# A UNIFIED LYAPUNOV FRAMEWORK FOR FINITE-SAMPLE ANALYSIS OF REINFORCEMENT LEARNING ALGORITHMS

A Dissertation
Presented to
The Academic Faculty

By

Zaiwei Chen

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial & Systems Engineering
College of Engineering

Georgia Institute of Technology

May 2022

# A UNIFIED LYAPUNOV FRAMEWORK FOR FINITE-SAMPLE ANALYSIS OF REINFORCEMENT LEARNING ALGORITHMS

Thesis committee:

Dr. Siva Theja Maguluri
Industrial and Systems Engineering
*Georgia Institute of Technology*

Dr. Justin Romberg
Electrical and Computer Engineering
*Georgia Institute of Technology*

Dr. John Paul Clarke
Aerospace Engineering
*The University of Texas at Austin*

Dr. Benjamin Van Roy
Management Science and Engineering
*Stanford University*

Dr. Ashwin Pananjady
Industrial and Systems Engineering and Electrical and Computer Engineering
*Georgia Institute of Technology*

Date approved: April 7th, 2022

May the force be with you.

*Star Wars*

To my family

# ACKNOWLEDGMENTS

First of all, I would like to thank my advisors Dr. Siva Theja Maguluri and Dr. John-Paul Clarke for their mentoring and support through my PhD. Their knowledge, patience, and passion deeply impact me. I will always remember the times we spent together working on challenging research problems and brainstorming about ideas.

Second, I want to thank the rest of my committee members Dr. Ashwin Pananjady, Dr. Justin Romberg, and Dr. Benjamin Van Roy. Their constructive suggestions and insightful comments on my thesis are extremely helpful not only for me to finish my dissertation, but also for my future research career.

Thirdly, I am grateful for having taken many unforgettable classes delivered by knowledgeable professors from the School of Industrial and Systems Engineering, the School of Mathematics, the College of Computing, and the School of Aerospace Engineering. In particular, I thank Dr. Wassim Haddad and Dr. Panagiotis Tsiotras for teaching me control theory, where I first learned the power of the Lyapunov method. I thank Dr. Michael Loss, Dr. Vladimir Koltchinskii, Dr. Christopher Heil, Dr. Christian Houdré, Dr. Lutz Warnke, Dr. Robert Foley, Dr. Siva Theja Maguluri, and Dr. Michael Damron for guiding me into the rigorous but elegant world of mathematics. I thank Dr. Shabbir Ahmed, Dr. Santanu Dey, Dr. Arkadi Nemirovski, Dr. David Goldsman, Dr. Guanghui Lan, and Dr. Alexander Shapiro for helping me develop a solid theoretical foundation in operations research. Lastly, I would like to thank Dr. Jacob Abernethy and Dr. Zsolt Kira for exposing me into the area of machine learning and artificial intelligence.

Last but not least, I want to thank my parents, my wife, and my friends for their help and support. In particular, I would not have made it this far if not for my wife Ci's unconditional love and care.

## TABLE OF CONTENTS

## III RL with Linear Function Approximation 165

# LIST OF TABLES

# LIST OF FIGURES

# SUMMARY

Reinforcement learning is a framework for solving sequential decision-making problems without requiring the environmental model, and is viewed as a promising approach to achieve artificial intelligence. However, there is a huge gap between the empirical successes and the theoretical understanding of reinforcement learning. In this thesis, we make an effort to bridging such gap.

More formally, this thesis focuses on designing data-efficient reinforcement learning algorithms and establishing their finite-sample guarantees. Specifically, we aim at answering the following question: suppose we carry out some reinforcement learning algorithm with finite amount of samples (or with finite number of iterations), then what can we say about the performance of the output of the algorithm? The more detailed motivation and the research background are presented in Chapter 1.

**Part I: Stochastic Approximation.** The main body of this thesis is divided into three parts. In the first part of the thesis, we focus on studying the stochastic approximation method. Stochastic approximation is the major workhorse for large-scale optimization and machine learning, and is widely used in reinforcement learning for both algorithm design and algorithm analysis. Therefore, understanding the behavior of SA algorithms is of fundamental interest to the analysis of RL algorithms.

In Chapter 2 and Chapter 3, we consider Markovian stochastic approximation under a contractive operator and under a strongly pseudo-monotone operator, and establish their finite-sample guarantees. These two results on stochastic approximation are used in later parts of the thesis to study reinforcement learning algorithms with a tabular representation and with linear function approximation. The main technique we use to analyze those stochastic approximation algorithms is the Lyapunov-drift method. Specifically, we construct *novel* Lyapunov functions (e.g., generalized Moreau envelope in the case of stochastic approximation under a contraction assumption) to capture the dynamics of the corre-

sponding stochastic approximation algorithms, and control the discretization error and the stochastic error. This enables us to derive the one-step drift inequality, which can be repeatedly used to establish the finite-sample bounds.

In Chapter 4, we switch our focus from finite-sample analysis to asymptotic analysis, and characterize the stationary distribution of the centered-scaled iterates of several popular stochastic approximation algorithms. Specifically, we show that for stochastic gradient descent, linear stochastic approximation, and contractive stochastic approximation, the stationary distribution of the centered iterates (after proper scaling) is a Gaussian distribution with mean zero and a covariance matrix being the unique solution of an appropriate Lyapunov equation. For stochastic approximation beyond these three types, we numerically demonstrate that the stationary distribution may not be Gaussian in general. The main technique we used for such asymptotic analysis is also Lyapunov method, where the characteristic function was used as the test function.

**Part II: Reinforcement Learning with a Tabular Representation.** In the second part of this thesis, we focus on reinforcement learning with a tabular representation. The preliminaries of reinforcement learning are presented in Chapter 5.

In Chapter 6 and Chapter 7, we consider the TD-learning algorithm for solving the policy evaluation problem, which refers to the problem of estimating the performance of a given policy. Solving the policy evaluation problem is an important intermediate step in the popular actor-critic framework for ultimately finding an optimal policy. More specifically, we consider on-policy TD-learning algorithms such as $n$-step TD and TD$(\lambda)$ in Chapter 6. By establishing finite-sample guarantees of $n$-step TD and TD$(\lambda)$ as explicit functions of the parameters $n$ and $\lambda$, we provide theoretical insight into the open problem about the efficiency of bootstrapping, which is about how to choose the parameters $n$ and $\lambda$ so that $n$-step TD and TD$(\lambda)$ achieve their best performance.

In Chapter 7, we study the problem of policy evaluation using off-policy sampling, where the policy used to collect samples and the policy whose value function we aim at estimating is different. We provide finite-sample analysis of a generic off-policy multi-step TD-learning algorithm, which subsumes several popular existing algorithms such as $Q^\pi(\lambda)$, Tree-Backup$(\lambda)$, Retrace$(\lambda)$, and $V$-trace as its special cases. In addition, our finite-sample bounds demonstrate a trade-off between the variance (which arises due to the product of the importance sampling ratios) and the bias in the limit point (which arises due to various modifications to the importance sampling ratios). Understanding such bias-variance trade-off is at the heart of off-policy learning.

In Chapter 8, we consider the $Q$-learning algorithm for directly finding an optimal policy and present its finite-sample guarantees. The finite-sample bounds imply an $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity, which is known to be optimal up to a logarithmic factor. In addition, our finite-sample bounds also capture the dependence on other importance parameters of the reinforcement learning problem, such as the size of the state-action space and the effective horizon.

**Part III: Reinforcement Learning with Linear Function Approximation.** In the last part of this thesis, to overcome the curse of dimensionality in reinforcement learning, we consider reinforcement learning with linear function approximation. Specifically, we focus on the off-policy setting, where the deadly triad is present, and can result in instability of reinforcement learning algorithms.

In Chapter 9, we consider off-policy TD-learning with linear function approximation, where the deadly triad appears. We design a single time-scale off-policy TD-learning using generalized importance sampling ratios and multi-step bootstrapping, and establish its finite-sample guarantees. The algorithm is provably convergent in the presence of the deadly triad, and does not suffer from the high variance in existing off-policy learning algorithms.

The TD-learning algorithm proposed in Chapter 9 is later used in Chapter 10 to solve the policy evaluation sub-problem in the general policy-based framework with various policy update rules, including approximate policy iteration and natural policy gradient. By only exploiting the contraction property and the monotonicity property of the Bellman operator, we establish an overall $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity for a wide class of policy-based methods using off-policy sampling and linear function approximation.

In Chapter 11, we focus on $Q$-learning with linear function approximation (where the deadly triad naturally appears), and establish its finite-sample bounds under an assumption on the discount factor of the problem. In particular, we show that when the discount factor is sufficiently small, the deadly triad challenge can be overcome.

In Chapter 12, we further remove the restriction on the discount factor by designing a convergent variant of $Q$-learning with linear function approximation using target network and truncation. This is the first variant of $Q$-learning with linear function approximation that uses a single trajectory of Markovian samples, and is provably stable without requiring strong assumptions. In addition, the algorithm achieves the optimal $\mathcal{O}(\epsilon^{-2})$ sample complexity (which matches with $Q$-learning in the tabular setting) up to a function approximation error.

# CHAPTER 1
# INTRODUCTION AND OVERVIEW

## 1.1 Introduction

Reinforcement learning (RL) is a paradigm where an agent learns to accomplish tasks by interacting with the environment, similar to how humans learn [1]. RL is therefore viewed as a promising approach to achieve artificial intelligence, as evidenced by the remarkable successes in solving many practical problems such as the game of Go [2], robotics [3], autonomous-driving [4], healthcare [5], and very recently, controlling the nuclear fusion plasma [6].

Despite the empirical successes, theoretically RL algorithms are in general not well-understood. A typical example is $Q$-learning with function approximation, which although achieves remarkable performance in practical applications [7], is theoretically known to diverge in general [8]. The focus of this thesis is on developing data-efficient RL algorithms with provable finite-sample guarantees. More formally, let $\{x_k\}$ be the iterates generated by some iterative RL algorithm. The goal is to study the decay of difference between $x_k$ and $x^*$ (which is the desired limit point) as a function of the number of iterations $k$. Such results not only provide theoretical insights into RL algorithms, but also can be used as guidelines for both practical algorithm implementation and new algorithm design.

