


Crosscutting Areas

A Lyapunov Theory for Finite-Sample Guarantees of Markovian Stochastic Approximation

Zaiwei Chen,^{a,*} Siva T. Maguluri,^a Sanjay Shakkottai,^b Karthikeyan Shanmugam^c

^aThe School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332; ^bDepartment of Electrical and Computer Engineering, The University of Texas at Austin, Austin, Texas 78712; ^cIBM Research AI Group, Yorktown Heights, New York 10598

*Corresponding author

Contact: zchen458@gatech.edu,  <https://orcid.org/0000-0001-9915-5595> (ZC); siva.theja@gatech.edu,  <https://orcid.org/0000-0002-5797-1639> (STM); sanjay.shakkottai@utexas.edu,  <https://orcid.org/0000-0002-4325-9050> (SS); KarthikeyanShanmugam88@gmail.com,  <https://orcid.org/0009-0008-2879-5868> (KS)

Received: May 16, 2022

Revised: November 1, 2022; May 21, 2023

Accepted: August 28, 2023

Published Online in Articles in Advance:
October 6, 2023

Area of Review: Machine Learning and Data
Science

<https://doi.org/10.1287/opre.2022.0249>

Copyright: © 2023 INFORMS

Abstract. This paper develops a unified Lyapunov framework for finite-sample analysis of a Markovian stochastic approximation (SA) algorithm under a contraction operator with respect to an arbitrary norm. The main novelty lies in the construction of a valid Lyapunov function called the *generalized Moreau envelope*. The smoothness and an approximation property of the generalized Moreau envelope enable us to derive a one-step Lyapunov drift inequality, which is the key to establishing the finite-sample bounds. Our SA result has wide applications, especially in the context of reinforcement learning (RL). Specifically, we show that a large class of value-based RL algorithms can be modeled in the exact form of our Markovian SA algorithm. Therefore, our SA results immediately imply finite-sample guarantees for popular RL algorithms such as n -step temporal difference (TD) learning, $\text{TD}(\lambda)$, off-policy V -trace, and Q -learning. As byproducts, by analyzing the convergence bounds of n -step TD and $\text{TD}(\lambda)$, we provide theoretical insight into the problem about the efficiency of bootstrapping. Moreover, our finite-sample bounds of off-policy V -trace explicitly capture the tradeoff between the variance of the stochastic iterates and the bias in the limit.

Funding: This work was supported by RTX, the National Science Foundation [Grants 2019844, 2107037, 211247, 2112533, 2144316, and 2240982], and the Machine Learning Laboratory at University of Texas at Austin.

Supplemental Material: The online appendix is available at <https://doi.org/10.1287/opre.2022.0249>.

Keywords: Markovian stochastic approximation • finite-sample analysis • Lyapunov drift method • generalized Moreau envelope • reinforcement learning

1. Introduction

In operations research, often the problem of interest is reduced to a root-finding problem for a properly defined equation. For example, solving $\min_{x \in \mathbb{R}^d} J(x)$ for a differentiable objective function $J(\cdot)$ is closely related to solving the equation $\nabla J(x) = 0$. In a Markov decision process (MDP) or its environment-agnostic variant reinforcement learning (RL) problem, essentially the problem is to solve a fixed-point equation known as the Bellman equation.

A popular approach for solving such root-finding problems is through iterative algorithms, with the popular gradient descent/ascent algorithm being a typical example thereof. However, sometimes we do not have enough information or enough computational power to carry out the desired iterative algorithm and have to work with its noise-corrupted variant. More generally, an iterative algorithm in the presence of noise is called a stochastic approximation (SA) algorithm (Robbins and

Monro 1951), which is the underlying workhorse for solving large-scale optimization and machine learning problems (Lan 2020).

The SA method is used at scale in the context of RL (Sutton and Barto 2018). Because the environmental model is unknown, classical iterative algorithms for solving MDPs, such as value iteration and policy iteration, are not directly implementable. Therefore, people develop data-driven algorithms such as Q -learning (Watkins and Dayan 1992) and actor-critic (Konda and Tsitsiklis 1999) for solving the RL problem, which are essentially SA algorithms. To guide practical implementations, for a certain SA algorithm, we naturally want to have an understanding on how many iterations are needed to achieve a certain level of accuracy. This motivates us to derive performance guarantees of SA algorithms with a finite number of iterations, which is called the finite-sample analysis.

Motivated by applications in RL, in this work, we focus on finite-sample analysis of an SA algorithm involving a contractive operator and under Markovian sampling. More formally, consider an SA algorithm of the form

$$x_{k+1} = x_k + \alpha_k(F(x_k, Y_k) - x_k + w_k), \quad (1)$$

where $\{\alpha_k\}$ is a sequence of step sizes, $\{Y_k\}$ is a Markov chain with a finite state space \mathcal{Y} and a unique stationary distribution μ_Y , $F: \mathbb{R}^d \times \mathcal{Y} \mapsto \mathbb{R}^d$ is a (possibly non-linear) operator, and $\{w_k\}$ is a random process representing the additive extraneous noise. Let $\bar{F}(\cdot) = \mathbb{E}_{Y \sim \mu_Y}[F(\cdot, Y)]$. We assume that the operator $\bar{F}(\cdot)$ is a *contraction mapping* with respect to some arbitrary norm $\|\cdot\|_c$, which implies that the fixed-point equation $\bar{F}(x) = x$ has a unique solution $x^* \in \mathbb{R}^d$. In view of Algorithm (1), it can be interpreted as an SA algorithm for solving the fixed-point equation $\bar{F}(x) = x$. In finite-sample analysis, our goal is to understand how the mean-square error $\mathbb{E}[\|x_k - x^*\|_c^2]$ decays as a function of the iteration number k .

1.1. Main Contributions

The main contributions of this work are summarized in the following.

1.1.1. Finite-Sample Analysis for Markovian SA. We establish finite-sample guarantees (with various choices of step sizes) of Algorithm (1). Specifically, we show that when using a constant step size, that is, $\alpha_k \equiv \alpha$, the convergence rate is geometric, with an asymptotic accuracy of the order $\mathcal{O}(\alpha \log(1/\alpha))$. When using diminishing step sizes of the form $\alpha/(k+h)^\xi$ (where $\xi \in (0, 1]$), the convergence rate is of the order $\mathcal{O}(\log(k)/k^\xi)$, provided that α and h are appropriately chosen. Furthermore, our bound also involves a (possibly dimension dependent) constant that is determined by the contraction norm $\|\cdot\|_c$. In the special case of ℓ_∞ -norm contraction, we show that such a constant scales only logarithmically in terms of the dimension d , which is not improvable in general. Our SA results rely on a novel construction of a Lyapunov function called the generalized Moreau envelope and controlling the stochastic error due to the Markovian noise.

1.1.2. Finite-Sample Analysis of RL Algorithms. Our SA results enable us to establish finite-sample bounds of a variety of value-based RL algorithms (including various temporal difference (TD) learning algorithms and Q-learning) in one shot. Specifically, for TD learning with on-policy sampling, we establish finite-sample guarantees the popular n -step TD and TD(λ). For these two families of algorithms, there is an important question about the efficiency of bootstrapping (Sutton 1999),

which refers to the question of how to choose the parameter n in n -step TD (or λ in TD(λ)) so that n -step TD (or TD(λ)) achieves its best performance. Our finite-sample analysis sheds light on this problem by explicitly capturing the dependence of the convergence bounds on the tunable parameters of interest (i.e., n in n -step TD or λ in TD(λ)). For example, in n -step TD, we show that the parameter n appears as $n/(1-\gamma)^2$ in the sample complexity bound, which leads to an estimate of the optimal choice of n as $n_{\text{opt}} = \mathcal{O}(1/\log(1/\gamma))$.

For TD learning with off-policy sampling, we establish for the *first* time the finite-sample bound of the off-policy V -trace algorithm (Espenholt et al. 2018), which is used at scale in the Google's city navigation project called *Street Learn* (Mirowski et al. 2018). The V -trace algorithm can be viewed as an off-policy variant of the n -step TD-learning algorithm, where the key is to truncate the importance sampling factors using two different truncation levels \bar{c} and $\bar{\rho}$ to separately control the variance in the stochastic iterates and the bias in the limit. Therefore, theoretically understanding the tradeoffs between the aforementioned variance and bias is of vital importance for the implementation of the V -trace algorithm. Our finite-sample analysis provides theoretical insights into such a bias-variance tradeoff.

Last, for the Q-learning algorithm, our finite-sample bound implies a sample complexity of $\tilde{\mathcal{O}}(\epsilon^{-2}(1-\gamma)^{-5} \mathcal{K}_{SA, \min}^{-3})$, where ϵ is the desired accuracy, γ is the discount factor, and $\mathcal{K}_{SA, \min}$ is the minimal component of the stationary distribution of the Markov chain induced by the behavior policy. See Section 3.4 for a detailed discussion about our results on Q-learning.

1.2. Summary of Our Technical Approach

In this section, we first provide a high-level overview of our Lyapunov approach for the finite-sample analysis of Algorithm (1). Then, we use the popular Q-learning algorithm as an example to elaborate on our blueprint for applying the SA results to RL algorithms to obtain sample complexity guarantees.

1.2.1. Analysis of Markovian SA: Motivation of a Smooth Lyapunov Function. We begin by rewriting Algorithm (1) as

$$\begin{aligned} x_{k+1} - x_k &= \underbrace{\alpha_k(\bar{F}(x_k) - x_k)}_{\text{Expected Update}} + \underbrace{\alpha_k(F(x_k, Y_k) - \bar{F}(x_k))}_{\text{Markovian Noise}} \\ &+ \underbrace{\alpha_k w_k}_{\text{Additive Noise}}. \end{aligned} \quad (2)$$

To provide intuition, we assume for now that the norm $\|\cdot\|_c$ with respect to which $\bar{F}(\cdot)$ being a contraction is the ℓ_p -norm for some $p \in [2, \infty)$, that is, $\|\bar{F}(x) - \bar{F}(y)\|_p \leq \beta \|x - y\|_p$ for all $x, y \in \mathbb{R}^d$, where $\beta \in (0, 1)$ is the contraction factor.

Consider the ordinary differential equation (ODE) associated with the SA algorithm: $\dot{x}(t) = \bar{F}(x(t)) - x(t)$. Intuitively, the ODE can be viewed as a continuous and deterministic counterpart of SA Algorithm (2). It was shown in Borkar (2009), chapter 10, that the function $W(x) = \|x - x^*\|_p$ satisfies $\frac{d}{dt} W(x(t)) \leq -\kappa W(x(t))$ for some $\kappa > 0$. This implies that the solution $x(t)$ of the ODE converges to its unique equilibrium point x^* geometrically fast, which further implies the asymptotic convergence of the SA algorithm via the ODE approach (Ljung 1977, Borkar 2009). The coefficient κ corresponds to a *negative drift*.

Although the ODE approach gives asymptotic convergence, it does not provide finite-sample guarantees. To obtain finite-sample bounds, in this paper we study the SA directly and not the ODE. Then, the Lyapunov function $W(x)$ cannot be used directly to analyze the SA algorithm due to the discretization error and stochastic error (see Equation 2). Suppose that we can find a function $M(x)$ that gives negative drift, and, in addition, $M(x)$ is L -smooth with respect to some norm $\|\cdot\|_s$. Then, we have a handle to deal with the discretization error and the stochastic error to obtain:

$$\begin{aligned} & \mathbb{E}[M(x_{k+1} - x^*)] \\ & \leq (1 - \mathcal{O}(\alpha_k) + o(\alpha_k))\mathbb{E}[M(x_k - x^*)] + o(\alpha_k), \end{aligned} \quad (3)$$

which implies a contraction in $\mathbb{E}[M(x_{k+1} - x^*)]$. Therefore, a finite-sample bound can be obtained by recursively applying the previous inequality. The key point is that $M(x)$'s smoothness and its negative drift with respect to the ODE produces a contraction $(1 - \mathcal{O}(\alpha_k) + o(\alpha_k))$ for $\{x_k\}$. Based on the previous analysis, we see that the Lyapunov function for the SA in the case of ℓ_p -norm contraction should be $M(x) = \frac{1}{2}\|x - x^*\|_p^2$, which is known to be a smooth function (Beck 2017).

