty \leq b_{\max}$ for all $y \in \mathcal{Y}$.

(3) The matrix $\bar{A}$ is a sub-stochastic matrix with modulus $\omega' \in (0, 1)$.

Then, for any $\theta \in (0, 1)$, when α is chosen such that $\alpha t_\alpha(\mathcal{MC}_Y) \leq \frac{\theta \omega' \mu_{\min}^{2/p}}{228 p (A_{\max} + 1)^2}$, there exists a weight vector $\mu \in \Delta^d$ satisfying $\mu_{\min} \geq \frac{\omega'(1-\theta)}{(1-\theta\omega')d}$ such that we have for all $k \geq t_\alpha(\mathcal{MC}_Y)$:

$$\mathbb{E}[\|x_k - x^*\|_{\mu,p}^2] \leq \tilde{c}_1 (1 - \theta\omega'\alpha)^{k - t_\alpha(\mathcal{MC}_Y)} + \frac{228 p \tilde{c}_2 (A_{\max} + 1)^2}{\mu_{\min}^{2/p} \theta\omega'} \alpha t_\alpha(\mathcal{MC}_Y),$$

20

where $\tilde{c}_1 = (\|x_0 - x^*\|_{\mu,p} + \|x_0\|_{\mu,p} + \frac{b_{\max}\mu_{\min}^{1/p}}{A_{\max}+1})^2$ and $\tilde{c}_2 = (\|x^*\|_{\mu,p} + \frac{b_{\max}\mu_{\min}^{1/p}}{A_{\max}+1})^2$.

Similarly, Theorem B.1 has the following two corollaries, where we provide finite-sample bounds on $\mathbb{E}[\|x_k - x^*\|_p^2]$ and $\mathbb{E}[\|x_k - x^*\|_\infty^2]$.

Corollary B.1.1. *Under the same assumptions as Theorem B.1, we have for all $k \geq t_\alpha(\mathcal{MC}_Y)$:*

$$\mathbb{E}[\|x_k - x^*\|_p^2] \leq \frac{\tilde{c}_1}{\mu_{\min}^{2/p}}(1 - \theta\omega'\alpha)^{k-t_\alpha(\mathcal{MC}_Y)} + \frac{228p\tilde{c}_2(A_{\max}+1)^2}{\mu_{\min}^{4/p}\theta\omega'}\alpha t_\alpha(\mathcal{MC}_Y).$$

Corollary B.1.2. *Under the same assumptions as Theorem B.1, we have for all $k \geq t_\alpha(\mathcal{MC}_Y)$:*

$$\mathbb{E}[\|x_k - x^*\|_\infty^2] \leq \tilde{c}_1\sqrt{e}\left(1 - \frac{\omega'\alpha}{2}\right)^{k-t_\alpha(\mathcal{MC}_Y)} + \frac{1824e\log(2d/\omega')\tilde{c}_2(A_{\max}+1)^2}{\omega'}\alpha t_\alpha(\mathcal{MC}_Y).$$

An alternative approach of studying linear SA algorithm with Markovian noise was provided in [29], which established convergence bounds for linear stochastic approximation when the matrix $\bar{A} - I$ is Hurwitz. The bounds in [29] are in terms of solution of the Lyapunov equation:

$$(\bar{A} - I)^\top \Sigma + \Sigma(\bar{A} - I) + I = 0. \tag{9}$$

In particular, the finite-sample bounds depend on the ratio between maximum eigenvalue $\lambda_{\max}$ and minimum eigenvalue $\lambda_{\min}$ of the solution Σ. In general, it is not clear how this ratio can be evaluated and it is unknown. In the context of off-policy TD algorithms, the dependence on the contraction factor, the variance and the sizes of state-action spaces are hidden in this ratio, and so the trade-offs that we presented in Section 2 are not evident.

In our approach, we overcome this challenge by finding a Σ that solves the inequality,

$$(\bar{A} - I)^\top \Sigma + \Sigma(\bar{A} - I) + \eta\Sigma \preceq^2 0 \tag{10}$$

instead of the Lyapunov equation. Here, $\eta > 0$ is a constant. It turns out that this is sufficient to obtain finite-sample bounds. We find such a Σ by essentially establishing the weighted ℓ_p-norm contraction property, in particular, the weighted ℓ_2-norm contraction property. To see this, observe that when the matrix $\bar{A}$ is a substochastic matrix with a positive modulus $\omega' \in (0, 1)$, Theorem 2.1 implies that $\bar{A}^\top N\bar{A} \preceq (1 - \omega')N$, where $N = \text{diag}(\mu)$. Therefore, we have

$$(1 - \omega')N \succeq (\bar{A} - I + I)^\top N(\bar{A} - I + I) \succeq (\bar{A} - I)^\top N + N(\bar{A} - I) + N,$$