RL has three major ingredients, viz., Markov decision process (MDP), stochastic approximation (SA), and function approximation . The RL problem is usually modeled as an MDP. However, the environmental model of the MDP (e.g. transition probabilities, reward function, etc) is unknown to the agent. In each time step, the agent is at some state, and can take an action to determine the next state, as well as the stage-wise rewards, in a stochastic manner. The goal is to find an optimal policy of choosing actions so that its corresponding long-term reward is maximized.

It turns out that solving the RL problem is equivalent to solving a system of equations known as the Bellman equation [9], which leads to several popular iterative algorithms, including but not limited to value iteration, policy iteration, and policy gradient. However, since the environmental model is unknown in RL, one needs to work with iterative algorithms in the presence of noise. This leads to the SA method [10]. In fact, most of the RL algorithms can be modeled by SA algorithms for solving some suitable target equations. Beyond RL, SA algorithms are used widely in other aspects of machine learning and optimization, with the popular stochastic gradient descent (SGD) being a typical example. Therefore, studying the behavior of general SA algorithms is of fundamental interest.

A major challenge of RL in implementation is that most of the classical algorithms are not tractable when facing large state and action spaces. To overcome this difficulty, RL algorithms are incorporated with function approximation, where the main idea is to restrict the searching space to a pre-defined subset, thereby reducing the complexity of the problem. Although this idea has led to many empirical successes, function approximation is one the infamous "deadly triad" [1] (the other two are off-policy learning and bootstrapping), which can result in divergence [8]. This motivates us to design algorithms that have provable convergence bounds when facing the "deadly triad".

## 1.2 Overview of Main Contributions

In this section we present a high-level overview of the main contributions.

### 1.2.1 Stochastic Approximation

As we mentioned earlier, RL algorithms in their nature are stochastic iterative algorithms for solving various Bellman equations. Due to the special sampling procedure in RL, the algorithms usually involve Markovian noise. In Part I of this thesis, we focus on studying Markovian SA algorithms. The results serve as the major theoretical workhorse for our

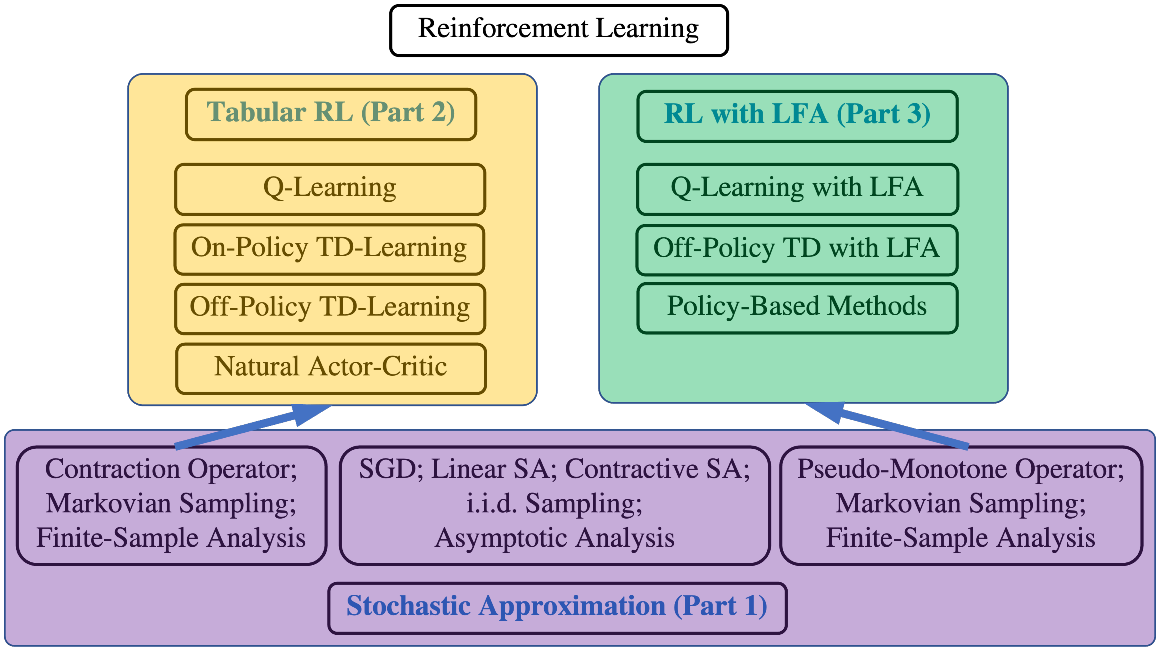
Reinforcement Learning
Tabular RL (Part 2)
Q-Learning
On-Policy TD-Learning
Off-Policy TD-Learning
Natural Actor-Critic
RL with LFA (Part 3)
Q-Learning with LFA
Off-Policy TD with LFA
Policy-Based Methods
Contraction Operator; Markovian Sampling; Finite-Sample Analysis
SGD; Linear SA; Contractive SA; i.i.d. Sampling; Asymptotic Analysis
Pseudo-Monotone Operator; Markovian Sampling; Finite-Sample Analysis
Stochastic Approximation (Part 1)

Figure 1.1: Summary of My Work

analysis of RL algorithms in Part II and Part III.

In Chapter 2 and Chapter 3, we focus on finite-sample analysis of Markovian SA algorithms. To provide a unified framework, we develop a *Lyapunov approach*, which involves two major challenges. One is the contruction of a valid Lyapunov function to capture the dynamics of the corresponding SA algorithm, and the other is to handle the stochastic error due to the Markovian noise.

In Chapter 2, we consider SA algorithms under contractive operators, where we construct a novel Lyapunov function called the *generalized Moreau envelope*. Previously, the lack of a Lyapunov function for studying contractive SA algorithms imposes major difficulties in the analysis [11, Section 4.3]. We overcome this challenge by constructing the generalized Moreau envelope, which serves as a valid Lyapunov function for SA algorithms under arbitrary norm contraction. In Chapter 3, we consider SA algorithms involving strongly pseudo-monotone operators, where the Euclidean norm-square function serves as a valid Lyapunov function.

To handle the Markovian noise for both contractive SA algorithms and SA algorithms under strongly pseudo-monotone operators, we use a conditioning argument together with the fast mixing of Markov chains. Such conditioning argument was first used in [11] for studying the asymptotic convergence of linear SA algorithms with Markovian noise. Later, it was used more explicitly in [12] for deriving finite-sample bounds of linear SA algorithms. In this thesis, we extend this technique to studying nonlinear Markovian SA algorithms, whose analysis is fundamentally more challenging.

Beyond finite-sample analysis, we also provide asymptotic analysis of SA algorithms in terms of the stationary distribution of the centered-scaled iterates in Chapter 4. Specifically, we show that for SGD, linear SA, and contractive SA, the corresponding stationary distribution is a Gaussian distribution with mean zero and a covariance matrix being the unique solution of an appropriate Lyapunov equation. We also adopt a Lyapunov approach here to establish the results, where the characteristic function serves as a test function. For more general SA algorithms, we show numerically that unlike central limit theorem type of results, the stationary distribution need not be Gaussian in general.

### 1.2.2 Reinforcement Learning with a Tabular Representation

In the second part of this thesis, we provide finite-sample guarantees of various tabular RL algorithms including on-policy TD-learning algorithms such as $n$-step TD and TD($\lambda$), off-policy TD-learning algorithms such as $Q^\pi(\lambda)$ [13], Tree-Backup($\lambda$) (henceforth denoted by TB($\lambda$)) [14], Retrace($\lambda$) [15], and $Q$-trace [16], etc, and off-policy control algorithms such as $Q$-learning [17].

*On-Policy TD-Learning:* For various on-policy bootstrapped TD-learning algorithms such as $n$-step TD and TD($\lambda$), there is key problem about the efficiency of bootstrapping [18], which refers to the question about how to choose the parameters $n$ (or $\lambda$) so that $n$-step TD (or TD($\lambda$)) achieves its optimal performance. By establishing finite-sample bounds of $n$-step TD (and TD($\lambda$)) as an explicit function of $n$ (and $\lambda$), we provide theoretical insights into the efficiency of bootstrapping. For example, the optimal choice of $n$ in $n$-step TD is roughly of the size $\mathcal{O}(1/\log(1/\gamma))$, where $\gamma$ is the discount factor of the RL problem.

*Off-Policy TD-Learning:* When the policy used to collect samples is different than the policy whose value function we want to estimate, the corresponding TD-learning algorithm is called off-policy TD-learning. Off-policy learning is sometimes preferred in practice due to both practical reasons and theoretical reasons. However, it is more difficult to analyze compared to on-policy learning algorithms. By identifying a generalized Bellman operator in off-policy TD-learning and investigating its contraction properties, we provide finite-sample guarantees for a variety class of multi-step off-policy TD-learning algorithms and compare their performance analytically. In addition, our results explicitly capture the trade-offs between the high variance (due to importance sampling ratios) and the bias in the limit point, which is a fundamental problem in off-policy TD-learning algorithms.

*$Q$-Learning:* Since $Q$-learning is the most well-known value-based RL algorithm, its behavior is of fundamental interest to the community. Our finite-sample bounds imply an $\tilde{\mathcal{O}}\left(\frac{(|\mathcal{S}||\mathcal{A}|)^3}{(1-\gamma)^5\epsilon^2}\right)$ sample complexity to achieve $\mathbb{E}[\|Q_k - Q^*\|_\infty] \leq \epsilon$, where $\epsilon$ is a given accuracy. This is the state-of-the-art mean-square sample complexity of $Q$-learning.

### 1.2.3 Reinforcement Learning with Linear Function Approximation

In the last part of the thesis, we consider RL with linear function approximation. In reality, RL algorithms usually face computational challenges when the size of the state-action space is large. This motivates the use of function approximation. However, when function approximation is used together with off-policy sampling, the infamous deadly triad [1] usually appears and the corresponding RL algorithm can diverge [8]. In Part III of this thesis, we design convergent RL algorithms in the presence of the deadly triad and establish their finite-sample guarantees.