Now consider the case where the contraction norm $\|\cdot\|_c$ is arbitrary. Because the function $f(x) = \frac{1}{2}\|x - x^*\|_c^2$ is not necessarily smooth, the key difficulty is to construct a smooth Lyapunov function that also has a negative drift. An important special case is when $\|\cdot\|_c = \|\cdot\|_\infty$, which is applicable to many RL algorithms as will be discussed later in Section 3. Previously, the lack of a suitable Lyapunov/potential function to study SA algorithms under $\|\cdot\|_\infty$ -contraction operators has been a fundamental open problem, as pointed out in the classical textbook (Bertsekas and Tsitsiklis 1996, section 4.3). According to Bertsekas and Tsitsiklis, "Unfortunately, it is unclear whether one can define a smooth potential function $M(\cdot)$ such that the update of any component of $J(\cdot)$ [referred to the objective function of RL] is along a descent direction with respect to $M(\cdot)$." We provide a solution to this problem by constructing a smoothed convex envelope $M(x)$ called the *generalized Moreau envelope* that is smooth with respect to some norm $\|\cdot\|_s$,

and is a tight approximation to $f(x)$ in the sense that $(1+a)M(x) \leq f(x) \leq (1+b)M(x)$ for some small enough constants $a, b > 0$. The approximation property ensures that $M(\cdot)$ is a valid Lyapunov function with a negative drift, and the smoothness property enables us to control the discretization error and the stochastic error in Algorithm (1). Together, they let us prove a convergence result similar to the case when $f(x)$ is smooth.

1.2.2. Applications to RL: Illustration via Q-Learning.

The Q-learning algorithm is a model-free recursive approach to find the optimal policy corresponding to an MDP (see Section 3.4 for details). At time step k , the algorithm updates a vector (of dimension state-space size \times action-space size) Q_k , which is an estimate of the optimal Q-function Q^* , using noisy samples collected along a single sample trajectory. After a sufficient number of iterations, the vector Q_k is a close approximation of Q^* , which (after some straightforward computations) delivers the optimal policy for the MDP. Concretely, let $\{(S_k, A_k)\}$ be a sample trajectory of state-action pairs collected by applying some behavior policy to the underlying MDP model. The Q-learning algorithm performs a scalar update of a (vector-valued) iterate Q_k according to

$$\begin{aligned} & Q_{k+1}(s, a) \\ & = \begin{cases} Q_k(s, a) + \alpha_k \left(\mathcal{R}(S_k, A_k) + \gamma \max_{a' \in \mathcal{A}} Q_k(S_{k+1}, a') \right. \\ \quad \left. - Q_k(S_k, A_k) \right), & (s, a) = (S_k, A_k), \\ Q_k(s, a), & (s, a) \neq (S_k, A_k). \end{cases} \end{aligned} \quad (4)$$

At a high level, this recursion approximates the fixed-point of the Bellman equation through samples along a single trajectory. There are, however, two sources of noise in this approximation: (1) *asynchronous update* where only one of the components in the vector Q_k is updated (component corresponding to the state-action pair (S_k, A_k) encountered at time k), and other components in the vector Q_k are left unchanged, and (2) *stochastic noise* due to the expectation in the Bellman operator being replaced by a single sample estimate.

To apply our SA results, the first step is to reformulate Q-learning in the form of Algorithm (1). Let $F : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \times \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ be an operator defined as $[F(Q, s_0, a_0, s_1)](s, a) = \mathbb{1}_{\{(s_0, a_0) = (s, a)\}} (\mathcal{R}(s_0, a_0) + \gamma \max_{a' \in \mathcal{A}} Q(s_1, a') - Q(s_0, a_0)) + Q(s, a)$ for all (s, a) . Then Q-Learning Algorithm (4) can be rewritten as

$$Q_{k+1} = Q_k + \alpha_k (F(Q_k, S_k, A_k, S_{k+1}) - Q_k), \quad (5)$$

which is in the form of Algorithm (1) with x_k being Q_k , $w_k = 0$, and $Y_k = (S_k, A_k, S_{k+1})$. The key takeaway is that, in Equation (5), the various noise terms are encoded through introducing the operator $F(\cdot)$ and the associated evolution of the Markov chain $\{Y_k\}$.

After the SA reformulation, to apply our SA results, we need to establish the contraction property of the operator $\bar{F}(\cdot) := \mathbb{E}[F(\cdot, S_k, A_k, S_{k+1})]$ associated with the Q -learning algorithm, where the expectation is taken with respect to the stationary distribution of the Markov chain $\{(S_k, A_k, S_{k+1})\}$. Under mild conditions, we show that $\bar{F}(Q) = \mathcal{K}_{SA}\mathcal{H}(Q) + (I - \mathcal{K}_{SA})Q$. Here $\mathcal{H}(\cdot)$ is the Bellman operator for the Q -function (Bertsekas and Tsitsiklis 1996), and the matrix \mathcal{K}_{SA} is a diagonal matrix with $\{p(s, a)\}_{(s, a) \in S \times A}$ sitting on its diagonal, where $p(s, a)$ is the stationary visitation probability of the state-action pair (s, a) . An important insight about the operator $\bar{F}(\cdot)$ is that it can be viewed as an asynchronous variant of the Bellman operator $\mathcal{H}(\cdot)$. To see this, consider a state-action pair (s, a) . The value of $[\bar{F}(Q)](s, a)$ can be interpreted as the expectation of a random variable, which takes $[\mathcal{H}(Q)](s, a)$ with probability $p(s, a)$, and takes $Q(s, a)$ with probability $1 - p(s, a)$. This precisely captures the asynchronous update in Q -Learning Algorithm (4) in that, in steady state, $Q_k(s, a)$ is updated with probability $p(s, a)$ and remains unchanged otherwise. Moreover, because $\mathcal{H}(\cdot)$ is known to be a contraction mapping with respect to $\|\cdot\|_\infty$, we also show that $\bar{F}(\cdot)$ is a contraction mapping with respect to $\|\cdot\|_\infty$ (while the contraction factor is different), and the optimal Q -function is its unique fixed point.

The SA reformulation together with the contraction property enables us to apply our SA results to get the finite-sample bounds and the sample complexity guarantees of Q -learning. Beyond Q -learning, TD-learning variants such as off-policy V -trace, n -step TD, and TD(λ) can all be modeled as Markovian SA algorithms involving contraction mappings (possibly with respect to different norms) and Markovian noise. Therefore, our SA results provide a *unified* approach for the finite-sample analysis of value-based RL algorithms.

1.3. Related Literature

In this section, we discuss the literature on SA algorithms. We defer the discussion of the related literature on RL algorithms to the corresponding sections where we present the results.

The SA method was first introduced in Robbins and Monro (1951) to iteratively solve systems of equations. Since then, the SA method has been widely used in the context of optimization and machine learning. For example, in optimization, a special case of SA known as stochastic gradient descent (SGD) has been a popular approach for solving large-scale optimization problems (Bottou et al. 2018, Lan 2020). In RL, popular algorithms such as Q -learning and TD learning are essentially SA algorithms for solving variants of the Bellman equation (Bertsekas and Tsitsiklis 1996, Sutton and Barto 2018).

The early literature on SA focused on asymptotic convergence (Bertsekas and Tsitsiklis 1996, Kushner 2010,

Benveniste et al. 2012, Kushner and Clark 2012). A popular approach known as the ODE method (Ljung 1977) was developed to analyze the behavior of an SA algorithm by studying the stability of its associated ODE. See Borkar and Meyn (2000), Borkar (2009), Benaim (1996), Yaji and Bhatnagar (2019), and Karmakar and Bhatnagar (2021) for more details about the ODE approach. The asymptotic convergence of other variants of SA such as multiple time-scale SA was studied in Bhatnagar and Borkar (1998, 1997).

More recently, finite-sample analysis of SA algorithms has seen a lot of attention, as it provides more information than asymptotic convergence and can be used to guide practical implementations. For SA with a linear update rule, finite-sample analysis was performed in Bhandari et al. (2018), Srikant and Ying (2019), Dalal et al. (2018), and Thoppe and Borkar (2019). Other variants of linear SA, such as two-time-scale linear SA and decentralized linear SA, were studied in Kaledin et al. (2020) and Doan (2021) and Zeng et al. (2021), respectively. For SA with nonlinear update equations, finite-sample guarantees were derived under a contractive (or cone-contractive) operator in Wainwright (2019) and Qu and Wierman (2020) and under a strongly pseudo-monotone operator in Chen et al. (2022). Both Wainwright (2019) and Qu and Wierman (2020) require the noise to be almost surely bounded by a constant. In addition, the Markovian noise presented in Qu and Wierman (2020) has a special structure, whereas our Markovian noise is more general.

A special case of nonlinear SA is SGD, the finite-sample bounds of which were established in Lan (2020), Moulines and Bach (2011), Duchi et al. (2012), Doan (2023), and Bansal and Gupta (2019) and the references therein. In SGD, the property of the gradient operator plays an important role in the analysis. For general SA (like the one we study in this work), the update equation may not involve a gradient operator of any objective function. Consequently, constructing valid Lyapunov functions in this case is more challenging.

2. Finite-Sample Analysis of Markovian Stochastic Approximation

In this section, we present our main results. We begin by formally stating our assumptions.

Assumption 1 (Contraction). *The operator $\bar{F}(\cdot)$ satisfies $\|\bar{F}(x_1) - \bar{F}(x_2)\|_c \leq \beta \|x_1 - x_2\|_c$ for all $x_1, x_2 \in \mathbb{R}^d$, where $\beta \in (0, 1)$ and $\|\cdot\|_c$ is an arbitrary norm in \mathbb{R}^d .*

Under Assumption 1, the fixed-point equation $\bar{F}(x) = x$ has a unique solution, which we denoted by x^* (Banach 1922).

Assumption 2 (Lipschitz Continuity). *There exists $A_1 > 0$ such that $\|F(x_1, y) - F(x_2, y)\|_c \leq A_1 \|x_1 - x_2\|_c$ for any $x_1, x_2 \in \mathbb{R}^d$ and $y \in \mathcal{Y}$.*

We also denote $B_1 := \max_{y \in \mathcal{Y}} \|F(\mathbf{0}, y)\|_c$, which is well defined and finite, because the state space \mathcal{Y} is finite. The Lipschitz continuity assumption can be viewed as a relaxation of the assumption that the SA algorithm has a linear update rule. This assumption is naturally satisfied for all the RL algorithms we are going to study in Section 3.

Let $P_Y \in \mathbb{R}^{|\mathcal{Y}| \times |\mathcal{Y}|}$ be the transition probability matrix of the Markov chain $\{Y_k\}$, and let $\|\cdot\|_{TV}$ be the total variation distance between probability distributions.

Assumption 3. *The Markov chain $\{Y_k\}$ is irreducible and aperiodic.*

Because $|\mathcal{Y}| < \infty$, Assumption 3 implies that $\{Y_k\}$ has a unique stationary distribution μ_Y (Levin and Peres 2017). In addition, there exist $C > 0$ and $\sigma \in (0, 1)$ such that $\max_{y \in \mathcal{Y}} \|P_Y^k(y, \cdot) - \mu_Y(\cdot)\|_{TV} \leq C\sigma^k$ for all $k \geq 0$ (Levin and Peres 2017). Assumption 3 is imposed to control the stochastic error due to the Markovian noise $\{Y_k\}$ in Algorithm (1). In RL, Assumption 3 translates into a requirement of exploration, which, to some extent, is a necessary requirement for successfully learning an optimal policy.

Let \mathcal{F}_k be the σ -field generated by $\{(x_i, Y_i, w_i)\}_{0 \leq i \leq k-1} \cup \{x_k\}$.

Assumption 4 (Martingale Difference Noise). *The random process $\{w_k\}$ satisfies (1) $\mathbb{E}[w_k | \mathcal{F}_k] = 0$ for all $k \geq 0$, and (2) $\|w_k\|_c \leq A_2 \|x_k\|_c + B_2$ for all $k \geq 0$, where $A_2, B_2 > 0$.*

Unlike in Wainwright (2019) and Qu and Wierman (2020), the additive noise here w_k can grow linearly with respect to the latest iterate x_k and does not need to be uniformly bounded by an absolute constant.

2.1. Generalized Moreau Envelope as a Smooth Lyapunov Function

From now on, we will present our Lyapunov approach for the finite-sample analysis of Algorithm (1). Recall from Equation (3) that, with respect to the iterates $\{x_k\}$ of Algorithm (1), an ideal Lyapunov function $M(x)$ acts as a potential function that contracts. In this section, we construct a novel Lyapunov function through the generalized Moreau envelope.

The following definitions are needed. In this paper, $\langle x, y \rangle = x^\top y$ represents the standard dot product, whereas the norm $\|\cdot\|$ in the following definition can be any arbitrary norm instead of just being the Euclidean norm $\|x\|_2 = \langle x, x \rangle^{1/2}$.

Definition 1. Let $g: \mathbb{R}^d \rightarrow \mathbb{R}$ be a convex differentiable function. Then $g(\cdot)$ is said to be L smooth with respect to $\|\cdot\|$ if and only if $g(y) \leq g(x) + \langle \nabla g(x), y - x \rangle + \frac{L}{2} \|x - y\|^2$ for all $x, y \in \mathbb{R}^d$.