which implies $(\bar{A} - I)^\top N + N(\bar{A} - I) + \omega'N \preceq 0$. Thus, $\Sigma = N$ satisfies (10) with $\eta = \omega'$. When compared to the approach in [29], we trade-off the unknown eigenvalues of the solution to the Lyapunov equation for the weight vector μ. Since we are able to establish a lower bound on μ, we can obtain a sample complexity bound that doesn't involve any unknowns (except $\mathcal{K}_{SA,\min}$, which is inevitable in both approaches). As a result, we are able to fully characterize the impact of the generalized importance sampling ratios in Corollary 2.2.1, and provide insights about the bias-variance trade-offs in multi-step off-policy TD-learning algorithms.

In this section, we present finite-sample bounds for linear stochastic approximation involving a substochastic matrix, whereas [29] considers Hurwitz matrices. However, these results are equivalent because of the following lemma.

Lemma B.1. *(1) Suppose $M \in \mathbb{R}^{d \times d}$ is a substochastic matrix with a positive modulus, then the matrix M' defined by $M' = M - I$ is Hurwitz.*

(2) Suppose $M' \in \mathbb{R}^{d \times d}$ is a Hurwitz matrix, then there exists $\phi \in (0, 1)$ such that the matrix $M = \phi M' + I$ is a contraction mapping with respect to a norm induced by an inner product.

Proof. The proof of Part (1) is straight-forward and we skip it. Part (2) is also not challenging, and we present an overview of the argument. When M' is Hurwitz, all its eigenvalues are located on the open left half of the complex plane. Therefore, there exists $\phi \in (0, 1)$ such that the eigenvalues of M' are within the unit ball centered at $(-1, 0)$ of the complex plane. It follows that $M = \phi M' + I$ has eigenvalues located inside the unit ball centered at the origin of the complex plane. This implies the desired contraction property. □

C   Technical Details in Section 3

C.1   Proof of Theorem 3.1

Since Vanilla IS is a special case of Algorithm 1, one can directly apply Theorem 2.2 to obtain the finite-sample bound. However, there is one special property of Vanilla IS we can exploit to obtain a tighter finite-sample bound. In particular, consider Proposition A.1 (1) (a). In the case of Vanilla IS, the corresponding Lispchitz constant is $\frac{2}{\mu_{\min}^{1/p}} f(\gamma r_{\max})(\gamma r_{\max} + 1)$. We next show that due to $c(s,a) = \rho(s,a)$ in Vanilla IS, we can use telescoping to improve the Lipschitz constant. Specifically, in Vanilla IS, for any $Q \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, $y \in \mathcal{Y}$, and (s,a), we have

$$[F(Q,y)](s,a)$$
$$= \mathbb{I}_{\{(s_0,a_0)=(s,a)\}} \sum_{i=0}^{n-1} \gamma^i \prod_{j=1}^{i} c(s_j,a_j)(\mathcal{R}(s_i,a_i) + \gamma c(s_{i+1},a_{i+1})Q(s_{i+1},a_{i+1}) - Q(s_i,a_i)) + Q(s,a)$$
$$= \mathbb{I}_{\{(s_0,a_0)=(s,a)\}} \sum_{i=0}^{n-1} \gamma^i \prod_{j=1}^{i} c(s_j,a_j)\mathcal{R}(s_i,a_i) + \mathbb{I}_{\{(s_0,a_0)=(s,a)\}} \sum_{i=0}^{n-1} \gamma^{i+1} \prod_{j=1}^{i+1} c(s_j,a_j)Q(s_{i+1},a_{i+1})$$
$$\quad - \mathbb{I}_{\{(s_0,a_0)=(s,a)\}} \sum_{i=0}^{n-1} \gamma^i \prod_{j=1}^{i} c(s_j,a_j)Q(s_i,a_i) + Q(s,a)$$
$$= \mathbb{I}_{\{(s_0,a_0)=(s,a)\}} \sum_{i=0}^{n-1} \gamma^i \prod_{j=1}^{i} c(s_j,a_j)\mathcal{R}(s_i,a_i) + \mathbb{I}_{\{(s_0,a_0)=(s,a)\}} \sum_{i=1}^{n} \gamma^i \prod_{j=1}^{i} c(s_j,a_j)Q(s_i,a_i)$$
$$\quad - \mathbb{I}_{\{(s_0,a_0)=(s,a)\}} \sum_{i=0}^{n-1} \gamma^i \prod_{j=1}^{i} c(s_j,a_j)Q(s_i,a_i) + Q(s,a)$$
$$= \mathbb{I}_{\{(s_0,a_0)=(s,a)\}} \sum_{i=0}^{n-1} \gamma^i \prod_{j=1}^{i} c(s_j,a_j)\mathcal{R}(s_i,a_i) + \mathbb{I}_{\{(s_0,a_0)=(s,a)\}}\gamma^n \prod_{j=1}^{n} c(s_j,a_j)Q(s_n,a_n)$$
$$\quad + (1 - \mathbb{I}_{\{(s_0,a_0)=(s,a)\}})Q(s,a).$$