*Off-Policy TD-Learning with Linear Function Approximation:* In Chapter 9, to overcome the deadly triad in TD-learning, we propose a generic single time-scale algorithm of multi-step TD-learning with generalized importance sampling ratios, including two specific algorithms: the $\lambda$-averaged $Q$-trace algorithm and the two-sided $Q$-trace algorithm. We establish their finite-sample convergence guarantees, characterize the limit points as solutions to generalized multi-step projected Bellman equations (PBEs), and provide performance bounds on the limit points in terms of the error compared to the true value functions.

*Policy-Based Algorithms under Off-policy Sampling and Linear Function Approximation:* In Chapter 10, we consider a general policy-based framework where the policy evaluation problem is solved with our proposed off-policy TD-learning algorithm (presented in Chapter 9), and the policy improvement uses various policy update rules, including approximate policy iteration and natural policy gradient. We provide a unified approach to show that the overall sample complexity for all these algorithms is $\tilde{\mathcal{O}}(\epsilon^{-2})$, which matches with the sample complexity of value-based RL algorithms such as $Q$-learning. Importantly, to establish the results, we only exploit the contraction property and the monotonicity property of the Bellman operators.

*$Q$-Learning with Linear Function Approximation:* $Q$-learning with function approximation is one of the most empirically successful while theoretically mysterious RL algorithms, and was identified in [18] as one of the most important theoretical open problems in the RL community. Even in the basic linear function approximation setting, there are well-known divergent examples [8].

In Chapter 11, we provide sufficient conditions under which $Q$-learning with linear function approximation provably converges. In Chapter 12, we further propose a stable design for $Q$-learning with linear function approximation using *target network* and *truncation*, and establish its $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity up to a function approximation error. This is the first variant of $Q$-learning with linear function approximation that uses a single trajectory of Markovian samples, and is provably stable without requiring strong assumptions or modifying the problem parameters.

# Part I

# Stochastic Approximation

# CHAPTER 2
# STOCHASTIC APPROXIMATION UNDER A CONTRACTIVE OPERATOR

## 2.1 Introduction

Solving optimization or machine learning problems usually reduces to solving root-finding problems. For example, minimizing a convex objective function is equivalent to finding the zeros of its gradient operator. Similarly in RL, finding an optimal policy essentially boils down to solving the Bellman equation.

To solve systems of equations, we usually resort to iterative algorithms. As we will see in the second part of this thesis, at the heart of RL is the problem of iteratively solving the Bellman equation using noisy samples, i.e. solving a fixed-point equation of the form $\bar{F}(x) = x$. Here, $\bar{F}(\cdot)$ is a contractive operator with respect to a suitable norm, where we only have access to samples from noisy versions of the operator. Such fixed-point equations, more broadly, are solved through the framework of SA algorithms [10], with several RL algorithms such as $Q$-learning and TD-learning being examples there-of. This chapter focuses on understanding the evolution of such a noisy fixed-point iteration through the lens of SA, and providing finite-sample convergence results.

More formally, motivated by applications in RL, we consider an SA algorithm of the following form.

**Algorithm 1** SA under a Contractive Operator

```
Input: Integer $k'$, and initialization $x_0 \in \mathbb{R}^d$
for $k = 0, 1, \cdots, k' - 1$ do
    $x_{k+1} = x_k + \alpha_k \left(F(x_k, Y_k) - x_k + w_k\right)$
end for
Output: $x_{k'}$
```

Here in Algorithm 1, $\{\alpha_k\}$ is a sequence of stepsizes, $\{Y_k\}$ is a Markov chain with a finite state-space $\mathcal{Y}$ and a unique stationary distribution $\mu_Y$, $F : \mathbb{R}^d \times \mathcal{Y} \mapsto \mathbb{R}^d$ is a nonlinear operator, and $\{w_k\}$ is a random process representing the additive extraneous noise. Let $\bar{F}(\cdot) = \mathbb{E}_{Y\sim\mu_Y}[F(\cdot, Y)]$, and we assume that $\bar{F}(\cdot)$ is a contraction mapping with respect to some arbitrary norm, denoted by $\|\cdot\|_c$. By rewriting the main update equation of Algorithm 1 as

$$x_{k+1} - x_k = \underbrace{\alpha_k(\bar{F}(x_k) - x_k)}_{\text{Expected Update}} + \underbrace{\alpha_k(F(x_k, Y_k) - \bar{F}(x_k))}_{\text{Markovian Noise}} + \underbrace{\alpha_k w_k}_{\text{Martingale Difference Noise}}, \tag{2.1}$$

we see that Algorithm 1 is a stochastic variant of the fixed-point iteration (with stepsizes) $x_{k+1} = (1-\alpha_k)x_k + \alpha_k \bar{F}(x_k)$, and hence is an SA algorithm for solving the fixed-point equation

$$\bar{F}(x) = x. \tag{2.2}$$

Our goal is to characterize the behavior of the quantity $\mathbb{E}[\|x_k - x^*\|_c^2]$ as a function of $k$, where $x^*$ is the unique solution to Equation 2.2.

To derive finite-sample bounds, two conditions are pertinent: (1) the norm in which the operator $\bar{F}(\cdot)$ contracts, and (2) the properties of the effective noise, i.e., $N_k := F(x_k, Y_k) - \bar{F}(x_k) + w_k$ in the case of Equation 2.1. In prior literature, if the conditional second moment of the noise $\{N_k\}$ is uniformly bounded by a constant, then the norm with respect to which $\bar{F}(\cdot)$ being a contraction becomes irrelevant, and it is possible to derive finite-sample convergence guarantees [19, 20, 21, 22]. When the second moment of the noise is not uniformly bounded, then finite-sample bounds can be derived in the case where the norm for contraction of $\bar{F}(\cdot)$ is the Euclidean norm [11, 23]. However, in many RL problems, the contraction of $\bar{F}(\cdot)$ occurs with respect to a different norm (e.g. the $\ell_\infty$-norm [17] or a weighted variant [24]). Further, due to the Markovian sampling in RL, conditioned on the past, the second moment of the norm of the noise scales affinely with the current iterate, and in general, no uniform bound exists.

An important practical application of this setting with $\ell_\infty$-norm contraction and unbounded noise is the well-known $V$-trace algorithm for solving the policy evaluation problem using off-policy TD-learning [1]. Its variants form the basis of today's distributed RL platforms like IMPALA [25] and TorchBeast [26] for multi-agent training. It has been used at scale in the recent Deepmind City Navigation Project "Street Learn" [27]. Therefore, deriving finite-sample convergence results for SA under contraction of $\bar{F}(\cdot)$ with respect to general norms, and handling unbounded noise are of fundamental interest. In this chapter, we answer the following general question in the affirmative:

*Can we provide finite-sample convergence guarantees for the SA algorithm when the norm of contraction of $\bar{F}(\cdot)$ is arbitrary, and the second moment of the effective noise conditioned on the past scales affinely with respect to the squared-norm of the current iterate?*

To the best of our knowledge, except under special conditions on the norm for contraction of $\bar{F}(\cdot)$ and/or strong assumptions on the noise, such finite-sample error bounds have not been established.

### 2.1.1 Main Contributions

We establish finite-sample guarantees (with various choices of stepsizes) of Algorithm 1. Specifically, we show that when using constant stepsize $\alpha_k \equiv \alpha$, the convergence rate is geometric, with asymptotic accuracy approximately $\mathcal{O}(\alpha \log(1/\alpha))$. When using diminishing stepsizes of the form $\alpha/(k+h)^\xi$ (where $\xi \in (0,1]$), the convergence rate is $\mathcal{O}(\log(k)/k^\xi)$, provided that $\alpha$ and $h$ are appropriately chosen. In addition, our bound also involves a (possibly dimension-dependent) constant that is determined by the contraction norm. In the special case of $\ell_\infty$-norm contraction, we show that such constant scales only logarithmically in terms of the dimension of the iterates, and is not improvable in general.

The key idea is to study the drift of a carefully constructed potential/Lyapunov function. We obtain such a potential function by smoothing the norm-squared function, and the resulting valid Lyapunov function is called the generalized Moreau envelope.

### 2.1.2 Related Literature

The SA method, originally proposed in [10], is an iterative method for solving root-finding problems with incomplete information. Early literature focus on asymptotic convergence of SA algorithms [28]. In particular, for SA under a contractive operator, its asymptotic convergence was established in [24, 29] using a supermnartingale convergence approach, and in [28, 30, 31] using an ODE approach. Specifically, given certain assumptions, it was shown in [32, 33] that the SA algorithm converges almost surely as long as the corresponding ODE is stable. The ODE approach was extended to more general cases in [34, 35, 36], where the ODE lacks stability, or has multiple equilibrium points. The convergence of various SA algorithms such as SA with Markovian noise and multiple time-scale SA was studied in [37, 36] and [38, 39] respectively. While the results presented were very general, they study SA algorithms in the asymptotic regime. In this part of the thesis, we perform finite-sample analysis, which is different in flavor and provides stronger finite-sample convergence guarantees.

For linear SA algorithms, finite-sample mean-square bounds were established for both i.i.d. sampling and Markovian sampling in [40, 12]. Concentration results were established in [41, 42]. For non-linear SA algorithms, finite-sample bounds in general are only derived for special forms of SA algorithms, such as SGD [23, 43, 44], and $Q$-learning. We will present a thorough literature review on $Q$-learning in the second part of this thesis. Moreover, unlike i.i.d. sampling, in the case of Markovian sampling, an artificial projection (onto a ball) is introduced in the algorithm to ensure that the iterates are bounded [45].