Definition 2 (Generalized Moreau Envelope). Let $h_1: \mathbb{R}^d \mapsto \mathbb{R}$ be a closed and convex function, and let $h_2: \mathbb{R}^d \mapsto \mathbb{R}$ be a convex and L smooth function. For any $\theta > 0$, the generalized Moreau envelope of $h_1(\cdot)$ with respect to $h_2(\cdot)$ is defined as $M_{h_1}^{\theta, h_2}(x) = \inf_{u \in \mathbb{R}^d} \{h_1(u) + \frac{1}{\theta} h_2(x - u)\}$.

The standard Moreau envelope was previously used in Guzmán and Nemirovski (2015) and Beck and Teboulle (2012) to study convex optimization problems. For any two functions $h_1, h_2: \mathbb{R}^d \mapsto \mathbb{R}$, the function defined by $(h_1 \square h_2)(x) := \inf_{u \in \mathbb{R}^d} \{h_1(u) + h_2(x - u)\}$ is called the infimal convolution of $h_1(\cdot)$ and $h_2(\cdot)$ (Beck 2017). Therefore, the generalized Moreau envelope in Definition 2 can be equivalently written as $M_{h_1}^{\theta, h_2}(x) = \left(h_1 \square \frac{h_2}{\theta}\right)(x)$.

2.1.1. Construction of a Valid Lyapunov Function. Let $f(x) = \frac{1}{2} \|x\|_c^2$, where $\|\cdot\|_c$ is the contraction norm given in Assumption 1. Let $\|\cdot\|_s$ be an arbitrary norm in \mathbb{R}^d such that $g(x) := \frac{1}{2} \|x\|_s^2$ is L smooth with respect to the same norm $\|\cdot\|_s$ in its definition. For example, $\|\cdot\|_s$ can be the ℓ_p -norm for any $p \in [2, \infty)$, where $L = p - 1$ (Beck 2017, example 5.11). Because of the equivalence between norms in \mathbb{R}^d (Lax 1997), there exist $\ell_{cs} \in (0, 1]$ and $u_{cs} \in [1, \infty)$ that depend only on the dimension d and universal constants, such that $\ell_{cs} \|\cdot\|_s \leq \|\cdot\|_c \leq u_{cs} \|\cdot\|_s$. We will use the generalized Moreau envelope of $f(\cdot)$ with respect to $g(\cdot)$, that is, $M_f^{\theta, g}(\cdot)$, as our Lyapunov function to analyze the behavior of Algorithm (1), where $\theta > 0$ is a tunable parameter.

The following proposition states that $M_f^{\theta, g}(\cdot)$ is a smooth approximation of the norm-squared function $f(\cdot)$.

Proposition 1 (Proof in Online Appendix 1.1). *The function $M_f^{\theta, g}(\cdot)$ has the following properties: (1) $M_f^{\theta, g}(\cdot)$ is convex, and $\frac{L}{\theta}$ smooth with respect to $\|\cdot\|_s$, (2) there exists a norm $\|\cdot\|_m$ such that $M_f^{\theta, g}(x) = \frac{1}{2} \|x\|_m^2$, and (3) it holds that $\ell_{cm} \|\cdot\|_m \leq \|\cdot\|_c \leq u_{cm} \|\cdot\|_m$, where $\ell_{cm} = (1 + \theta \ell_{cs}^2)^{1/2}$ and $u_{cm} = (1 + \theta u_{cs}^2)^{1/2}$.*

Proposition 1(1) is restated from Beck (2017), and we include it here for completeness. This, together with Proposition 1(3), implies that $M_f^{\theta, g}(\cdot)$ is a smooth approximation of the norm-squared function $f(\cdot)$. Proposition 1(2) states that $M_f^{\theta, g}(\cdot)$ itself is also a norm-squared function.

2.2. Establishing a One-Step Contractive Inequality

Using the smooth approximation property of the generalized Moreau envelope $M_f^{\theta, g}(\cdot)$, we next establish a

one-step contractive inequality (Equation 3) of $M_f^{\theta, g}$ ($x_k - x^*$). To state the result, we first introduce necessary notation and specify the condition needed for choosing the step sizes.

Let $\varphi_1 = \frac{1+\theta u_{cs}^2}{1+\theta l_{cs}^2}$, $\varphi_2 = 1 - \beta\varphi_1^{1/2}$, and $\varphi_3 = \frac{114L(1+\theta u_{cs}^2)}{\theta l_{cs}^2}$.

The tunable parameter θ is chosen such that $\varphi_2 > 0$, which is always possible because $\lim_{\theta \rightarrow 0} \varphi_1 = 1$ and $\beta \in (0, 1)$. For any $\delta > 0$, let $t_\delta = \min\{k \geq 0 : \max_{y \in \mathcal{Y}} \|P_Y^k(y, \cdot) - \mu_Y(\cdot)\|_{TV} \leq \delta\}$, which can be interpreted as the mixing time of the Markov chain $\{Y_k\}$ with precision δ . We have $t_\delta = \mathcal{O}(\log(1/\delta))$ under Assumption 3. For simplicity of presentation, we denote $A = A_1 + A_2 + 1$, $B = B_1 + B_2$, $t_k = t_{\alpha_k}$ and $\alpha_{i,j} = \sum_{k=i}^j \alpha_k$.

Condition 1. The step size sequence $\{\alpha_k\}$ is nonincreasing and satisfies $\alpha_{k-t_k, k-1} \leq \min\left(\frac{\varphi_2}{\varphi_3 A^2}, \frac{1}{4A}\right)$ for all $k \geq t_k$.

Condition 1 is analogous to the requirements for choosing step sizes imposed in Srikant and Ying (2019) and Chen et al. (2022), which study linear Markovian SA and nonlinear Markovian SA under a strongly pseudo-monotone operator, respectively. We will verify in Online Appendix 1.8 that Condition 1 is satisfied when using either a small enough constant step size (i.e., $\alpha_k \equiv \alpha$) or linearly diminishing step sizes (i.e., $\alpha_k = \alpha/(k+h)$ with properly chosen α and h), or polynomially diminishing step sizes (i.e., $\alpha_k = \alpha/(k+h)^\xi$ for all $\xi \in (0, 1)$), provided that α and h are properly chosen.

Now we are ready to state the Lyapunov drift inequality. The proof of the following proposition follows from a sequence of lemmas. Please see Sections 2.4.1 and 2.4.2 for more details.

Proposition 2. The following inequality holds for all $k \geq t_k$:

$$\begin{aligned} & \mathbb{E}[M_f^{\theta, g}(x_{k+1} - x^*)] \\ & \leq (1 - 2\varphi_2\alpha_k + \varphi_3 A^2 \tilde{\alpha}_k) \mathbb{E}[M_f^{\theta, g}(x_k - x^*)] \\ & \quad + \frac{\varphi_3 \tilde{\alpha}_k}{2u_{cm}^2} (A\|x^*\|_c + B)^2, \end{aligned} \quad (6)$$

where $\tilde{\alpha}_k = \alpha_k \alpha_{k-t_k, k-1}$.

Equation (6) is in the form of the desired one-step contractive inequality presented in Equation (3). To see this, suppose that we use a constant step size $\alpha_k \equiv \alpha$. Then we have $\tilde{\alpha}_k = \alpha^2 t_\alpha$, which is of order $o(\alpha)$ because $\lim_{\alpha \rightarrow 0} \alpha t_\alpha = 0$ under Assumption 3. Similarly, we show in Online Appendix 1.8 that $\tilde{\alpha}_k = o(\alpha_k)$ when using either linearly diminishing step sizes or polynomially diminishing step sizes.

2.3. Finite-Sample Analysis

In view of Proposition 2, to establish finite-sample bounds of Algorithm (1), we repeatedly use Equation (6)

and evaluate the final expression using the explicit choice of the step sizes. To present the results, let $c_1 = (\|x_0 - x^*\|_c + \|x_0\|_c + B/A)^2$ and $c_2 = (A\|x^*\|_c + B)^2$. Define $K = \min\{k \geq 0 : k \geq t_k\}$, which is well defined because t_k scales polynomially with k under Assumption 3.

Theorem 1. Suppose that Assumptions 1–4 are satisfied and $\{\alpha_k\}$ satisfies Condition 1. Then, for any $k \in [0, K-1]$, we have $\|x_k - x^*\|_c^2 \leq c_1$ almost surely. For any $k \geq K$, we have the following finite-sample bounds.

(1) When $\alpha_k \equiv \alpha$, we have

$$\mathbb{E}[\|x_k - x^*\|_c^2] \leq \varphi_1 c_1 (1 - \varphi_2 \alpha)^{k-t_\alpha} + \frac{\varphi_3 c_2}{\varphi_2} \alpha t_\alpha.$$

(2) Consider using diminishing stepsizes

(a) When $\alpha_k = \alpha/(k+h)$ with $\alpha < 1/\varphi_2$, we have

$$\mathbb{E}[\|x_k - x^*\|_c^2] \leq \varphi_1 c_1 \left(\frac{K+h}{k+h}\right)^{\varphi_2 \alpha} + \frac{8\alpha^2 \varphi_3 c_2}{1 - \varphi_2 \alpha} \frac{t_k}{(k+h)^{\varphi_2 \alpha}}.$$

(b) When $\alpha_k = \alpha/(k+h)$ with $\alpha = 1/\varphi_2$, we have

$$\mathbb{E}[\|x_k - x^*\|_c^2] \leq \varphi_1 c_1 \frac{K+h}{k+h} + 8\alpha^2 \varphi_3 c_2 \frac{t_k \log(k+h)}{k+h}.$$

(c) When $\alpha_k = \alpha/(k+h)$ with $\alpha > 1/\varphi_2$, we have

$$\mathbb{E}[\|x_k - x^*\|_c^2] \leq \varphi_1 c_1 \left(\frac{K+h}{k+h}\right)^{\varphi_2 \alpha} + \frac{8e\alpha^2 \varphi_3 c_2}{\varphi_2 \alpha - 1} \frac{t_k}{k+h}.$$

(3) When $\alpha_k = \alpha/(k+h)^\xi$ with $\xi \in (0, 1)$, we have

$$\mathbb{E}[\|x_k - x^*\|_c^2] \leq \varphi_1 c_1 e^{-\frac{\varphi_2 \alpha}{1-\xi} ((k+h)^{1-\xi} - (K+h)^{1-\xi})} + \frac{4\varphi_3 c_2 \alpha}{\varphi_2} \frac{t_k}{(k+h)^\xi}.$$

Remark 1. Because $t_\delta \leq \frac{\log(C/\sigma) + \log(1/\alpha)}{\log(1/\sigma)}$ under Assumption 3, we have $t_k \leq \frac{\xi \log(k+h) + \log(C/(\alpha\sigma))}{\log(1/\sigma)}$, which introduces an additional logarithmic factor in the bound.

In all cases of Theorem 1, we state the results as a combination of two terms. The first term is usually viewed as the “bias,” and it involves the error in the initial estimate x_0 through the constant c_1 . The second term is usually understood as the “variance” and hence involves the constant c_2 , which represents the noise variance at x^* . In view of Theorem 1, we see that constant step size is very efficient in driving the bias to zero but cannot eliminate the variance even asymptotically. This suggests using diminishing step sizes to eliminate the variance. When using linearly diminishing step sizes $\alpha_k = \alpha/(k+h)$, the convergence bounds crucially depend on the value of α , and the best convergence rate of $\tilde{\mathcal{O}}(1/k)$ is achieved with $\alpha > 1/\varphi_2$. When using $\alpha \leq 1/\varphi_2$ in Algorithm (1), in view of Theorem 1(2a), the convergence rate can be arbitrarily

slow. When using polynomially diminishing step sizes, although the convergence rate is the suboptimal $\mathcal{O}(\log(k)/k^\xi)$, it is more robust in the sense that it does not depend on α . Note that α in this case appears only as a multiplicative constant in the dominant term (Theorem 1(3)).

2.3.1. Connection to SGD. Although Theorem 1 is derived for SA algorithms that involve a contractive operator, they also recover finite-sample bounds for SGD with a smooth and strongly convex objective. To see this, let $J(x)$ be a differentiable objective function that is smooth and strongly convex with parameters C_L and C_σ , respectively. Consider the following SGD algorithm for minimizing $J(\cdot)$: $x_{k+1} = x_k + \alpha_k(-\eta\nabla J(x_k) + w_k)$, where $\eta > 0$ is a constant (Nemirovski et al. 2009, Bottou et al. 2018, Lan 2020). The SGD algorithm can be written in the form of our SA Algorithm (1) with $\bar{F}(x) = F(x, y) := -\nabla J(x) + x$. Furthermore, it is known that $\bar{F}(\cdot)$ is a Lipschitz operator with respect to the Euclidean norm $\|\cdot\|_2$, with Lipschitz constant $L_{SGD} = \max(|1 - \eta C_\sigma|, |1 - \eta C_L|)$ (Ryu and Boyd 2016). Therefore, when $\eta \in (0, 2/C_L)$, we have $L_{SGD} < 1$, and hence the operator $\bar{F}(\cdot)$ is a contraction with respect to $\|\cdot\|_2$.