Therefore, we have for any $Q_1, Q_2 \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, and $y \in \mathcal{Y}$:

$$\|F(Q_1,y) - F(Q_2,y)\|_{\mu,p}$$
$$\leq \left[\sum_{s,a} \mu(s,a) \left| \mathbb{I}_{\{(s_0,a_0)=(s,a)\}}\gamma^n \prod_{j=1}^{n} c(s_j,a_j)(Q_1(s_n,a_n) - Q_2(s_n,a_n)) \right|^p \right]^{1/p} + \|Q_1 - Q_2\|_{\mu,p}$$
$$\leq \left[\sum_{s,a} \mu(s,a) |(\gamma r_{\max})^n \|Q_1 - Q_2\|_\infty|^p \right]^{1/p} + \|Q_1 - Q_2\|_{\mu,p}$$
$$\leq (\gamma r_{\max})^n \|Q_1 - Q_2\|_\infty + \|Q_1 - Q_2\|_{\mu,p}$$
$$\leq \frac{(\gamma r_{\max})^n + 1}{\mu_{\min}^{1/p}} \|Q_1 - Q_2\|_{\mu,p}.$$

Using this improved Lipschitz constant and we obtain Theorem 3.1, where the rest of the proof is identical to that of Theorem 2.2.

C.2   Comparison to the n-Step TD-Learning Results in [10]

The sample complexity of on-policy n-step TD-learning provided in [10, Corollary 3.3.1.] is

$$\tilde{\mathcal{O}}\left(\frac{n \log^2(1/\epsilon)}{\epsilon^2 \mathcal{K}_{S,\min}^2 (1-\gamma)^2 (1-\gamma^n)^2} \right) \tilde{\mathcal{O}}(|\mathcal{S}|^{1/2}). \tag{11}$$

In this work, by setting $\pi_b = \pi$, Theorem 3.1 implies a sample complexity of

$$\tilde{\mathcal{O}}\left(\frac{n\log^2(1/\epsilon)}{\epsilon^2\mathcal{K}_{SA,\min}^2(1-\gamma^n)^2(1-\gamma)^2}\right). \tag{12}$$

These two results have the same dependency on ϵ, n, and $1/(1-\gamma)$, but have two differences. First is that in Eq. (11) there is $\mathcal{K}_{S,\min}^{-2}$ while we have $\mathcal{K}_{SA,\min}^{-2}$. This is because we are evaluating the Q-function while [10] studies policy evaluation for the V-function. Another difference is that in Eq. (11) there is an additional factor of $|\mathcal{S}|^{1/2}$. This is because we find the sample complexity to obtain $\mathbb{E}[\|\cdot\|_\infty] \leq \epsilon$ while [10] finds the sample complexity to achieve $\mathbb{E}[\|\cdot\|_2] \leq \epsilon$.

C.3 Proof of Theorems 3.2 to 3.5

The results are obtained by directly applying Theorem 2.2.

C.4 Computing the Sample Complexity of Q-Trace from [19]

To compute the sample complexity of the Q-trace algorithm from [19], we will adopt the notation from this paper for consistency. In view of [19, Theorem 2.1], to obtain $\mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_\infty] \leq \epsilon$, we need

$$\alpha \sim \mathcal{O}\left(\frac{\epsilon^2}{\log(1/\epsilon)}\right)\tilde{\mathcal{O}}\left(\frac{\mathcal{K}_{SA,\min}^2 f(\gamma D_{c,\min})^2(1-\gamma D_{\rho,\max})^4}{nf(\gamma\bar{c})^2(\gamma\bar{\rho}+1)^2}\right),$$

which implies

$$k \sim \mathcal{O}\left(\frac{\log(1/\epsilon)^2}{\epsilon^2}\right)\tilde{\mathcal{O}}\left(\frac{nf(\gamma\bar{c})^2(\gamma\bar{\rho}+1)^2}{\mathcal{K}_{SA,\min}^3 f(\gamma D_{c,\min})^3(1-\gamma D_{\rho,\max})^5}\right)$$

$$= \tilde{\mathcal{O}}\left(\frac{\log^2(1/\epsilon)nf(\gamma\bar{c})^2(\gamma\bar{\rho}+1)^2}{\epsilon^2\omega^3(1-\gamma D_{\rho,\max})^2}\right).$$