### 2.1.3 Summary of Our Techniques

We now give a more detailed description of the techniques we use. To provide intuition, assume for now that the norm with respect to which $\bar{F}(\cdot)$ being a contraction is the $\ell_p$-norm for some $p \in [2, \infty)$. Consider the ODE associated with the SA: $\dot{x}(t) = \bar{F}(x(t)) - x(t)$. It is shown in [33, chapter 10] that the function $W(x) = \|x - x^*\|_p$ satisfies $\frac{d}{dt}W(x(t)) \leq$

$-\kappa W(x(t))$ for some $\kappa > 0$, which implies the solution $x(t)$ of the ODE converges to its equilibrium point $x^*$ geometrically fast. The term $\kappa$ corresponds to a *negative drift*.

In order to obtain finite-sample bounds, in this chapter we study the SA directly, and not the ODE. Then, the Lyapunov function $W(x)$ cannot be directly used to analyze the SA algorithm due to the discretization error and stochastic error. However, suppose we can find a function $M(x)$ that gives negative drift, and is $L$ – smooth, where $L > 0$ is the smoothness parameter. Then, we have a handle to deal with the discretization error and the error caused by the noise to obtain:

$$\mathbb{E}[M(x_{k+1} - x^*)] \leq (1 - \mathcal{O}(\alpha_k) + o(\alpha_k))\mathbb{E}[M(x_k - x^*)] + o(\alpha_k), \tag{2.3}$$

which implies a contraction in $\mathbb{E}[M(x_{k+1} - x^*)]$. Therefore, a finite-sample error 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 above analysis, we see that the Lyapunov function for the SA in the case of $\ell_p$-norm contraction should be $M(x) = \frac{1}{2}\|x - x^*\|_p^2$, which is known to be smooth [46].

However, in the case where the contraction norm $\|\cdot\|_c$ is arbitrary, since 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. An important special case is when $\|\cdot\|_c = \|\cdot\|_\infty$, which is applicable to many RL algorithms. We provide a solution to this where we construct a smoothed convex envelope $M(x)$ called the *generalized Moreau envelope* that is smooth with respect to some norm $\|\cdot\|_s$, and it is a tight approximation to $f(x)$, which essentially guarantees that it is a Lyapunov function for the ODE with a negative drift. This lets us prove a convergence result akin to the case when $f(x)$ is smooth.

## 2.2 Assumptions

In this section, we formally state our assumptions for studying Algorithm 1.

**Assumption 2.2.1** (Contraction Mapping)**.** The operator $\bar{F}(\cdot)$ is a contraction mapping with respect to some arbitrary norm $\|\cdot\|_c$, i.e., there exists $\beta \in (0,1)$ such that

$$\|\bar{F}(x_1) - \bar{F}(x_2)\|_c \leq \beta \|x_1 - x_2\|_c, \ \forall\, x_1, x_2 \in \mathbb{R}^d.$$

Under Assumption 2.2.1, Equation 2.2 has a unique solution [47], which we have denoted by $x^*$.

**Assumption 2.2.2** (Lipschitz continuity)**.** There exist $A_1, B_1 > 0$ such that

(1) $\|F(x_1, y) - F(x_2, y)\|_c \leq A_1 \|x_1 - x_2\|_c$ for any $x_1, x_2 \in \mathbb{R}^d$ and $y \in \mathcal{Y}$,

(2) $\|F(\mathbf{0}, y)\|_c \leq B_1$ for any $y \in \mathcal{Y}$.

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

**Assumption 2.2.3** (Uniform Ergodicity)**.** The Markov chain $\mathcal{M} = \{Y_k\}$ has a unique stationary distribution $\mu_Y$, and there exist $C > 0$ and $\sigma \in (0,1)$ such that

$$\max_{y \in \mathcal{Y}} \|P^k(y, \cdot) - \mu_Y(\cdot)\|_{\text{TV}} \leq C\sigma^k, \ \forall\, k \geq 0.$$

*Remark.* Since the state-space $\mathcal{Y}$ of the Markov chain $\{Y_k\}$ is finite, Assumption 2.2.3 is satisfied when $\{Y_k\}$ is irreducible and aperiodic [48].

Under Assumption 2.2.3, we next introduce the notion of Markov chain mixing.

**Definition 2.2.1.** For any $\delta > 0$, the mixing time $t_\delta(\mathcal{M})$ of the Markov chain $\mathcal{M} = \{Y_k\}$ with precision $\delta$ is defined by

$$t_\delta(\mathcal{M}) = \min\left\{k \geq 0 : \max_{y \in \mathcal{Y}} \|P^k(y, \cdot) - \mu_Y(\cdot)\|_{\text{TV}} \leq \delta\right\}.$$

For simplicity of notation, in this chapter of the thesis, we will just write $t_\delta$ for $t_\delta(\mathcal{M})$. Note that Assumption 2.2.3 implies $t_\delta \leq \frac{\log(C/\sigma)+\log(1/\delta)}{\log(1/\sigma)}$ for any $\delta > 0$, which further implies that $\lim_{\delta\to 0} \delta t_\delta = 0$. This property is important in our analysis for controlling the Markovian noise $\{Y_k\}$ in Algorithm 1.

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

**Assumption 2.2.4** (Additive Martingale Difference Noise)**.** There exist $A_2, B_2 > 0$ such that

(1) $\mathbb{E}[w_k \mid \mathcal{F}_k] = 0$ for all $k \geq 0$,

(2) $\|w_k\|_c \leq A_2\|x_k\|_c + B_2$ for all $k \geq 0$.

Assumption 2.2.4 states that $\{w_k\}$ is a martingale difference sequence with respect to the filtration $\mathcal{F}_k$, and can grow at most affinely with respect to the iterate $x_k$.

Finally, we specify the requirements for choosing the stepsize sequence $\{\alpha_k\}$. We will consider using stepsizes of the form $\alpha_k = \frac{\alpha}{(k+h)^\xi}$, where $\alpha, h > 0$ and $\xi \in [0, 1]$.

**Condition 2.2.1.** *(1) Constant Stepsize.* When $\xi = 0$, there exists a threshold $\bar{\alpha} \in (0, 1)$ such that we need to choose $\alpha \in (0, \bar{\alpha})$. *(2) Linear Stepsize.* When $\xi = 1$, for each $\alpha > 0$, there exists a threshold $\bar{h} > 0$ such that we need to choose $h \in [\bar{h}, \infty)$. *(3) Polynomial Stepsize.* For any $\xi \in (0, 1)$ and $\alpha > 0$, there exists a threshold $\bar{h} > 0$ such that we need to choose $h \in [\bar{h}, \infty)$.

The existence of the thresholds $\bar{\alpha}$ and $\bar{h}$ is verified in Subsection 2.6.3.

The asymptotic convergence of $\{x_k\}$ under similar assumptions has been established in the literature. In particular, an approach based on studying the ODE

$$\dot{x}(t) = \bar{F}(x(t)) - x(t) \tag{2.4}$$

was used in [31, 33], where it was shown that $x_k$ converges to $x^*$ almost surely under some stability assumptions of the ODE. The focus of this chapter is to establish the finite-sample bounds for Algorithm 1. We do this by studying the drift of a smooth potential/Lyapunov function [12, 49]. While we do not explicitly use the ODE approach, the potential function we are going to construct in Section 2.3 is inspired by the Lyapunov function used to study the ODE.

## 2.3 The Generalized Moreau Envelope as A Smooth Lyapunov Function

In this section, we construct a novel Lyapunov function through the generalized Moreau envelope, and investigate its properties. In particular, the smoothness and an approximation property of the Lyapunov function we specify here are used in the next subsection to show the desired recursive contractive bound of Algorithm 1 (i.e., Equation 2.3).

To construct such a Lyapunov function, the following definitions are needed. In this thesis, $\langle x, y\rangle = x^\top y$ represents the standard dot product, while $\|\cdot\|$ 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 2.3.1.** Let $g : \mathbb{R}^d \to \mathbb{R}$ be a convex differentiable function. Then $g(\cdot)$ is said to be $L$ – smooth with respect to some norm $\|\cdot\|$ if and only if

$$g(y) \leq g(x) + \langle \nabla g(x), y - x\rangle + \frac{L}{2}\|x - y\|^2,\ \forall\, x, y \in \mathbb{R}^d.$$

**Definition 2.3.2** (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 by

$$M_{h_1}^{\theta,h_2}(x) = \min_{u\in\mathbb{R}^d}\left\{h_1(u) + \frac{1}{\theta}h_2(x-u)\right\}.$$

The standard Moreau envelope was previously used in [50, 51] to study convex optimization problems. As an aside, we note that for any two functions $h_1, h_2 : \mathbb{R}^d \mapsto \mathbb{R}$, the function defined by $(h_1 \Box h_2)(x) := \inf_{u\in\mathbb{R}^d}\{h_1(u) + h_2(x-u)\}$ is called the infimal convolution of $h_1(\cdot)$ and $h_2(\cdot)$ [46]. Therefore, the generalized Moreau envelope in Definition 2.3.2 can be written as $M_{h_1}^{\theta,h_2}(x) = (h_1 \Box \frac{h_2}{\theta})(x)$.

Let $f(x) = \frac{1}{2}\|x\|_c^2$, where $\|\cdot\|_c$ is given in Assumption 2.2.1. 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 $\ell_p$-norm for any $p \in [2,\infty)$ [46, Example 5.11]. Due to the norm equivalence in $\mathbb{R}^d$ [52], 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 $\ell_{cs}\|\cdot\|_s \leq \|\cdot\|_c \leq u_{cs}\|\cdot\|_s$.

**Construction of the Lyapunov Function.** With a suitable choice of $\theta$, we will use the generalized Moreau envelope of $f(\cdot)$ with respect to $g(\cdot)$, i.e., $M_f^{\theta,g}(\cdot)$ as our Lyapunov function to analyze the behavior of Algorithm 1. The following proposition states that $M_f^{\theta,g}(\cdot)$ is a *smooth approximation* of the norm-squared function $f(\cdot)$.