2.3.2. Logarithmic Dependence on Dimension. Switching focus, we now revisit the constants $\{\varphi_i\}_{1 \leq i \leq 3}$ in Theorem 1. Note that $\{\varphi_i\}_{1 \leq i \leq 3}$ are determined by the choice of the smoothing norm $\|\cdot\|_s$ (through the constants u_{cs} and ℓ_{cs}) and the parameter θ . Depending on the contraction norm $\|\cdot\|_c$, the smoothing norm should be chosen accordingly to optimize the constants $\{\varphi_i\}_{1 \leq i \leq 3}$. In the following lemma, we consider two cases where $\|\cdot\|_c = \|\cdot\|_2$ and $\|\cdot\|_c = \|\cdot\|_\infty$, both of which will be useful when we study convergence bounds of RL algorithms.

Lemma 1 (Proof in Online Appendix 1.3). (1) When $\|\cdot\|_c = \|\cdot\|_2$, by choosing $\|\cdot\|_s = \|\cdot\|_2$ and $\theta = 1$, we have $\varphi_1 \leq 1$, $\varphi_2 \geq 1 - \beta$, and $\varphi_3 \leq 228$. (2) When $\|\cdot\|_c = \|\cdot\|_\infty$, by choosing $\|\cdot\|_s = \|\cdot\|_p$ with $p = 2 \log(d)$ and $\theta = \left(\frac{1+\beta}{2\beta}\right)^2 - 1$, we have $\varphi_1 \leq 3$, $\varphi_2 \geq \frac{1-\beta}{2}$, and $\varphi_3 \leq \frac{456e \log(d)}{1-\beta}$.

2.3.3. Order-Wise Tightness. Compared with ℓ_2 -norm contraction, where the constant φ_3 is bounded by a numerical constant, the upper bound for φ_3 has an additional factor of $\frac{\log(d)}{1-\beta}$ when we have ℓ_∞ -norm contraction. In general, we cannot hope to improve the convergence rate beyond $\tilde{\mathcal{O}}(1/k)$ or the dimension dependence beyond $\log(d)$ in the case of ℓ_∞ -norm contraction. To see this, consider the trivial case where $\bar{F}(x) \equiv \mathbf{0}$ and $\{w_k\}$ is an independent and identically distributed sequence of standard normal random vectors. In this case, Algorithm (1) becomes $x_{k+1} = x_k + \alpha_k(-x_k + w_k)$, which can

be viewed as an SA algorithm for solving the trivial equation $x = \mathbf{0}$, or an SGD algorithm for minimizing a quadratic objective $J(x) = \frac{1}{2}\|x\|_2^2$. When $\alpha_k = \frac{1}{k+1}$, the iterates x_k are simply the running averages of $\{w_k\}$, that is, $x_k = \frac{1}{k} \sum_{i=0}^{k-1} w_i$ for all $k \geq 1$, which implies $x_k \sim \frac{1}{\sqrt{k}} \mathcal{N}(0, I_d)$. It follows that $\mathbb{E}[\|x_k\|_\infty^2] = \mathcal{O}\left(\frac{\log(d)}{k}\right)$ (Vershynin 2018). Therefore, in this setting, our finite-sample bounds under ℓ_∞ -norm contraction are order-wise tight both in terms of the convergence rate and the dimensional dependence.

2.4. Proof of Theorem 1

The proof consists of three major steps. The first step is to show that the generalized Moreau envelope we constructed as a Lyapunov function produces a negative drift with respect to the stochastic iterates of the SA algorithm. The second step is to show that, all the error terms (i.e., discretization error and the stochastic error) in the SA algorithm are dominated by the negative drift, hence we have an overall one-step contractive inequality of the SA algorithm with respect to the Lyapunov function (Proposition 2). The smoothness property of the Lyapunov function and the geometric mixing property of the Markov chain $\{Y_k\}$ play important roles in this step. The last step is to repeatedly use the one-step contractive inequality to obtain the finite-sample bounds and to evaluate the final expression when using different step sizes. The proofs of all the technical lemmas used in this section are presented in Online Appendix 1.

2.4.1. Establishing the Negative Drift. Using the smoothness property of $M_f^{\theta, \mathcal{G}}(\cdot)$ (Proposition 1(1)) and the update equation in Equation (1), we have for all $k \geq 0$ that

$$\begin{aligned} & \mathbb{E}[M_f^{\theta, \mathcal{G}}(x_{k+1} - x^*)] \\ & \leq \mathbb{E}[M_f^{\theta, \mathcal{G}}(x_k - x^*)] + \underbrace{\alpha_k \mathbb{E}[\langle \nabla M_f^{\theta, \mathcal{G}}(x_k - x^*), \bar{F}(x_k) - x_k \rangle]}_{T_1: \text{Expected update}} \\ & \quad + \underbrace{\alpha_k \mathbb{E}[\langle \nabla M_f^{\theta, \mathcal{G}}(x_k - x^*), F(x_k, Y_k) - \bar{F}(x_k) \rangle]}_{T_2: \text{Error due to } Y_k} \\ & \quad + \underbrace{\alpha_k \mathbb{E}[\langle \nabla M_f^{\theta, \mathcal{G}}(x_k - x^*), w_k \rangle]}_{T_3: \text{Error due to } w_k} \\ & \quad + \underbrace{\frac{L\alpha_k^2}{2\theta} \mathbb{E}[\|F(x_k, Y_k) - x_k + w_k\|_s^2]}_{T_4: \text{Error due to discretization and noises}}. \end{aligned} \tag{7}$$

The term T_1 represents the expected update and produces a negative drift.

Lemma 2. It holds for all $k \geq 0$ that $\langle \nabla M_f^{\theta, g}(x_k - x^*), \bar{F}(x_k) - x_k \rangle \leq -2 \left(1 - \beta \frac{u_{cm}}{\ell_{cs}}\right) M_f^{\theta, g}(x_k - x^*)$.

2.4.2. Handling the Error Terms. What remains to do is to bound the error terms T_2 , T_3 , and T_4 . Using the assumption that $\{w_k\}$ is a martingale difference sequence and the tower property of conditional expectations, we see that $T_3 = 0$. As for the term T_4 , using the triangle inequality and the Lipschitz property of the operator $F(\cdot, \cdot)$ (Assumption 2), we have the following result.

Lemma 3. It holds for any $k \geq 0$ that $T_4 \leq \frac{2LA^2 u_{cm}^2 \alpha_k^2}{\theta \ell_{cs}^2} M_f^{\theta, g}(x_k - x^*) + \frac{L\alpha_k^2}{\theta \ell_{cs}^2} (A\|x^*\|_c + B)^2$.

To control the term T_2 , we need to carefully use a conditioning argument along with the geometric mixing of $\{Y_k\}$. The following lemma is useful for us to control T_2 .

Lemma 4. Given nonnegative integers $k_1 \leq k_2$ satisfying $\alpha_{k_1, k_2-1} \leq \frac{1}{4A}$, we have for all $k \in [k_1, k_2]$:

$$\|x_k - x_{k_1}\|_c \leq 2\alpha_{k_1, k_2-1} (A\|x_{k_1}\|_c + B), \quad \text{and}$$

$$\|x_k - x_{k_1}\|_c \leq 4\alpha_{k_1, k_2-1} (A\|x_{k_2}\|_c + B).$$

Because $\alpha_{k_1, k_2-1} \leq \frac{1}{4A}$ (Condition 1), Lemma 4 has the following corollary, which will also be frequently used in the derivation.

Corollary 1. Under the same conditions as in Lemma 4, we have for all $k \in [k_1, k_2]$ that

$$\|x_k - x_{k_1}\|_c \leq \min(\|x_{k_1}\|_c, \|x_{k_2}\|_c) + B/A.$$

To proceed and bound the term T_2 in Equation (7), we first show that the induced error is small ($o(\alpha_k)$ to be precise) if we replace x_k by x_{k-t_k} in the term T_2 , where we recall that t_k is the mixing time of the Markov chain $\{Y_k\}$ with precision α_k . This is where we use Lemma 4. After such replacement, the term T_2 becomes $\tilde{T}_2 = \alpha_k \mathbb{E}[\langle \nabla M_f^{\theta, g}(x_{k-t_k} - x^*), F(x_{k-t_k}, Y_k) - \bar{F}(x_{k-t_k}) \rangle]$. By the tower property of conditional expectations, we have

$$\begin{aligned} \tilde{T}_2 &= \alpha_k \mathbb{E}[\langle \nabla M_f^{\theta, g}(x_{k-t_k} - x^*), \\ &\quad \underbrace{\mathbb{E}[F(x_{k-t_k}, Y_k) | x_{k-t_k}, Y_{k-t_k}] - \bar{F}(x_{k-t_k})}_{=o(1) \text{ by geometric mixing}} \rangle]. \end{aligned}$$

Using the geometric mixing of $\{Y_k\}$, we see that the difference between $\mathbb{E}[F(x_{k-t_k}, Y_k) | x_{k-t_k}, Y_{k-t_k}]$ and $\bar{F}(x_{k-t_k})$ (which can be written as $\mathbb{E}_{Y \sim \mu_Y}[F(x, Y)]$ evaluated at $x = x_{k-t_k}$) is of $o(1)$, which implies $\tilde{T}_2 = o(\alpha_k)$. Formally, we have the following lemma.

Lemma 5. It holds for all $k \geq t_k$ that

$$\begin{aligned} T_2 &\leq \frac{112LA^2 u_{cm}^2 \alpha_k \alpha_{k-t_k, k-1}}{\theta \ell_{cs}^2} \mathbb{E}[M_f^{\theta, g}(x_k - x^*)] \\ &\quad + \frac{56L\alpha_k \alpha_{k-t_k, k-1}}{\theta \ell_{cs}^2} (A\|x^*\|_c + B)^2. \end{aligned}$$

Combining the upper bounds we obtained for the terms T_1 to T_4 in Equation (7), we arrive at the desired one-step contractive inequality presented in Proposition 2.

2.5. Solving the Recursion

The rest of the proof follows by repeatedly using Proposition 2 and evaluating the final expression when using different step size sequences. Specifically, because $\alpha_{k-t_k, k-1} \leq \varphi_2 / (\varphi_3 A^2)$ for all $k \geq K$ (Condition 1), we have by Proposition 2 that

$$\begin{aligned} \mathbb{E}[M(x_{k+1} - x^*)] &\leq (1 - \varphi_2 \alpha_k) \mathbb{E}[M(x_k - x^*)] \\ &\quad + \frac{\varphi_3 \alpha_k \alpha_{k-t_k, k-1}}{2u_{cm}^2} (A\|x^*\|_c + B)^2 \end{aligned}$$

for all $k \geq K$. Recursively using the previous inequality, we have for any $k \geq K$ that

$$\begin{aligned} &\mathbb{E}[\|x_k - x^*\|_c^2] \\ &\leq 2u_{cm}^2 \mathbb{E}[M(x_k - x^*)] \quad (\text{Proposition 1 (3)}) \\ &\leq 2u_{cm}^2 \mathbb{E}[M(x_K - x^*)] \prod_{j=K}^{k-1} (1 - \varphi_2 \alpha_j) \\ &\quad + \varphi_3 (A\|x^*\|_c + B)^2 \sum_{i=K}^{k-1} \alpha_i \alpha_{i-t_i, i-1} \prod_{j=i+1}^{k-1} (1 - \varphi_2 \alpha_j) \\ &\leq \frac{u_{cm}^2}{\ell_{cs}^2} \mathbb{E}[\|x_K - x^*\|_c^2] \prod_{j=K}^{k-1} (1 - \varphi_2 \alpha_j) \\ &\quad + \varphi_3 (A\|x^*\|_c + B)^2 \sum_{i=K}^{k-1} \alpha_i \alpha_{i-t_i, i-1} \prod_{j=i+1}^{k-1} (1 - \varphi_2 \alpha_j) \\ &= \varphi_1 \mathbb{E}[\|x_K - x^*\|_c^2] \prod_{j=K}^{k-1} (1 - \varphi_2 \alpha_j) \\ &\quad + \varphi_3 c_2 \sum_{i=K}^{k-1} \alpha_i \alpha_{i-t_i, i-1} \prod_{j=i+1}^{k-1} (1 - \varphi_2 \alpha_j), \end{aligned}$$

where we recall that $c_2 = (A\|x^*\|_c + B)^2$. According to Condition 1, we also have $\alpha_{0, k-1} \leq \frac{1}{4A}$ for any $k \in [0, K]$. Therefore, we have for any $k \in [0, K]$ that

$$\begin{aligned} \mathbb{E}[\|x_k - x^*\|_c^2] &\leq \mathbb{E}[(\|x_k - x_0\|_c + \|x_0 - x^*\|_c)^2] \\ &\leq (\|x_0 - x^*\|_c + \|x_0\|_c + B/A)^2 = c_1, \end{aligned}$$

where the second inequality follows from Corollary 1. Because the previous inequality implies $\mathbb{E}[\|x_k - x^*\|_c^2] \leq c_1$, we obtain for all $k \geq K$ that

$$\begin{aligned} \mathbb{E}[\|x_k - x^*\|_c^2] &\leq \varphi_1 c_1 \prod_{j=K}^{k-1} (1 - \varphi_2 \alpha_j) \\ &\quad + \varphi_3 c_2 \sum_{i=K}^{k-1} \alpha_i \alpha_{i-t, i-1} \prod_{j=i+1}^{k-1} (1 - \varphi_2 \alpha_j). \end{aligned} \quad (8)$$

Evaluating the right-hand side (RHS) of the previous inequality for different choices of the step sizes, we obtain all the cases presented in Theorem 1. See Online Appendix 1.8 for more details.