**Proposition 2.3.1.** *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 $M_f^{\theta,g}(x) = \frac{1}{2}\|x\|_m^2$.*

*(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 2.3.1 (1) is restated from [46], and we include it here for completeness. This, together with Proposition 2.3.1 (3) implies that $M_f^{\theta,g}(\cdot)$ is a smooth approximation of the norm-squared function $f(\cdot)$. Intuitively, suppose that the $f(\cdot)$ itself is smooth, then $f(\cdot)$ can be directly used as a Lyapunov function to study Algorithm 1. However, for an arbitrary contraction norm $\|\cdot\|_c$, the function $f(\cdot)$ is not necessarily smooth. One typical example is when $\|\cdot\|_c = \|\cdot\|_\infty$. In this case, we use the generalized Moreau envelope to construct $M_f^{\theta,g}(\cdot)$ as a smooth approximation of $f(\cdot)$. Proposition 2.3.1 (2) states that the generalized Moreau envelope itself is also a norm-squared function.

## 2.4 Recursive Contractive Bounds for the Generalized Moreau Envelope

In this section, using the smooth approximation property of the generalized Moreau envelope $M_f^{\theta,g}(\cdot)$, we establish the desired one-step contractive inequality of $M_f^{\theta,g}(x_k - x^*)$.

Let $\varphi_1 = \frac{1+\theta u_{cs}^2}{1+\theta \ell_{cs}^2}$, $\varphi_2 = 1 - \beta\varphi_1^{1/2}$, and $\varphi_3 = \frac{114L(1+\theta u_{cs}^2)}{\theta \ell_{cs}^2}$. The tunable parameter $\theta$ is chosen such that $\varphi_2 > 0$, which is always possible since $\lim_{\theta\to 0}\varphi_1 = 1$ and $\beta \in (0,1)$. Let $t_k$ be the mixing time of the Markov chain $\{Y_k\}$ with precision $\alpha_k$ (see Definition 2.2.1). For simplicity of notation, denote $\alpha_{i,j} = \sum_{k=i}^{j}\alpha_k$ for any $i \leq j$ and $\hat{\alpha}_k = \alpha_k\alpha_{k-t_k,k-1}$ for all $k \geq t_k$.

**Proposition 2.4.1.** *The following inequality holds for all $k \geq t_k$:*

$$\mathbb{E}[M_f^{\theta,g}(x_{k+1} - x^*)] \leq \left(1 - 2\varphi_2\alpha_k + \varphi_3 A^2\hat{\alpha}_k\right)\mathbb{E}[M_f^{\theta,g}(x_k - x^*)] + \frac{\varphi_3\hat{\alpha}_k}{2u_{cm}^2}(A\|x^*\|_c + B)^2, \tag{2.5}$$

*where $A = A_1 + A_2 + 1$ and $B = B_1 + B_2$.*

Since $\lim_{\delta\to 0}\delta t_\delta = 0$ under Assumption 2.2.3, we have $\lim_{k\to\infty}\alpha_{k-t_k,k-1} = 0$ when $\{\alpha_k\}$ satisfies Condition 2.2.1. Therefore, Equation 2.5 is in the form of the desired one-step contractive inequality (cf. Equation 2.3), which can then be repeatedly used to establish finite-sample guarantees of Algorithm 1.

## 2.5 Finite-Sample Convergence Guarantees

In light of Proposition 2.4.1, to establish finite-sample bounds of Algorithm 1, we repeatedly use Equation 2.5 and evaluate the final expression for using different stepsizes $\{\alpha_k\}$.

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 under Assumption 2.2.3. We now present the finite-sample guarantees of Algorithm 1.

**Theorem 2.5.1.** *Consider $\{x_k\}$ of Algorithm 1. Suppose that Assumptions 2.2.1, 2.2.2, 2.2.3 and 2.2.4 are satisfied, and $\{\alpha_k\}$ satisfies Condition 2.2.1. Then we have the following results.*

*(1) When $k \in [0, K-1]$, we have $\|x_k - x^*\|_c^2 \leq c_1$ almost surely.*

*(2) When $k \geq K$, we have the following finite-sample guarantees.*

*(a) When $\{\alpha_k\}$ satisfies Condition 2.2.1 (1), we have:*

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

*(b) When $\{\alpha_k\}$ satisfies Condition 2.2.1 (2), we have:*

*(i) when $\alpha < 1/\varphi_2$:*

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

*(ii) when $\alpha = 1/\varphi_2$:*

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

*(iii) when $\alpha > 1/\varphi_2$:*

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

*(c) When $\{\alpha_k\}$ satisfies Condition 2.2.1 (3), we have:*

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

*Remark.* Recall that $t_\delta \leq \frac{\log(C/\sigma) + \log(1/\delta)}{\log(1/\sigma)}$ under Assumption 2.2.3. Therefore, 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 2.5.1, 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 $c_1$), and the geometric decay term (for constant stepsize case). The second term is usually understood as the "variance", and hence involves the constant $c_2$, which represents the noise variance at $x^*$. This form of convergence bounds is qualitatively similar to that of SGD type of algorithms presented in [23, 43]. However, compared to [23, 43], Algorithm 1 does not involve the gradient of any function, and has Markovian noise $\{Y_k\}$. Together they impose fundamental challenges in analyzing Algorithm 1.

From Theorem 2.5.1, we see that constant stepsize is very efficient in driving the bias the zero, but cannot eliminate the variance even asymptotically. This suggests using diminishing stepsizes to eliminate the variance. When using linear stepsize $\alpha_k = \frac{\alpha}{k+h}$, the convergence bounds crucially depend on the value of $\alpha$. In order to balance the bias and the variance terms to achieve the optimal convergence rate, we need to choose $\alpha > 1/\varphi_2$, and the resulting optimal convergence rate is roughly $\mathcal{O}(\log(k)/k)$. When using polynomial stepsize, although the convergence rate is the sub-optimal $\mathcal{O}(\log(k)/k^\xi)$, it is more robust in the sense that it does not depend on $\alpha$.

Switching focus, we now revisit the constants $\{\varphi_i\}_{1\leq i\leq 3}$ in Theorem 2.5.1, which as mentioned earlier, depend only on the contraction norm $\|\cdot\|_c$ and the contraction factor $\beta$. In the following lemma, we consider two cases where $\|\cdot\|_c = \|\cdot\|_2$ and $\|\cdot\|_c = \|\cdot\|_\infty$. Both of them will be useful when we study convergence bounds of RL algorithms.

**Corollary 2.5.1.** *The following bounds hold regarding the constants* $\{\varphi_i\}_{1\leq i\leq 3}$.

*(1) When* $\|\cdot\|_c = \|\cdot\|_2$, *we have* $\varphi_1 \leq 1$, $\varphi_2 \geq 1-\beta$, *and* $\varphi_3 \leq 228$.

*(2) When* $\|\cdot\|_c = \|\cdot\|_\infty$, *we have* $\varphi_1 \leq 3$, $\varphi_2 \geq \frac{1-\beta}{2}$, *and* $\varphi_3 \leq \frac{456e\log(d)}{1-\beta}$.

Note that when compared to $\|\cdot\|_2$-contraction, where the constant $\varphi_3$ is bounded by a numerical constant, the upper bound for $\varphi_3$ has an additional $\frac{\log(d)}{1-\beta}$ factor under the $\|\cdot\|_\infty$-contraction. In general, we cannot hope to improve the dimension dependence beyond

$\log(d)$. To see this, consider the trivial case where $F(\cdot,\cdot)$ is identically equal to zero, and $\{w_k\}$ is an i.i.d. sequence of standard normal random vectors. 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 $x = 0$, 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 iterate $x_k$ is simply the running averages of $\{w_k\}$, i.e., $x_k = \frac{1}{k}\sum_{i=0}^{k-1} w_i$ for all $k \geq 1$, which implies $x_k \sim \frac{1}{\sqrt{k}}\mathcal{N}(0, I_d)$. It follows that $\mathbb{E}[\|x_k\|_\infty^2] = O(\frac{\log(d)}{k})$ [53]. Thus in this setting, our resulting finite-sample bounds under $\ell_\infty$-norm contraction are order-wise tight both in terms of the convergence rate and the dimensional dependence.

## 2.6 Proof of All Theoretical Results

In this section, we present the proofs of Proposition 2.3.1, Proposition 2.4.1 and Theorem 2.5.1. The proofs of all technical lemmas used here are provided in Section 2.7.

### 2.6.1 Proof of Proposition 2.3.1

(1) The convexity of $M_f^{\theta,g}(x)$ follows from Theorem 2.19 of [46]. Since $f(\cdot)$ is proper, closed, and convex, and $g(\cdot)$ is $L$ – smooth with respect to $\|\cdot\|_s$, we have by [46, Theorem 5.30 (a)] that $M_f^{\theta,g}(x) = (f\Box\frac{g}{\theta})(x)$ is $\frac{L}{\theta}$ – smooth with respect to $\|\cdot\|_s$.

(2) It is clear from the definition of $M_f^{\theta,g}(x)$ that it is non-negative and is equal to zero if and only if $x = 0$. Now for any $c \in \mathbb{R}$, we have

$$\begin{aligned}
M_f^{\theta,g}(cx) &= \min_u \left\{\frac{1}{2}\|u\|_c^2 + \frac{1}{2\theta}\|cx - u\|_s^2\right\} \\
&= \min_v \left\{\frac{1}{2}\|cv\|_c^2 + \frac{1}{2\theta}\|cx - cv\|_s^2\right\} \qquad \text{(change of variable } u = cv\text{)} \\
&= |c|^2 M_f^{\theta,g}(x).
\end{aligned}$$

Thus, $\sqrt{M_f^{\theta,g}(cx)} = |c|\sqrt{M_f^{\theta,g}(x)}$. It remains to show the triangle inequality.

For any $x_1, x_1 \in \mathbb{R}^d$, let

$$u_1 \in \arg\min_{u\in\mathbb{R}^d} \left\{ \frac{1}{2}\|u\|_c^2 + \frac{1}{2\theta}\|x_1 - u\|_s^2 \right\},$$
$$u_2 \in \arg\min_{u\in\mathbb{R}^d} \left\{ \frac{1}{2}\|u\|_c^2 + \frac{1}{2\theta}\|x_2 - u\|_s^2 \right\}.$$

Then we have

$$\begin{aligned}
& M_f^{\theta,g}(x_1+x_2) \\
=\ & \min_u \left\{ \frac{1}{2}\|u\|_c^2 + \frac{1}{2\theta}\|x_1 + x_2 - u\|_s^2 \right\} \\
\leq\ & \frac{1}{2}\|u_1+u_2\|_c^2 + \frac{1}{2\theta}\|x_1+x_2-u_1-u_2\|_s^2 && \text{(choose } u = u_1+u_2\text{)} \\
\leq\ & \frac{1}{2}(\|u_1\|_c + \|u_2\|_c)^2 + \frac{1}{2\theta}(\|x_1-u_1\|_s + \|x_2-u_2\|_s)^2 \\
=\ & M_f^{\theta,g}(x_1) + M_f^{\theta,g}(x_2) + \|u_1\|_c\|u_2\|_c + \frac{1}{\theta}\|x_1-u_1\|_s\|x_2-u_2\|_s \\
\leq\ & M_f^{\theta,g}(x_1) + M_f^{\theta,g}(x_2) + 2\sqrt{\frac{1}{2}\|u_1\|_c^2 + \frac{1}{2\theta}\|x_1-u_1\|_s^2}\sqrt{\frac{1}{2}\|u_2\|_c^2 + \frac{1}{2\theta}\|x_2-u_2\|_s^2} \\
=\ & M_f^{\theta,g}(x_1) + M_f^{\theta,g}(x_2) + 2\sqrt{M_f^{\theta,g}(x_1)M_f^{\theta,g}(x_2)}.
\end{aligned}$$

It follows that $\sqrt{M_f^{\theta,g}(x_1+x_2)} \leq \sqrt{M_f^{\theta,g}(x_1)} + \sqrt{M_f^{\theta,g}(x_2)}$ for any $x_1, x_2 \in \mathbb{R}^d$. Therefore, $M_f^{\theta,g}(\cdot)$ is a norm-square function and we can write $M_f^{\theta,g}(x) = \frac{1}{2}\|x\|_m^2$ for some norm $\|\cdot\|_m$.

(3) We first derive the upper bound. By definition of $M_f^{\theta,g}(x)$, we have

$$\begin{aligned}
M_f^{\theta,g}(x) &= \min_{u\in\mathbb{R}^d} \left\{ \frac{1}{2}\|u\|_c^2 + \frac{1}{2\theta}\|x-u\|_s^2 \right\} \\
&\geq \min_{u\in\mathbb{R}^d} \left\{ \frac{1}{2}\|u\|_c^2 + \frac{1}{2\theta u_{cs}^2}\|x-u\|_c^2 \right\} && (\|\cdot\|_c \leq u_{cs}\|\cdot\|_s) \\
&\geq \min_{u\in\mathbb{R}^d} \left\{ \frac{1}{2}\|u\|_c^2 + \frac{1}{2\theta u_{cs}^2}(\|x\|_c - \|u\|_c)^2 \right\} && \text{(triangle inequality)} \\
&= \min_{y\in\mathbb{R}} \left\{ \frac{1}{2}y^2 + \frac{1}{2\theta u_{cs}^2}(\|x\|_c - y)^2 \right\} && \text{(change of variable: } y = \|u\|_c^2\text{)}
\end{aligned}$$

$$
\begin{aligned}
&= \min_{y\in\mathbb{R}} \left\{ \left( \frac{1}{2} + \frac{1}{2\theta u_{cs}^2} \right) y^2 - \frac{1}{\theta u_{cs}^2} \|x\|_c y + \frac{1}{2\theta u_{cs}^2} \|x\|_c^2 \right\} \\
&= \frac{1}{2}\|x\|_c^2 \frac{1}{\theta u_{cs}^2 + 1} && \text{(minimum of a quadratic function)} \\
&= \frac{1}{\theta u_{cs}^2 + 1} f(x).
\end{aligned}
$$

It follows that $f(x) \leq (1 + \theta u_{cs}^2)\, M_f^{\theta,g}(x)$ for all $x$, which implies $\|\cdot\|_c \leq (1+\theta u_{cs}^2)^{1/2}\|\cdot\|_m$.

Next we show the lower bound. Similarly, by definition we have for any $x \in \mathbb{R}^d$ that

$$
\begin{aligned}
M_f^{\theta,g}(x) &= \min_{u\in\mathbb{R}^d} \left\{ \frac{1}{2}\|u\|_c^2 + \frac{1}{2\theta}\|x-u\|_s^2 \right\} \\
&\leq \min_{\alpha\in(0,1)} \left\{ \frac{1}{2}\|\alpha x\|_c^2 + \frac{1}{2\theta}\|x-\alpha x\|_s^2 \right\} && \text{(restrict } u = \alpha x \text{ for } \alpha \in (0,1)) \\
&\leq \frac{1}{2}\|x\|_c^2 \min_{\alpha\in(0,1)} \left\{ \alpha^2 + \frac{(1-\alpha)^2}{\theta \ell_{cs}^2} \right\} && (\ell_{cs}\|\cdot\|_s \leq \|\cdot\|_c) \\
&= \frac{1}{1+\theta\ell_{cs}^2} \frac{1}{2}\|x\|_c^2 && \text{(minimum of the quadratic function)} \\
&= \frac{1}{1+\theta\ell_{cs}^2} f(x).
\end{aligned}
$$

It follows that $f(x) \geq (1 + \theta \ell_{cs}^2)\, M_f^{\theta,g}(x)$ for all $x$, which implies $\|\cdot\|_c \geq (1+\theta\ell_{cs}^2)^{1/2}\|\cdot\|_m$.

### 2.6.2 Proof of Proposition 2.4.1

Before proving Proposition 2.4.1, we first explicitly state the requirement for choosing the stepsize sequence $\{\alpha_k\}$.

**Condition 2.6.1.** The sequence $\{\alpha_k\}$ is non-increasing 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 2.6.1 boils down to Condition 2.2.1 when the expression of the stepsize is explicitly specified.

Using Proposition 2.3.1 (1) and the update equation of Algorithm 1, we have for any $k \geq 0$ that

$$\begin{aligned}
M_f^{\theta,g}(x_{k+1}-x^*) \leq\;& M_f^{\theta,g}(x_k-x^*) + \langle \nabla M_f^{\theta,g}(x_k-x^*), x_{k+1}-x_k\rangle + \frac{L}{2\theta}\|x_{k+1}-x_k\|_s^2 \\
=\;& M_f^{\theta,g}(x_k-x^*) + \alpha_k\langle \nabla M_f^{\theta,g}(x_k-x^*), F(x_k,Y_k)-x_k+w_k\rangle \\
&+ \frac{L\alpha_k^2}{2\theta}\|F(x_k,Y_k)-x_k+w_k\|_s^2 \\
=\;& M_f^{\theta,g}(x_k-x^*) + \underbrace{\alpha_k\langle \nabla M_f^{\theta,g}(x_k-x^*), \bar{F}(x_k)-x_k\rangle}_{T_1\text{: Expected update}} \\
&+ \underbrace{\alpha_k\langle \nabla M_f^{\theta,g}(x_k-x^*), w_k\rangle}_{T_2\text{: Error due to Martingale difference noise } w_k} \\
&+ \underbrace{\alpha_k\langle \nabla M_f^{\theta,g}(x_k-x^*), F(x_k,Y_k)-\bar{F}(x_k)\rangle}_{T_3\text{: Error due to Markovian noise } Y_k} \\
&+ \underbrace{\frac{L\alpha_k^2}{2\theta}\|F(x_k,Y_k)-x_k+w_k\|_s^2}_{T_4\text{: Error due to discretization and noises}}. \qquad (2.6)
\end{aligned}$$

The term $T_1$ represents the expected update of Algorithm 1, and is bounded in the following lemma.

**Lemma 2.6.1.** *The following inequality holds for all $k \geq 0$:*

$$T_1 \leq -2\left(1-\beta\frac{u_{cm}}{\ell_{cm}}\right)\alpha_k M_f^{\theta,g}(x_k-x^*).$$

As we have seen in Lemma 2.6.1, the term $T_1$ provides us the desired negative drift, i.e., the $-\mathcal{O}(\alpha_k)$ term in the target one-step contractive inequality (cf. Equation 2.3). What remains to do is to control all the error terms $T_2$ to $T_4$ in Equation 2.6.

We begin with the term $T_2$. Since $\{w_k\}$ is a martingale difference sequence with respect to the filtration $\mathcal{F}_k$ (cf. Assumption 2.2.4), while $x_k$ is measurable with respect to $\mathcal{F}_k$, we have by the tower property of conditional expectation that

$$\mathbb{E}[T_2] = \mathbb{E}[\mathbb{E}[T_2 \mid \mathcal{F}_k]] = \alpha_k \mathbb{E}[\langle \nabla M_f^{\theta,g}(x_k - x^*), \mathbb{E}[w_k \mid \mathcal{F}_k]]\rangle = 0.$$

Next we analyze the error term $T_3$, which is due to the Markovian noise $\{Y_k\}$. We first decompose $T_3$ in the following way:

$$\begin{aligned} T_3 &= \alpha_k \langle \nabla M_f^{\theta,g}(x_k - x^*), F(x_k, Y_k) - \bar{F}(x_k)\rangle \\ &= \alpha_k \underbrace{\langle \nabla M_f^{\theta,g}(x_k - x^*) - \nabla M_f^{\theta,g}(x_{k-t_k} - x^*), F(x_k, Y_k) - \bar{F}(x_k)\rangle}_{T_{31}} \\ &\quad + \alpha_k \underbrace{\langle \nabla M_f^{\theta,g}(x_{k-t_k} - x^*), F(x_k, Y_k) - F(x_{k-t_k}, Y_k) + \bar{F}(x_{k-t_k}) - \bar{F}(x_k)\rangle}_{T_{32}} \\ &\quad + \alpha_k \underbrace{\langle \nabla M_f^{\theta,g}(x_{k-t_k} - x^*), F(x_{k-t_k}, Y_k) - \bar{F}(x_{k-t_k})\rangle}_{T_{33}}. \end{aligned} \tag{2.7}$$

To proceed, we need the following lemma, which allows us to control the difference between $x_{k_1}$ and $x_{k_2}$ when $|k_1 - k_2|$ is relatively small.