Remark 2. We handled Markovian noise in Section 2.4.2 by using a conditioning argument that exploits the geometric mixing (Assumption 3) of the underlying Markov chain, which in turn is a consequence of irreducibility and aperiodicity. An alternate approach to handle the Markovian noise is based on the Poisson equation (Benveniste et al. 2012, part II, chapter 1) and may need a different set of assumptions.

3. Applications in RL

We begin by introducing the underlying model for RL. The RL problem is usually modeled as an MDP where the transition probabilities and the reward function are unknown. In this work, we consider an MDP consisting of a finite set of states \mathcal{S} , a finite set of actions \mathcal{A} , a set of unknown transition probability matrices that are indexed by actions $\{P_a \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|} \mid a \in \mathcal{A}\}$, a reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$, and a discount factor $\gamma \in (0, 1)$. Because we work with finite MDPs, assuming bounded reward is indeed without loss of generality.

The goal in RL is to find an optimal policy π^* so that the cumulative reward received by using π^* is maximized. More formally, given a policy π , define its state-value function $V^\pi : \mathcal{S} \mapsto \mathbb{R}$ as $V^\pi(s) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k \mathcal{R}(S_k, A_k) \mid S_0 = s]$ for all s , where $\mathbb{E}_\pi[\cdot]$ means that the actions are selected according to the policy π . Then, a policy π^* is said to be optimal if and only if $V^{\pi^*}(s) \geq V^\pi(s)$ for any state s and policy π . It was shown that such an optimal policy always exists (Bertsekas and Tsitsiklis 1996).

In RL, the problem of finding an optimal policy is called the *control* problem. The most popular algorithm for solving the control problem is Q-learning (Watkins and Dayan 1992). Although the ultimate goal is to find an optimal policy, in RL, there is usually a smaller goal of finding the value function of a given policy, which is called the *prediction* problem and is usually solved with TD-learning and its variants (Sutton 1988). Both Q-learning and TD learning are by nature SA algorithms for solving variants of the Bellman equation. Therefore, our results on SA unify the finite-sample analysis of value-based RL algorithms.

We next present three case studies: the off-policy V-trace algorithm, the on-policy n -step TD algorithm, and the Q-learning algorithm. We also establish finite-sample guarantees of the TD(λ) algorithm, which is presented in Section 5.

3.1. Off-Policy Prediction: V-Trace

TD learning for solving the prediction problem can be divided into two categories: on-policy TD and off-policy TD. In off-policy TD, one uses samples generated by a *behavior* policy π_b to learn the value function of the *target* policy $\pi \neq \pi_b$. Off-policy sampling is used for three important reasons. (1) It is typically necessary to have an exploration component in the behavior policy π_b which makes it different from the target policy π . (2) It is used in multiagent training where various agents collect rewards using a behavior policy that is lagging with respect to the target policy in an actor-critic framework (Espelholt et al. 2018). (3) It enables learning using historical data, which improves sample efficiency.

Off-policy TD learning is usually implemented through importance sampling to obtain an unbiased estimate of the desired value function. However, the variance in the estimate can explode because the importance sampling factor can be very large (Glynn and Iglehart 1989). Therefore, a well-known and fundamental difficulty in off-policy TD learning with importance sampling is to balance the bias-variance tradeoff.

Recently, Espelholt et al. (2018) proposed an off-policy TD learning algorithm called the V-trace, where they introduced two truncation levels in the importance sampling weights. Their construction (through two separate clippers) crucially allows the algorithm to control the bias in the limit (through one clipper), whereas the other clipper mainly controls the variance in the estimate. The V-trace algorithm has had a huge practical impact: It has been implemented in distributed RL architectures and platforms like IMPALA (Espelholt et al. 2018), a TensorFlow implementation, and TorchBeast (Küttler et al. 2019), a PyTorch implementation, for multiagent training besides being used at scale in a recent Deepmind City Navigation Project “Street Learn” (Mirowski et al. 2018). Given its impact, a theoretical understanding of the effects of the truncation levels on the convergence rate is important for us to determine how to tune them to get the best performance of V-trace.

3.2. Algorithm

We next present the V-trace algorithm for off-policy TD learning. Recall that we denote π_b as the behavior policy and π as the target policy. Let n be a positive integer. Define $c(s, a) = \min\left(\bar{c}, \frac{\pi(a|s)}{\pi_b(a|s)}\right)$ and $\rho(s, a) = \min\left(\bar{\rho}, \frac{\pi(a|s)}{\pi_b(a|s)}\right)$ as the two truncated importance sampling factors

at state-action pair (s, a) , where \bar{c} and $\bar{\rho}$ are the two different truncation levels satisfying $\bar{\rho} \geq \bar{c} > 0$.

Let $\{(S_k, A_k)\}$ be a sequence of samples collected under the behavior policy π_b . Then, with initialization $V_0 \in \mathbb{R}^{|\mathcal{S}|}$, for each $k \geq 0$, the V -trace algorithm updates the estimate V_k of the target value function V^π according to

$$V_{k+1}(s) = \begin{cases} V_k(s) + \alpha_k \sum_{i=k}^{k+n-1} \gamma^{i-k} \left(\prod_{j=k}^{i-1} c(S_j, A_j) \right) \\ \rho(S_i, A_i) \Gamma_1(V_k, S_i, A_i, S_{i+1}), & S_k = s, \\ V_k(s), & S_k \neq s, \end{cases} \quad (9)$$

where $\Gamma_1(V_k, S_i, A_i, S_{i+1}) = \mathcal{R}(S_i, A_i) + \gamma V_k(S_{i+1}) - V_k(S_i)$ is the temporal difference. The V -trace algorithm presented above can be viewed as an extension of the well-known n -step on-policy TD learning to the setting where we use off-policy sampling. Specifically, the truncated importance sampling factors are introduced due to the discrepancy between the behavior policy π_b and the target policy π . Consider the special case where $\pi = \pi_b$. By choosing $\bar{\rho} = \bar{c} = 1$, which implies $c(s, a) = \rho(s, a) = 1$, Algorithm (9) reduces to the standard n -step TD learning.

To establish finite-sample bounds of Algorithm (9), we make the following assumption.

Assumption 5. *The behavior policy π_b satisfies $\{a \in \mathcal{A} | \pi(a|s) > 0\} \subseteq \{a \in \mathcal{A} | \pi_b(a|s) > 0\}$ for all $s \in \mathcal{S}$, and the Markov chain $\{S_k\}$ induced by π_b is irreducible and aperiodic.*

The first part of Assumption 5 is called the *coverage* assumption, which states that, for any state, if it is possible to explore a specific action under the target policy π , then it is also possible to explore such an action under the behavior policy π_b . This requirement is necessary for off-policy RL. The second part of Assumption 5 is imposed to ensure the exploration of π_b and implies that $\{S_k\}$ has a unique stationary distribution, denoted by $\kappa_S \in \Delta^{|\mathcal{S}|}$, the minimum component of which is denoted as $\mathcal{K}_{S, \min}$. Moreover, the Markov chain $\{S_k\}$ induced by π_b mixes at a geometric rate (Levin and Peres 2017).

3.2.1. Properties of the V-Trace Algorithm. We next follow the blueprint presented in Section 1.2.2 to establish the finite-sample bounds of the V -trace algorithm using our SA results. We begin with the reformulation. For any $k \geq 0$, let $Y_k = (S_k, A_k, \dots, S_{k+n-1}, A_{k+n-1}, S_{k+n})$. It is clear that $\{Y_k\}$ is also a Markov chain, the state space of which is denoted by \mathcal{Y} and is finite. Define an

operator $F: \mathbb{R}^{|\mathcal{S}|} \times \mathcal{Y} \mapsto \mathbb{R}^{|\mathcal{S}|}$ as

$$\begin{aligned} [F(V, y)](s) &= [F(V, s_0, a_0, \dots, s_{n-1}, a_{n-1}, s_n)](s) \\ &= \mathbb{1}_{\{s_0=s\}} \sum_{i=0}^{n-1} \gamma^i \left(\prod_{j=0}^{i-1} c(S_j, a_j) \right) \rho(s_i, a_i) \\ &\quad \Gamma_1(V, s_i, a_i, s_{i+1}) + V(s) \end{aligned}$$

for all $V \in \mathbb{R}^{|\mathcal{S}|}$, $y = (s_0, a_0, \dots, s_n) \in \mathcal{Y}$, and $s \in \mathcal{S}$. Then, Equation (9) can be equivalently written as

$$V_{k+1} = V_k + \alpha_k (F(V_k, Y_k) - V_k), \quad (10)$$

which is the same form of Algorithm (1) with $x_k = V_k$ and $w_k = 0$. Under Assumption 5, we next establish the properties of the operator $F(\cdot, \cdot)$ and the Markov chain $\{Y_k\}$, which will enable us to apply Theorem 1. The following notation is needed to state the result.

For any policy π , let $P_\pi \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ be the transition probability matrix of the Markov chain $\{S_k\}$ induced by π and let $R_\pi \in \mathbb{R}^{|\mathcal{S}|}$ be such that $R_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}(s, a)$ for all s . Let $D_c, D_\rho \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ be diagonal matrices with diagonal components $\{\mathbb{E}_{\pi_b}[c(S, A) | S = s]\}_{s \in \mathcal{S}}$ and $\{\mathbb{E}_{\pi_b}[\rho(S, A) | S = s]\}_{s \in \mathcal{S}}$, respectively. Let $D_{c, \min}$ (respectively, $D_{\rho, \min}$) be the minimum diagonal component of D_c (respectively, D_ρ). Let $\mathcal{K}_S = \text{diag}(\kappa_S) \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ and $\mathcal{K}_{S, \min} = \min_{s \in \mathcal{S}} \kappa_S(s)$. Define two policies $\pi_{\bar{c}}$ and $\pi_{\bar{\rho}}$ as $\pi_{\bar{c}}(a|s) = \frac{c(s, a) \pi_b(a|s)}{\mathbb{E}_{\pi_b}[c(S, A) | S = s]}$ and $\pi_{\bar{\rho}}(a|s) = \frac{\rho(s, a) \pi_b(a|s)}{\mathbb{E}_{\pi_b}[\rho(S, A) | S = s]}$ for all state-action pairs (s, a) . Let $\eta_n(x) = \sum_{i=0}^{n-1} x^i$ for any $x > 0$ and positive integer n .