**Lemma 2.6.2.** *Given non-negative integers $k_1 \leq k_2$ satisfying $\alpha_{k_1,k_2-1} \leq \frac{1}{4A}$, we have for all $k \in [k_1, k_2]$:*

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

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

Using the assumption that $\alpha_{k_1,k_2-1} \leq \frac{1}{4A}$ in the resulting inequality of Lemma 2.6.2, we have the following corollary, which will also be frequently used in our analysis.

**Corollary 2.6.1.** *Under same conditions given in Lemma 2.6.2, we have for all $k \in [k_1, k_2]$ that $\|x_k - x_{k_1}\|_c \leq \max(\|x_{k_1}\|_c, \|x_{k_2}\|_c) + \frac{B}{A}$.*

Recall that we require $\alpha_{k-t_k,k-1} \leq \frac{1}{4A}$ for all $k \geq t_k$ in Condition 2.6.1. Therefore, Lemma 2.6.2 and Corollary 2.6.1 are applicable when $k_1 = k - t_k$ and $k_2 = k - 1$ for any $k \geq t_k$.

Now we are ready to control the terms $T_{31}$, $T_{32}$, and $T_{33}$ in the following lemma. The terms $T_{31}$ and $T_{32}$ are controlled mainly by constantly applying Lemma 2.6.2 and the Lipschitz property of the operator $F(\cdot)$ (cf. Assumptions 2.2.2). Bounding the term $T_{33}$ requires using the geometric mixing of the Markov chain $\{Y_k\}$ (cf. Assumption 2.2.3).

**Lemma 2.6.3.** *The following inequalities hold for all $k \geq t_k$:*

*(1)* $T_{31} \leq \frac{16LA^2u_{cm}^2\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}M_f^{\theta,g}(x_k - x^*) + \frac{8L\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}(A\|x^*\|_c + B)^2,$

*(2)* $T_{32} \leq \frac{64LA^2u_{cm}^2\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}M_f^{\theta,g}(x_k - x^*) + \frac{32L\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}(A\|x^*\|_c + B)^2,$

*(3)* $\mathbb{E}[T_{33}] \leq \frac{32LA^2u_{cm}^2\alpha_k}{\theta\ell_{cs}^2}\mathbb{E}[M_f^{\theta,g}(x_k - x^*)] + \frac{16L\alpha_k}{\theta\ell_{cs}^2}(A\|x^*\|_c + B)^2.$

Now that Lemma 2.6.3 provides upper bounds on the terms $T_{31}$, $T_{32}$, and $T_{33}$, using them in Equation 2.7 and we have the following result.

**Lemma 2.6.4.** *The following inequality holds for all $k \geq t_k$:*

$$\mathbb{E}[T_3] \leq \frac{112LA^2u_{cm}^2\alpha_k\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}\mathbb{E}[M_f^{\theta,g}(x_k - x^*)] + \frac{56L\alpha_k\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}(A\|x^*\|_c + B)^2.$$

Lastly, we bound the error term $T_4$ in the following lemma.

**Lemma 2.6.5.** *It holds for any $k \geq 0$ that*

$$T_4 \leq \frac{2LA^2u_{cm}^2\alpha_k^2}{\theta\ell_{cs}^2}M_f^{\theta,g}(x_k - x^*) + \frac{L\alpha_k^2}{\theta\ell_{cs}^2}(A\|x^*\|_c + B)^2.$$

Now we have control on all the error terms $T_1$ to $T_4$. Using them in Equation 2.6, and we have for all $k \geq t_k$:

$$\begin{aligned}
&\mathbb{E}[M_f^{\theta,g}(x_{k+1} - x^*)] \\
\leq\ &\left(1 - 2\varphi_2\alpha_k + \frac{114LA^2u_{cm}^2\alpha_k\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}\right)\mathbb{E}[M_f^{\theta,g}(x_k - x^*)] \\
&+ \frac{57L\alpha_k\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}(A\|x^*\|_c + B)^2
\end{aligned}$$

$$= \left(1 - 2\varphi_2\alpha_k + \varphi_3 A^2 \alpha_k \alpha_{k-t_k,k-1}\right) \mathbb{E}[M_f^{\theta,g}(x_k - x^*)] + \frac{\varphi_3 c_2 \alpha_k \alpha_{k-t_k,k-1}}{2u_{cm}^2}.$$

This proves Proposition 2.4.1.

### 2.6.3 Proof of Theorem 2.5.1

Note that Proposition 2.4.1 provides the one-step contractive inequality. We next repeatedly use Proposition 2.4.1 to derive finite-sample convergence bounds of Algorithm 1. Since $\alpha_{k-t_k,k-1} \leq \varphi_2/(\varphi_3 A^2)$ for all $k \geq K$ (cf. Condition 2.6.1), we have by Proposition 2.4.1 that

$$\mathbb{E}[M_f^{\theta,g}(x_{k+1} - x^*)] \leq (1 - \varphi_2\alpha_k)\, \mathbb{E}[M_f^{\theta,g}(x_k - x^*)] + \frac{c_2\varphi_3\alpha_k\alpha_{k-t_k,k-1}}{2u_{cm}^2}$$

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

$$\begin{aligned}
&\mathbb{E}[\|x_k - x^*\|_c^2] \\
&\leq 2u_{cm}^2 \mathbb{E}[M_f^{\theta,g}(x_k - x^*)] && \text{(Proposition 2.3.1 (3))} \\
&\leq 2u_{cm}^2 \mathbb{E}[M_f^{\theta,g}(x_K - x^*)] \prod_{j=K}^{k-1}(1 - \varphi_2\alpha_j) + c_2\varphi_3 \sum_{i=K}^{k-1} \alpha_i \alpha_{i-t_i,i-1} \prod_{j=i+1}^{k-1} (1 - \varphi_2\alpha_j) \\
&\leq \frac{u_{cm}^2}{\ell_{cm}^2} \mathbb{E}[\|x_K - x^*\|_c^2] \prod_{j=K}^{k-1}(1 - \varphi_2\alpha_j) + c_2\varphi_3 \sum_{i=K}^{k-1} \alpha_i \alpha_{i-t_i,i-1} \prod_{j=i+1}^{k-1} (1 - \varphi_2\alpha_j) \\
& && \text{(Proposition 2.3.1 (3))} \\
&= \varphi_1 \mathbb{E}[\|x_K - x^*\|_c^2] \prod_{j=K}^{k-1}(1 - \varphi_2\alpha_j) + \varphi_3 c_2 \sum_{i=K}^{k-1} \alpha_i \alpha_{i-t_i,i-1} \prod_{j=i+1}^{k-1} (1 - \varphi_2\alpha_j).
\end{aligned}$$

According to Condition 2.6.1, we also have $\alpha_{0,k-1} \leq 1/(4A)$ for any $k \in [0, K]$. Using Corollary 2.6.1 one more time and we have for any $k \in [0, K]$ that

$$\mathbb{E}[\|x_k - x^*\|_c^2] \leq \mathbb{E}[(\|x_k - x_0\|_c + \|x_0 - x^*\|_c)^2]$$

$$
\begin{aligned}
&\leq \left(\|x_0 - x^*\|_c + \|x_0\|_c + \frac{B}{A}\right)^2 \\
&= c_1.
\end{aligned}
$$

This proves Theorem 2.5.1 (1). Since the previous inequality implies $\mathbb{E}[\|x_K - x^*\|_c^2] \leq c_1$, we obtain for all $k \geq K$:

$$
\mathbb{E}[\|x_k - x^*\|_c^2] \leq \varphi_1 c_1 \prod_{j=K}^{k-1}(1 - \varphi_2\alpha_j) + \varphi_3 c_2 \sum_{i=K}^{k-1} \alpha_i \alpha_{i-t_i,i-1} \prod_{j=i+1}^{k-1}(1 - \varphi_2\alpha_j). \tag{2.8}
$$

To proceed and prove Theorem 2.5.1 (2), we next evaluate the RHS of the previous inequality when the stepsize sequence $\{\alpha_k\}$ is explicitly chosen.

**Constant Stepsize.** Consider using constant stepsize $\alpha_k \equiv \alpha$. It is clear that Condition 2.6.1 is satisfied when $\alpha t_\alpha \leq \min(\frac{\varphi_2}{\varphi_3 A^2}, \frac{1}{4A})$. We first verify that there exists a threshold $\bar{\alpha}$ such that $\alpha t_\alpha \leq \min(\frac{\varphi_2}{\varphi_3 A^2}, \frac{1}{4A})$ for all $\alpha \in (0, \bar{\alpha})$.

Note that we have by definition of $t_\alpha$ and Assumption 2.2.3 that

$$
\begin{aligned}
t_\alpha &\leq \min\left\{k \geq 0 \ : \ C\sigma^k \leq \alpha\right\} \\
&= \min\left\{k \geq 0 \ : \ k \geq \frac{\log(1/\alpha) + \log(C)}{\log(1/\sigma)}\right\} \\
&\leq \frac{\log(1/\alpha) + \log(C/\sigma)}{\log(1/\sigma)}.
\end{aligned}
$$

It follows that $\lim_{\alpha \to 0} \alpha t_\alpha = 0$. Hence there exists $\bar{\alpha} \in (0, 1)$ such that Condition 2.6.1 is satisfied for all $\alpha \in (0, \bar{\alpha})$, which is stated in Condition 2.2.1 (1). We next evaluate Equation 2.8. When $\alpha_k \equiv \alpha$, we have for all $k \geq t_\alpha$:

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

$$\leq \varphi_1 c_1 (1-\varphi_2\alpha)^{k-t_\alpha} + \frac{\varphi_3 c_2}{\varphi_2}\alpha t_\alpha.$$

This proves Theorem 2.5.1 (2) (a).