Proposition 3 (Proof in Online Appendix 2.1). *Under Assumptions 5, we have the following.*

(1) *The operator $F(\cdot)$ satisfies (a) $\|F(V_1, y) - F(V_2, y)\|_\infty \leq (2\bar{\rho} + 1) \eta_n(\gamma \bar{c}) \|V_1 - V_2\|_\infty$ for all $V_1, V_2 \in \mathbb{R}^{|\mathcal{S}|}$ and $y \in \mathcal{Y}$, and (b) $\|F(\mathbf{0}, y)\|_\infty \leq \bar{\rho} \eta_n(\gamma \bar{c})$ for all $y \in \mathcal{Y}$.*

(2) *The Markov chain $\{Y_k\}$ has a unique stationary distribution μ_Y . Moreover, there exist $C_1 > 0$ and $\sigma_1 \in (0, 1)$ such that $\max_{y \in \mathcal{Y}} \|P_Y^{k+n}(y, \cdot) - \mu_Y(\cdot)\|_{TV} \leq C_1 \sigma_1^k$ for all $k \geq 0$.*

(3) *Define the expected operator $\bar{F}: \mathbb{R}^{|\mathcal{S}|} \mapsto \mathbb{R}^{|\mathcal{S}|}$ of $F(\cdot, \cdot)$ as $\bar{F}(V) = \mathbb{E}_{Y \sim \mu_Y}[F(V, Y)]$ for all $V \in \mathbb{R}^{|\mathcal{S}|}$. Then, (a) $\bar{F}(\cdot)$ is explicitly given as $\bar{F}(V) = [I - \mathcal{K}_S \sum_{i=0}^{n-1} (\gamma D_c P_{\pi_{\bar{c}}})^i] D_\rho (I - \gamma P_{\pi_{\bar{\rho}}}) V + \mathcal{K}_S \sum_{i=0}^{n-1} (\gamma D_c P_{\pi_{\bar{c}}})^i D_\rho R_{\pi_{\bar{\rho}}}$, (b) $\bar{F}(\cdot)$ is a contraction mapping with respect to $\|\cdot\|_\infty$, with contraction factor $\beta_1 := 1 - \mathcal{K}_{S, \min} \frac{(1-\gamma)(1-(\gamma D_{c, \min})^n) D_{\rho, \min}}{1-\gamma D_{c, \min}}$, and (c) $\bar{F}(\cdot)$ has a unique fixed point $V^{\pi_{\bar{\rho}}}$, which is the value function of the policy $\pi_{\bar{\rho}}$.*

3.2.2. Finite-Sample Bounds of V-Trace. Proposition 3 enables us to apply Theorem 1 to establish the finite-sample bounds of V -trace. For ease of presentation, we

here only state the result for using a constant step size (i.e., $\alpha_k \equiv \alpha$), the proof of which is presented in Online Appendix 2.2. The convergence rate for using diminishing step sizes is presented in Online Appendix 2.4.

Theorem 2. *Suppose that Assumption 5 is satisfied, and the constant step size α is chosen such that $\alpha(t_\alpha + n) \leq c_{V,0} \frac{(1-\beta_1)^2}{(\bar{\rho}+1)^2 \eta_n^2(\gamma\bar{c}) \log(|S|)^2}$, where t_α is the mixing time of the Markov chain $\{S_k\}$ (induced by π_b) with precision α . Then we have for all $k \geq t_\alpha + n$ that*

$$\mathbb{E}[\|V_k - V^{\pi_{\bar{\rho}}}\|_\infty^2] \leq c_{V,1} \left(1 - \frac{1-\beta_1}{2}\alpha\right)^{k-t_\alpha-n} + c_{V,2} \frac{\log(|S|)(\bar{\rho}+1)^2 \eta_n(\gamma\bar{c})^2}{(1-\beta_1)^2} \alpha(t_\alpha + n),$$

where $c_{V,1} = 3(\|V_0 - V^{\pi_{\bar{\rho}}}\|_\infty + \|V_0\|_\infty + 1)^2$ and $c_{V,2} = 3648 e(\|V^{\pi_{\bar{\rho}}}\|_\infty + 1)^2$. Moreover, we have

$$\|V^{\pi_{\bar{\rho}}} - V^\pi\|_\infty \leq \frac{\gamma \max_{s \in S} \|\pi(\cdot|s) - \pi_{\bar{\rho}}(\cdot|s)\|_1}{(1-\gamma)^2}.$$

The convergence bound here is qualitatively similar to Theorem 1. The truncation level $\bar{\rho}$ determines the limit point $V^{\pi_{\bar{\rho}}}$. In addition, $\bar{\rho}$ plays a role in the second term (which captures the variance in the algorithm) on the right-hand side of the finite-sample bound. In practice, $\bar{\rho}$ should be tuned to balance the tradeoff between the bias at the limit point and the convergence variance. The truncation level \bar{c} mainly controls the variance term in the convergence bound through the factor $\eta_n(\gamma\bar{c})$. To formally characterize how the parameters of V -trace impact the convergence rate, we next derive the sample complexity bound.

Corollary 2 (Proof in Online Appendix 2.3). *When $\bar{\rho} = 1/\min_{s,a} \pi_b(a|s)$, to make $\mathbb{E}[\|V_k - V^\pi\|_\infty] \leq \epsilon$, the sample complexity is*

$$\underbrace{\mathcal{O}\left(\frac{\log^2(1/\epsilon)}{\epsilon^2}\right)}_{\text{Accuracy}} \underbrace{\tilde{\mathcal{O}}\left(\frac{1}{(1-\gamma)^5}\right)}_{\text{Effective horizon}} \underbrace{\tilde{\mathcal{O}}\left(\frac{n\bar{\rho}^2 \eta_n(\gamma\bar{c})^2}{D_{\rho,\min}^3 \eta_n(\gamma D_{c,\min})^3}\right)}_{\text{Off-policy } n\text{-step TD}} \underbrace{\tilde{\mathcal{O}}(\mathcal{K}_{S,\min}^{-3})}_{\text{Quality of exploration}}.$$

Remark 3. When there is a nonvanishing bias in the bound, which in our case corresponds to $V^\pi \neq V^{\pi_{\bar{\rho}}}$, the sample complexity is not well defined. Khodadadian et al. (2021, appendix C.1) provide a detailed discussion. Therefore, we choose $\bar{\rho} = 1/\min_{s,a} \pi_b(a|s) \geq \max_{s,a} \pi(a|s)/\pi_b(a|s)$ to eliminate the bias due to introducing the truncation level $\bar{\rho}$. In this case, because

$\rho(s,a) = \frac{\pi(a|s)}{\pi_b(a|s)}$, we have $V^{\pi_{\bar{\rho}}} = V^\pi$. This is merely for mathematical rigor.

From Corollary 2, we see that the dependence on the required accuracy level is $\tilde{\mathcal{O}}(\epsilon^2)$, which is known to be optimal up to a logarithmic factor. In addition, we have an $\tilde{\mathcal{O}}(1/(1-\gamma)^5)$ dependence on the effective horizon, and at least a cubic dependence on the size of the state-space $|S|$. To see this, observe that $\mathcal{K}_{S,\min} \leq 1/|S|$ implies $\mathcal{K}_{S,\min}^{-3} \geq |S|^3$.

The feature of the V -trace algorithm is captured by the term $\tilde{\mathcal{O}}\left(\frac{n\bar{\rho}^2 \eta_n(\gamma\bar{c})^2}{D_{\rho,\min}^3 \eta_n(\gamma D_{c,\min})^3}\right)$, which is a consequence of performing n -step off-policy TD learning with truncated importance sampling factors. The impact of the parameter n will be analyzed in detail in Section 3.3, where we study on-policy n -step TD and the efficiency of bootstrapping. We here focus on the two truncation levels \bar{c} and $\bar{\rho}$. First, we choose $\bar{\rho} = 1/\min_{s,a} \pi_b(a|s) \geq 1/|\mathcal{A}|$ to ensure that $V^{\pi_{\bar{\rho}}} = V^\pi$, which introduces a factor of at least $|\mathcal{A}|^{-2}$ in the sample complexity. The dependence of the sample complexity on the truncation level \bar{c} is through the term $\eta_n(\gamma\bar{c})$. To avoid an exponential factor of n , we need to aggressively truncate the importance sampling factors by choosing $\bar{c} < 1/\gamma$.

3.2.3. Related Literature on V -Trace. The V -trace algorithm was first proposed in Espeholt et al. (2018) as an off-policy variant of the n -step TD learning. The key novelty in V -trace is that the two truncation levels \bar{c} and $\bar{\rho}$ are introduced in the importance sampling factors to separately control the asymptotic bias and the variance. The asymptotic convergence of V -trace in the case where $n = \infty$ was established in Espeholt et al. (2018). This is the first finite-sample analysis of V -trace with asynchronous update. Other algorithms that are closely related to V -trace are the off-policy $Q^\pi(\lambda)$ (Hartuyunyan et al. 2016), tree-backup $TB(\lambda)$ (Precup et al. 2000), and retrace(λ) (Munos et al. 2016). A recent work (Chen et al. 2020) presents a unified analysis of these algorithms.

3.3. On-Policy Prediction: n -Step TD

In this section, we consider on-policy n -step TD learning algorithm, which can be viewed as a special case of the V -trace algorithm with $\pi_b = \pi$ and $\bar{c} = \bar{\rho} = 1$. Therefore, one can directly apply Theorem 2 to this setting and obtain finite-sample bounds for n -step TD. However, we will show that due to on-policy sampling, there are other properties (i.e., ℓ_2 -norm contraction) of the n -step TD learning we can exploit to obtain tighter bounds.

Similarly as in the previous section, we make the following assumption to ensure the exploration of the

behavior policy π , which is also the target policy in on-policy TD.

Assumption 6. *The Markov chain $\{S_k\}$ induced by π is irreducible and aperiodic.*

Assumption 6 implies that $\{S_k\}$ has a unique stationary distribution $\kappa_S \in \Delta^{|S|}$ (the smallest component of which is denoted as $\mathcal{K}_{S,\min}$). In addition, the Markov chain $\{S_k\}$ mixes at a geometric rate (Levin and Peres 2017).

3.3.1. Finite-Sample Analysis of n-Step TD. To reformulate Equation (9) (with $\pi_b = \pi$ and $\bar{c} = \bar{\rho} = 1$) in the form of Algorithm (1), the operator $F(\cdot, \cdot)$ and the Markov chain $\{Y_k\}$ are defined in the same way as in Section 3.2.1. We next present the ℓ_p -norm contraction property of the operator $\bar{F}(\cdot) = \mathbb{E}_{Y \sim \mu_Y}[F(\cdot, Y)]$ associated with on-policy n -step TD. Other properties regarding the operator $F(\cdot, \cdot)$ and the Markov chain $\{Y_k\}$ (e.g., Lipschitz continuity, geometric mixing) are presented in Online Appendix 3.

Proposition 4. *The operator $\bar{F}(\cdot)$ is a contraction mapping with respect to the ℓ_p -norm $\|\cdot\|_p$ for any $p \in [1, \infty]$, with a common contraction factor $\beta_2 = 1 - \mathcal{K}_{S,\min}(1 - \gamma^n)$.*

Unlike in the off-policy V -trace setting, where the operator $\bar{F}(\cdot)$ is only shown to be a contraction mapping with respect to the ℓ_∞ -norm, the operator $\bar{F}(\cdot)$ associated with on-policy n -step TD is a contraction mapping with respect to $\|\cdot\|_p$ for any $p \in [1, \infty]$, in particular, the ℓ_2 -norm. The ℓ_2 -norm contraction is the property we are going to exploit to establish the finite-sample bounds of n -step TD. For ease of presentation, we next state the guarantees when using a constant step size $\alpha_k \equiv \alpha$. The results for using diminishing step sizes are presented in Online Appendix 3.2.

Theorem 3 (Proof in Online Appendix 3.1). *Suppose that Assumption 6 is satisfied and α is chosen such that $\alpha(t_\alpha + n) \leq \hat{c}_0(1 - \beta_2)$ (where \hat{c}_0 is a numerical constant and t_α is the mixing time of the Markov chain $\{S_k\}$ induced by π with precision α). Then we have for all $k \geq t_\alpha + n$:*

$$\mathbb{E}[\|V_k - V^\pi\|_2^2] \leq \hat{c}_1(1 - (1 - \beta_2)\alpha)^{k-t_\alpha-n} + \hat{c}_2 \frac{\alpha(t_\alpha + n)}{(1 - \gamma)^2(1 - \beta_2)},$$

where $\hat{c}_1 = (\|V_0 - V^\pi\|_2 + \|V_0\|_2 + 4)^2$ and $\hat{c}_2 = 228(4(1 - \gamma)\|V^\pi\|_2 + |S|^{1/2})^2$.

An important idea in n -step TD is to use the parameter n to adjust the bootstrapping effect. Specifically, $n = 0$ corresponds to extreme bootstrapping, whereas $n = \infty$ corresponds to using the Monte Carlo method for estimating V^π and hence no bootstrapping. A long-

standing question in RL is about the efficiency of bootstrapping, that is, the choice of n that leads to the optimal performance of the algorithm (Sutton and Barto 2018).

By evaluating the convergence bounds in Theorem 3 with only n -dependent terms, we see that the bias term is of $(1 - \Theta(1 - \gamma^n))^k$. Because the mixing time t_α of the Markov chain $\{S_k\}$ does not depend on n , the variance term is of $\mathcal{O}(n/(1 - \gamma^n))$. Now we can clearly see that as n increases, the bias goes down while the variance goes up, thereby demonstrating a bias-variance tradeoff in the n -step TD learning algorithm. To provide an estimate of the optimal value of n , we next derive the sample complexity of n -step TD based on Theorem 3.

Corollary 3. *To achieve $\mathbb{E}[\|V_k - V^\pi\|_2] \leq \epsilon$, the sample complexity is*

$$\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2}\right) \tilde{\mathcal{O}}\left(\frac{1}{(1 - \gamma)^4 \mathcal{K}_{S,\min}^2}\right) \underbrace{\tilde{\mathcal{O}}\left(\frac{n}{(1 - \gamma^n)^2}\right)}_{\text{The impact of } n}.$$

In light of the dependence on the parameter n , we can optimize the choice of n to minimize the function $\frac{n}{(1 - \gamma^n)^2}$ over all positive integers. By doing that, we have $n_{\text{opt}} \sim \min(1, \lfloor 1/\log(1/\gamma) \rfloor)$, where $\lfloor x \rfloor$ stands for the integer closest to x . We point out that this choice of n was derived based on minimizing our upper bound. To ensure that it is indeed the optimal choice, we need to derive a matching lower bound on the sample complexity, which is a future research direction.

Compared with the off-policy V -trace, it is clear that the on-policy n -step TD has a better sample complexity. Specifically, it has a better dependency on the effective horizon, which is $\tilde{\mathcal{O}}((1 - \gamma)^{-4})$, and a better dependency on the minimum component $\mathcal{K}_{S,\min}$ of the stationary distribution of $\{S_k\}$. The main reason for such an improvement in the sample complexity is that we are able to exploit the ℓ_2 -norm contraction of the operator $\bar{F}(\cdot)$ in n -step TD.

3.3.2. Related Literature on n-Step TD. The notion of using multistep returns instead of only one-step return was introduced in Watkins (1989). Sutton and Barto (2018, chapter 7) provide more details about n -step TD. The asymptotic convergence of n -step TD was established using standard SA results under contraction assumption (Bertsekas and Tsitsiklis 1996). Regarding the choice of n , it was observed in empirical experiments that n -step TD (with a suitable choice of n) usually outperforms the one-step TD and Monte Carlo methods (Singh and Sutton 1996, Sutton and Barto 2018). However, a theoretical understanding of this phenomenon is not well established in the literature.

3.4. Off-Policy Control: Q-Learning

Thus far, we considered TD learning algorithms for solving the prediction problem. In this section, we consider the Q-learning algorithm (Watkins and Dayan 1992) for solving the control problem (i.e., finding an optimal policy). Define the Q-function associated with a policy π as $Q_\pi(s, a) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k \mathcal{R}(S_k, A_k) | S_0 = s, A_0 = a]$ for all (s, a) . Denote Q^* as the Q-function associated with an optimal policy π^* . All optimal policies share the same optimal Q-function. The motivation of the Q-learning algorithm is based on the following result (Bertsekas and Tsitsiklis 1996, Sutton and Barto 2018): π^* is an optimal policy if and only if $\pi^*(\cdot | s)$ is supported on the set $\arg \max_{a \in \mathcal{A}} Q^*(s, a)$ for any (s, a) . The previous result implies that knowing the optimal Q-function is sufficient to compute an optimal policy.

To find the optimal Q-function, we next introduce the Bellman equation. Let $\mathcal{H} : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \mapsto \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ be the Bellman operator defined as

$$\begin{aligned} & [\mathcal{H}(Q)](s, a) \\ &= \mathcal{R}(s, a) + \gamma \mathbb{E} \left[\max_{a' \in \mathcal{A}} Q(S_{k+1}, a') \mid S_k = s, A_k = a \right], \\ & \quad \forall Q \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}, (s, a). \end{aligned}$$

Then it has been shown that Q^* is the unique solution to the fixed-point equation $\mathcal{H}(Q) = Q$ (Bertsekas and Tsitsiklis 1996). The Q-learning algorithm can be viewed as an SA algorithm to solve the Bellman equation.

In Q-learning, we first collect a sample trajectory $\{(S_k, A_k)\}$ using a suitable behavior policy π_b . Then, with initialization $Q_0 \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, the iterate Q_k is updated as

$$Q_{k+1}(s, a) = \begin{cases} Q_k(s, a) + \alpha_k \left(\mathcal{R}(S_k, A_k) \right. \\ \left. + \gamma \max_{a' \in \mathcal{A}} Q_k(S_{k+1}, a') - Q_k(S_k, A_k) \right), & (s, a) = (S_k, A_k), \\ Q_k(s, a), & (s, a) \neq (S_k, A_k). \end{cases} \quad (11)$$

To establish the finite-sample bounds of Q-learning, we make the following assumption.

Assumption 7. The behavior policy π_b satisfies $\pi_b(a|s) > 0$ for all (s, a) , and the Markov chain $\{S_k\}$ induced by π_b is irreducible and aperiodic.

The requirement that $\pi_b(a|s) > 0$ for all (s, a) is necessary even for the asymptotic convergence of Q-learning (Tsitsiklis 1994). The irreducibility and aperiodicity assumption is also standard in the existing work (Tsitsiklis and Van Roy 1997, 1999). Because we work with finite-state MDPs, Assumption 7 implies that $\{S_k\}$ has a unique stationary distribution, denoted by $\kappa_S \in \Delta^{|\mathcal{S}|}$, and $\{S_k\}$ mixes at a geometric rate (Levin and Peres 2017). Similarly, we let $\mathcal{K}_{S, \min} = \min_{s \in \mathcal{S}} \kappa_S(s)$.

3.4.1. Properties of the Q-Learning Algorithm. Recall the definition of the operator $F(\cdot, \cdot)$ and the Markov chain $\{Y_k\}$ in Section 1.2.2. We next establish their properties in the following proposition, which guarantees that the assumptions needed to apply Theorem 1 are satisfied in the context of Q-learning. Let $\mathcal{K}_{SA} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}||\mathcal{A}|}$ be the diagonal matrix with $\{\kappa_S(s)\pi_b(a|s)\}_{(s,a) \in \mathcal{S} \times \mathcal{A}}$ on its diagonal. Let $\mathcal{K}_{SA, \min} = \min_{(s,a)} \kappa_S(s)\pi_b(a|s)$, which is strictly positive under Assumption 7.

Proposition 5 (Proof of Online Appendix 4.1). *Suppose that Assumption 7 is satisfied, Then, we have the following results.*

(1) The operator $F(\cdot, \cdot)$ satisfies (a) $\|F(Q_1, y) - F(Q_2, y)\|_\infty \leq 2\|Q_1 - Q_2\|_\infty$ for any $Q_1, Q_2 \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, and $y \in \mathcal{Y}$, and (b) $\|F(\mathbf{0}, y)\|_\infty \leq 1$ for all $y \in \mathcal{Y}$.

(2) The Markov chain $\{Y_k\}$ has a unique stationary distribution μ_Y , and there exist $C_3 > 0$ and $\sigma_3 \in (0, 1)$ such that $\max_{y \in \mathcal{Y}} \|P^{k+1}(y, \cdot) - \mu_Y(\cdot)\|_{TV} \leq C_3 \sigma_3^k$ for any $k \geq 0$.

(3) Define the expected operator $\bar{F} : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \mapsto \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ of $F(\cdot, \cdot)$ as $\bar{F}(Q) = \mathbb{E}_{Y \sim \mu_Y}[F(Q, Y)]$. Then, (a) $\bar{F}(\cdot)$ is explicitly given by $\bar{F}(Q) = \mathcal{K}_{SA} \mathcal{H}(Q) + (I - \mathcal{K}_{SA})Q$, (b) $\bar{F}(\cdot)$ is a contraction mapping with respect to $\|\cdot\|_\infty$, with contraction factor $\beta_3 := 1 - \mathcal{K}_{SA, \min}(1 - \gamma)$, and (c) $\bar{F}(\cdot)$ has a unique fixed-point Q^* .

Observe that the (s, a) th entry of $\bar{F}(Q)$ is given by $\kappa_S(s)\pi_b(a|s)[\mathcal{H}(Q)](s, a) + (1 - \kappa_S(s)\pi_b(a|s))Q(s, a)$, which can be viewed as a convex combination of “performing update” and “not performing update,” hence captures the nature of asynchronism as illustrated in Section 1.2.2.

3.4.2. Finite-Sample Bounds of Q-Learning. Proposition 5 enables us to apply Theorem 1 and Corollary 1(2) to Q-learning. For ease of presentation, we only state the result for using a constant step size $\alpha_k \equiv \alpha$. See Online Appendix 4.3 for the results when using diminishing step sizes.

Theorem 4. *Suppose that Assumption 7 is satisfied and α is chosen such that $\alpha(t_\alpha + 1) \leq c_{Q,0} \frac{(1-\beta_3)^2}{\log(|\mathcal{S}||\mathcal{A}|)}$ (where $c_{Q,0}$ is a numerical constant and t_α is the mixing time of the Markov chain $\{S_k\}$ induced by π_b with precision α). Then we have for all $k \geq t_\alpha + 1$ that*

$$\begin{aligned} \mathbb{E}[\|Q_k - Q^*\|_\infty^2] &\leq c_{Q,1} \left(1 - \frac{(1-\beta_3)\alpha}{2} \right)^{k-t_\alpha-1} \\ &\quad + c_{Q,2} \frac{\log(|\mathcal{S}||\mathcal{A}|)}{(1-\beta_3)^2} \alpha(t_\alpha + 1), \end{aligned}$$

where $c_{Q,1} = 3(\|Q_0 - Q^*\|_\infty + \|Q_0\|_\infty + 1)^2$ and $c_{Q,2} = 912e(3\|Q^*\|_\infty + 1)^2$.

Based on Theorem 4, we next derive the sample complexity of Q -learning.

Corollary 4. Given $\epsilon > 0$, to achieve $\mathbb{E}[\|Q_k - Q^*\|_\infty] \leq \epsilon$, the sample complexity is

$$\underbrace{\mathcal{O}\left(\frac{\log^2(1/\epsilon)}{\epsilon^2}\right)}_{\text{Accuracy}} \underbrace{\tilde{\mathcal{O}}\left(\frac{1}{(1-\gamma)^5}\right)}_{\text{Effective horizon}} \underbrace{\tilde{\mathcal{O}}(\mathcal{K}_{SA,\min}^{-3})}_{\text{Quality of exploration}}.$$

From Corollary 4, we see that the dependence on the accuracy ϵ is $\mathcal{O}(\epsilon^{-2}\log^2(1/\epsilon))$, and the dependence on the effective horizon is $\tilde{\mathcal{O}}((1-\gamma)^{-5})$. These two results match with known results in the literature (Beck and Srikant 2013, Li et al. 2020). The parameter $\mathcal{K}_{SA,\min}$ captures the quality of exploration of the behavior policy π_b . Because $\mathcal{K}_{SA,\min} \geq 1/|\mathcal{S}||\mathcal{A}|$, we see that there is at least a cubic dependence on the size of the state-action space.

3.4.3. Related Literature on Q-Learning. The Q -learning algorithm (Watkins and Dayan 1992) is perhaps one of the most well-known algorithms in the RL literature. The asymptotic convergence of Q -learning was established in Tsitsiklis (1994), Jaakkola et al. (1993), and Borkar and Meyn (2000) and the asymptotic convergence rate in Szepesvári et al. (1997) and Devraj and Meyn (2017). Beyond asymptotic behavior, finite-sample analysis of Q -learning was also thoroughly studied in the literature (Even-Dar and Mansour 2003, Beck and Srikant 2013, Jin et al. 2018, Li et al. 2020, Qu and Wierman 2020). The state-of-the-art sample complexity for asynchronous Q -learning goes to Li et al. (2020), which has a better dependence on the size of the state-action space compared with this work. In addition to being a contractive SA, Q -learning has many other properties, such as the update equation being asynchronous, the iterates being uniformly bounded by a constant (Gosavi 2006), which are used in Li et al. (2020) for their analysis. Although our SA framework did not exploit these properties of Q -learning (which results in a suboptimal sample complexity), it is a more general framework that enables us to study a wide variety of algorithms beyond Q -learning. A typical example is the V -trace algorithm studied earlier. Because of off-policy sampling, the iterates of V -trace do not admit a uniform upper bound.

4. Conclusion

In this paper, we perform finite-sample analysis of a Markovian SA algorithm under a contractive operator with respect to an arbitrary norm, and derive the convergence rates under different schedules of the step sizes. We develop a Lyapunov approach, and the key technical

novelty is the construction of a valid Lyapunov function called the generalized Moreau envelope, which is capable of handling arbitrary norm (especially the ℓ_∞ -norm) contraction. Our SA results unify the finite-sample analysis of value-based RL algorithms. Specifically, we establish finite-sample convergence guarantees of various TD-learning algorithms (e.g., off-policy V -trace, n -step TD, and TD(λ)) for solving the prediction problem and Q -learning for solving the control problem. In addition, we provide theoretical insights about the efficiency of bootstrapping in on-policy bootstrapped TD and demonstrate a bias-variance tradeoff in off-policy TD.

Acknowledgments

Z. Chen recently moved to Caltech as a postdoctoral fellow in August 2022. This work was done when Z. Chen was affiliated with Georgia Tech. K. Shanmugam recently moved to Google Research India (Bengaluru) in April 2022. This work was done when K. Shanmugam was affiliated with IBM.

References

- Banach S (1922) Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamentals Math.* 3(1):133–181.
- Bansal N, Gupta A (2019) Potential-function proofs for gradient methods. *Theory Comput.* 15(1):1–32.
- Beck A (2017) *First-Order Methods in Optimization* (SIAM, Philadelphia).
- Beck A, Teboulle M (2012) Smoothing and first order methods: A unified framework. *SIAM J. Optim.* 22(2):557–580.
- Beck CL, Srikant R (2013) Improved upper bounds on the expected error in constant step-size Q -learning. *Proc. Amer. Control Conf.* (IEEE, Piscataway, NJ), 1926–1931.
- Benaim M (1996) A dynamical system approach to stochastic approximations. *SIAM J. Control Optim.* 34(2):437–472.
- Benveniste A, Métivier M, Priouret P (2012) *Adaptive Algorithms and Stochastic Approximations*, vol. 22 (Springer Science & Business Media, Boston).
- Bertsekas DP, Tsitsiklis JN (1996) *Neuro-Dynamic Programming* (Athena Scientific, Belmont, MA).
- Bhandari J, Russo D, Singal R (2018) A finite time analysis of temporal difference learning with linear function approximation. *Proc. Conf. on Learning Theory*, 1691–1692.
- Bhatnagar S, Borkar VS (1997) Multiscale stochastic approximation for parametric optimization of hidden Markov models. *Probability Engrg. Inform. Sci.* 11(4):509–522.
- Bhatnagar S, Borkar VS (1998) A two timescale stochastic approximation scheme for simulation-based parametric optimization. *Probability Engrg. Inform. Sci.* 12(4):519–531.
- Borkar VS (2009) *Stochastic Approximation: A Dynamical Systems Viewpoint*, vol. 48 (Springer, Berlin).
- Borkar VS, Meyn SP (2000) The ODE method for convergence of stochastic approximation and reinforcement learning. *SIAM J. Control Optim.* 38(2):447–469.
- Bottou L, Curtis FE, Nocedal J (2018) Optimization methods for large-scale machine learning. *SIAM Rev.* 60(2):223–311.
- Chen Z, Maguluri ST, Shakkottai S, Shanmugam K (2020) Finite-sample analysis of contractive stochastic approximation using smooth convex envelopes. Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, eds. *Advances in Neural Information Processing Systems* (Curran Associates, Inc., Red Hook, NY), 8223–8234.
- Chen Z, Zhang S, Doan TT, Clarke JP, Maguluri ST (2022) Finite-sample analysis of nonlinear stochastic approximation with

- applications in reinforcement learning. *Automatica J. IFAC* 146:110623.
- Dalal G, Szörényi B, Thoppe G, Mannor S (2018) Finite sample analysis for TD(0) with function approximation. *Proc. 32nd AAAI Conf. on Artificial Intelligence* (AAAI, Washington, DC).
- Devraj AM, Meyn S (2017) Zap Q-learning. Guyon I, Von Luxburg U, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, eds. *Adv. Neural Inform. Processing Systems* (Curran Associates, Inc., Red Hook, NY), 2235–2244.
- Doan TT (2021) Finite-time analysis and restarting scheme for linear two-time-scale stochastic approximation. *SIAM J. Control Optim.* 59(4):2798–2819.
- Doan TT (2023) Finite-time analysis of Markov gradient descent. *IEEE Trans. Automated Controls.* 68(4):2140–2153.
- Duchi JC, Agarwal A, Johansson M, Jordan MI (2012) Ergodic mirror descent. *SIAM J. Optim.* 22(4):1549–1578.
- Espeholt L, Soyer H, Munos R, Simonyan K, Mnih V, Ward T, Doron Y, et al. (2018) IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. *Proc. Internat. Conf. on Machine Learn.* (PMLR, New York), 1407–1416.
- Even-Dar E, Mansour Y (2003) Learning rates for Q-learning. *J. Machine Learn. Res.* 5(Dec):1–25.
- Glynn PW, Iglehart DL (1989) Importance sampling for stochastic simulations. *Management Sci.* 35(11):1367–1392.
- Gosavi A (2006) Boundedness of iterates in Q-learning. *Systems Control Lett.* 55(4):347–349.
- Guzmán C, Nemirovski A (2015) On lower complexity bounds for large-scale smooth convex optimization. *J. Complexity* 31(1):1–14.
- Harutyunyan A, Bellemare MG, Stepleton T, Munos R (2016) $Q(\lambda)$ with off-policy corrections. *Proc. Internat. Conf. on Algorithmic Learn. Theory* (Springer, Berlin), 305–320.
- Jaakkola T, Jordan MI, Singh SP (1993) Convergence of stochastic iterative dynamic programming algorithms. Cowan J, Tesauro G, Alspecter J, eds. *Adv. Neural Inform. Processing Systems* (Morgan-Kaufmann, Burlington, MA), 703–710.
- Jin C, Allen-Zhu Z, Bubeck S, Jordan MI (2018) Is Q-learning provably efficient? *Proc. 32nd Internat. Conf. on Neural Inform. Processing Systems* (ACM, New York), 4868–4878.
- Kaledin M, Moulines E, Naumov A, Tadic V, Wai HT (2020) Finite time analysis of linear two-timescale stochastic approximation with Markovian noise. *Proc. Conf. on Learn. Theory* (PMLR, New York), 2144–2203.
- Karmakar P, Bhatnagar S (2021) Stochastic approximation with iterate-dependent Markov noise under verifiable conditions in compact state space with the stability of iterates not ensured. *IEEE Trans. Automated Control.* 66(12):5941–5954.
- Khodadadian S, Chen Z, Maguluri ST (2021) Finite-sample analysis of off-policy natural actor-critic algorithm. *Proc. Internat. Conf. on Machine Learn.* (PMLR, New York), 5420–5431.
- Konda V, Tsitsiklis J (1999) Actor-critic algorithms. Solla S, Leen T, Müller K, eds. *Adv. Neural Inform. Processing Systems* (MIT Press, Cambridge, MA), 1008–1014.
- Kushner H (2010) Stochastic approximation: A survey. *Wiley Interdisciplinary Rev. Comput. Statist.* 2(1):87–96.
- Kushner HJ, Clark DS (2012) *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, vol. 26 (Springer Science & Business Media, Boston).
- Küttler H, Nardelli N, Lavril T, Selvatici M, Sivakumar V, Rocktäschel T, Grefenstette E (2019) Torchbeast: A PyTorch platform for distributed RL. Preprint, submitted October 8, <https://arxiv.org/abs/1910.03552>.
- Lan G (2020) *First-Order and Stochastic Optimization Methods for Machine Learning* (Springer, Berlin).
- Lax P (1997) *Linear Algebra. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts* (Wiley, New York).
- Levin DA, Peres Y (2017) *Markov Chains and Mixing Times*, vol. 107 (American Mathematical Society, Providence, RI).
- Li G, Wei Y, Chi Y, Gu Y, Chen Y (2020) Sample complexity of asynchronous Q-learning: Sharper analysis and variance reduction. *Adv. Neural Inform. Processing Systems* 33:7031–7043.
- Ljung L (1977) Analysis of recursive stochastic algorithms. *IEEE Trans. Automated Control* 22(4):551–575.
- Mirowski P, Grimes M, Malinowski M, Hermann KM, Anderson K, Teplyashin D, Simonyan K, et al. (2018) Learning to navigate in cities without a map. *Proc. 32nd Internat. Conf. Neural Inform. Processing Systems* (Curran Associates, Inc., Red Hook, NY), 2424–2435.
- Moulines E, Bach F (2011) Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Adv. Neural Inform. Processing Systems* 24:451–459.
- Munos R, Stepleton T, Harutyunyan A, Bellemare MG (2016) Safe and efficient off-policy reinforcement learning. *Proc. 30th Internat. Conf. on Neural Inform. Processing Systems*, 1054–1062.
- Nemirovski A, Juditsky A, Lan G, Shapiro A (2009) Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.* 19(4):1574–1609.
- Precup D, Sutton RS, Singh SP (2000) Eligibility traces for off-policy policy evaluation. *Proc. 17th Internat. Conf. on Machine Learn.*, 759–766.
- Qu G, Wierman A (2020) Finite-time analysis of asynchronous stochastic approximation and Q-learning. *Proc. Conf. on Learn. Theory* (PMLR, New York), 3185–3205.
- Robbins H, Monro S (1951) A stochastic approximation method. *Ann. Math. Statist.* 22(3):400–407.
- Ryu EK, Boyd S (2016) Primer on monotone operator methods. *Appl. Comput. Math.* 15(1):3–43.
- Singh SP, Sutton RS (1996) Reinforcement learning with replacing eligibility traces. *Machine Learn.* 22(1):123–158.
- Srikant R, Ying L (2019) Finite-time error bounds for linear stochastic approximation and TD learning. *Proc. Conf. on Learn. Theory* (PMLR, New York), 2803–2830.
- Sutton RS (1988) Learning to predict by the methods of temporal differences. *Machine Learn.* 3(1):9–44.
- Sutton RS (1999) Open theoretical questions in reinforcement learning. *Proc. Eur. Conf. on Comput. Learn. Theory* (Springer, Berlin), 11–17.
- Sutton RS, Barto AG (2018) *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA).
- Szepesvári C, et al. (1997) The asymptotic convergence-rate of Q-learning. Jordan M, Kearns M, Solla S, eds. *Adv. Neural Inform. Processing Systems* (MIT Press, Cambridge, MA), 1064–1070.
- Thoppe G, Borkar V (2019) A concentration bound for stochastic approximation via Alekseev’s formula. *Stochastic Systems* 9(1):1–26.
- Tsitsiklis JN (1994) Asynchronous stochastic approximation and Q-learning. *Machine Learn.* 16(3):185–202.
- Tsitsiklis JN, Van Roy B (1997) Analysis of temporal-difference learning with function approximation. *IEEE Trans. Automatic Control* 42(5):674–690.
- Tsitsiklis JN, Van Roy B (1999) Average cost temporal-difference learning. *Automatica J. IFAC* 35(11):1799–1808.
- Vershynin R (2018) *High-Dimensional Probability: An Introduction with Applications in Data Science*, vol. 47 (Cambridge University Press, Cambridge, UK).
- Wainwright MJ (2019) Stochastic approximation with cone-contractive operators: Sharp ℓ_∞ -bounds for Q-learning. Preprint, submitted May 15, <https://arxiv.org/abs/1905.06265>.
- Watkins C (1989) Learning from delayed rewards. PhD thesis, King’s College, University of Cambridge, Cambridge, UK.
- Watkins CJ, Dayan P (1992) Q-learning. *Machine Learn.* 8(3–4):279–292.
- Yaji VG, Bhatnagar S (2019) Analysis of stochastic approximation schemes with set-valued maps in the absence of a stability guarantee and their stabilization. *IEEE Trans. Automated Control* 65(3):1100–1115.

Zeng S, Doan TT, Romberg J (2021) Finite-time analysis of decentralized stochastic approximation with applications in multi-agent and multi-task learning. *Proc. 60th IEEE Conf. Decision and Control* (IEEE, Piscataway, NJ).

Zaiwei Chen is a postdoctoral researcher in the Computing + Mathematical Sciences Department at Caltech. He is mainly interested in addressing sequential decision-making challenges through reinforcement learning. His Ph.D. thesis received the Georgia Tech Sigma Xi Best Ph.D. Thesis Award and was selected as a runner-up for the 2022 SIGMETRICS Doctoral Dissertation Award.

Siva Theja Maguluri is Fouts Family Early Career Professor and Associate Professor in the H. Milton Stewart School of Industrial and Systems Engineering at Georgia Tech. His research interests span the areas of Networks, Control, Optimization, Algorithms, Applied Probability, and Reinforcement Learning. He is a recipient

of the biennial “Best Publication in Applied Probability” award and the NSF CAREER award.

Sanjay Shakkottai is with The University of Texas at Austin, where he is a Professor in the Chandra Family Department of Electrical and Computer Engineering and holds the Cockrell Family Chair in Engineering \# 15. His research interests lie at the intersection of algorithms for resource allocation, statistical learning, and networks, with applications to wireless communication networks and online platforms. He received the NSF CAREER award in 2004 and was elected as an IEEE Fellow in 2014.

Karthikeyan Shanmugam is currently a senior research scientist at Google Research India in the machine learning and optimization team. His current research focus is on causal inference, online learning, representation learning, and interpretability in machine learning. He is also interested in information theory and coding theory. He is a recipient of the IBM Corporate Technical Award in 2021 for his work on Trustworthy AI.