**Linearly Diminishing stepsize.** Consider using linearly diminishing stepsizes of the form $\alpha_k = \frac{\alpha}{k+h}$. We first verify that for any $\alpha > 0$, there exists a threshold $\bar{h}$ such that Condition 2.6.1 is satisfied for all $h \geq \bar{h}$. We begin by comparing $\alpha_{k-t_k}$ with $\alpha_k$. Using Assumption 2.2.3 and we have

$$t_k \leq \frac{\log(k+h) + \log(C/(\sigma\alpha))}{\log(1/\sigma)}.$$

It follows that

$$\frac{\alpha_k}{\alpha_{k-t_k}} = 1 - \frac{t_k}{k+h} \to 1 \text{ as } (k+h) \to \infty.$$

Therefore, there exists $\bar{h}_1 > 0$ such that $\alpha_{k-t_k} \leq 2\alpha_k$ holds for any $k \geq t_k$ when $h \geq \bar{h}_1$. Now consider the requirement stated in Condition 2.6.1. Using the fact that $\{\alpha_k\}$ is non-increasing, we have

$$\alpha_{k-t_k,k-1} \leq t_k \alpha_{k-t_k} \leq 2\alpha_k t_k \to 0 \text{ as } (k+h) \to \infty.$$

Hence there exists $\bar{h}_2 > 0$ such that $\alpha_{k-t_k,k-1} \leq \min(\frac{\varphi_2}{\varphi_3 A^2}, \frac{1}{4A})$ holds for any $k \geq t_k$ when $h \geq \bar{h}_2$. Now choosing $\bar{h} = \max(\bar{h}_1, \bar{h}_2)$, Condition 2.6.1 is satisfied. This is stated in Condition 2.2.1 (2). Furthermore, by construction we have $\alpha_{k-t_k} \leq 2\alpha_k$ for any $k \geq t_k$. We next evaluate the RHS of Equation 2.8 in the following lemma.

**Lemma 2.6.6.** *The following inequality holds for all $k \geq K$:*

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

This proves Theorem 2.5.1 (2) (b).

**Polynomially Diminishing stepsize.** Finally we consider using polynomially diminishing stepsize of the form $\alpha_k = \frac{\alpha}{(k+h)^\xi}$, where $\xi \in (0,1)$ and $\alpha, h > 0$. Using the same line of proof, one can show that for any $\xi \in (0,1)$ and $\alpha > 0$, there exists $\bar{h} > 0$ such that Condition 2.6.1 is satisfied for all $h \geq \bar{h}$. Furthermore, we assume without loss of generality that $\alpha_{k-t_k} \leq 2\alpha_k$ for all $k \geq t_k$ and $\bar{h} \geq [2\xi/(\varphi_2\alpha)]^{1/(1-\xi)}$. We next evaluate the RHS of Equation 2.8 in the following lemma.

**Lemma 2.6.7.** *The following inequality hold for all $k \geq K$:*

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

This proves Theorem 2.5.1 (2) (c).

## 2.7 Proof of All Technical Lemmas

In this section, we present the proofs of all technical lemmas used to establish our main theoretical results in Section 2.6.

### 2.7.1 Proof of Lemma 2.6.1

Using the fact that $\bar{F}(x^*) = x^*$, we have

$$\begin{aligned} &\langle \nabla M_f^{\theta,g}(x_k - x^*), \bar{F}(x_k) - x_k \rangle \\ =\ &\underbrace{\langle \nabla M_f^{\theta,g}(x_k - x^*), \bar{F}(x_k) - \bar{F}(x^*) \rangle}_{T_{1,1}} - \underbrace{\langle \nabla M_f^{\theta,g}(x_k - x^*), x_k - x^* \rangle}_{T_{1,2}}. \end{aligned} \tag{2.9}$$

For the gradient of $M_f^{\theta,g}(x)$, since $M_f^{\theta,g}(x) = \frac{1}{2}\|x\|_m^2$, we have by the chain rule of subdifferential calculus [46, Theorem 3.47] that $\nabla M_f^{\theta,g}(x) = \|x\|_m v_x$, where $v_x \in \partial \|x\|_m$ is a subgradient of the function $\|x\|_m$ at $x$. In fact, from the equation $\nabla M_f^{\theta,g}(x) = \|x\|_m v_x$, we see that $v_x$ is unique (i.e., $v_x = \nabla \|x\|_m$) for all $x \neq 0$.

Now consider the term $T_{1,1}$. Using Hölder's inequality, we have

$$\begin{aligned} T_{1,1} &= \|x_k - x^*\|_m \langle v_{x_k - x^*}, \bar{F}(x_k) - \bar{F}(x^*) \rangle \\ &\leq \|x_k - x^*\|_m \|v_{x_k - x^*}\|_m^* \|\bar{F}(x_k) - \bar{F}(x^*)\|_m, \end{aligned} \tag{2.10}$$

where $\|\cdot\|_m^*$ is the dual norm of $\|\cdot\|_m$. To further control $T_{1,1}$, the following result is needed.

**Lemma 2.7.1** ([54]). *Let $h : \mathcal{D} \to \mathbb{R}$ be a convex function. Then $h$ is $L$ – Lipschitz over $\mathcal{D}$ with respect to some norm $\|\cdot\|$ if and only if for all $w \in \mathcal{D}$ and $z \in \partial h(w)$ we have that $\|z\|_* \leq L$, where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$.*

Since $\|x\|_m$ as a function of $x$ is $1$ – Lipschitz with respect to $\|\cdot\|_m$, we have by Lemma 2.7.1 that $\|v_{x_k - x^*}\|_m^* \leq 1$. For the term $\|\bar{F}(x_k) - \bar{F}(x^*)\|_m$ in Equation 2.10, using Proposition 2.3.1 (3) and the contraction of $\bar{F}(\cdot)$ with respect to $\|\cdot\|_c$, we have

$$\begin{aligned} \frac{1}{2}\|\bar{F}(x_k) - \bar{F}(x^*)\|_m^2 &= M_f^{\theta,g}(\bar{F}(x_k) - \bar{F}(x^*)) \\ &\leq \frac{1}{1 + \theta \ell_{cs}^2} f(\bar{F}(x_k) - \bar{F}(x^*)) \qquad \text{(Proposition 2.3.1 (3))} \end{aligned}$$

$$
\begin{aligned}
&\leq \frac{\beta^2}{1+\theta\ell_{cs}^2} f(x_k - x^*) && \text{(Assumption 2.2.1)}\\
&\leq \beta^2 \frac{1+\theta u_{cs}^2}{1+\theta\ell_{cs}^2} M_f^{\theta,g}(x_k - x^*) && \text{(Proposition 2.3.1 (3))}\\
&= \frac{\beta^2}{2}\frac{1+\theta u_{cs}^2}{1+\theta\ell_{cs}^2}\|x_k - x^*\|_m^2\\
&= \frac{\beta^2}{2}\frac{u_{cm}^2}{\ell_{cm}^2}\|x_k - x^*\|_m^2,
\end{aligned}
$$

which implies

$$\|\bar{F}(x_k) - \bar{F}(x^*)\|_m \leq \beta \frac{u_{cm}}{\ell_{cm}} \|x_k - x^*\|_m.$$

Substituting the upper bounds we obtained for $\|v_{x_k-x^*}\|_m^*$ and $\|\bar{F}(x_k) - \bar{F}(x^*)\|_m$ into Equation 2.10, we have

$$
\begin{aligned}
T_{1,1} &\leq \|x_k - x^*\|_m \|v_{x_k-x^*}\|_m^* \|\bar{F}(x_k) - \bar{F}(x^*)\|_m\\
&\leq \beta\frac{u_{cm}}{\ell_{cm}}\|x_k - x^*\|_m^2\\
&= 2\beta\frac{u_{cm}}{\ell_{cm}} M_f^{\theta,g}(x_k - x^*).
\end{aligned}
$$

Now consider the term $T_{1,2}$ in Equation 2.9. Since the norm $\|\cdot\|_m$ is a convex function of $x$, we have by definition of convexity that $\|0\|_m - \|x_k - x^*\|_m \geq \langle v_{x_k-x^*}, -(x_k - x^*)\rangle$. Therefore, we have

$$T_{1,2} = \|x_k - x^*\|_m \langle v_{x_k-x^*}, x_k - x^*\rangle \geq \|x_k - x^*\|_m^2 = 2M_f^{\theta,g}(x_k - x^*).$$

Combining the bounds on $T_{1,1}$ and $T_{1,2}$ and we obtain

$$T_1 = \alpha_k(T_{1,1} - T_{1,2}) \leq -2\left(1 - \beta\frac{u_{cm}}{\ell_{cm}}\right)\alpha_k M_f^{\theta,g}(x_k - x^*).$$

### 2.7.2 Proof of Lemma 2.6.2

We first show that under Assumption 2.2.2, the size of $\|F(x,y)\|_c$ and $\|\bar{F}(x)\|_c$ can grow at most affinely in terms of $\|x\|_c$. Using triangle inequality, we have

$$\|F(x,y)\|_c - \|F(\mathbf{0},y)\|_c \leq \|F(x,y) - F(\mathbf{0},y)\|_c \leq A_1\|x\|_c, \quad \forall\, x \in \mathbb{R}^d, y \in \mathcal{Y},$$

where the last inequality follows from Assumption 2.2.2. It follows that

$$\|F(x,y)\|_c \leq A_1\|x\|_c + \|F(\mathbf{0},y)\|_c \leq A_1\|x\|_c + B_1.$$

Furthermore, we have by Jensen's inequality and the convexity of norms that

$$\|\bar{F}(x)\|_c = \|\mathbb{E}_{Y\sim\mu_Y}[F(x,Y)]\|_c \leq \mathbb{E}_{Y\sim\mu_Y}[\|F(x,Y)\|_c] \leq A_1\|x\|_c + B_1.$$

The previous two inequalities will be frequently used in the derivation here after. Now we proceed to prove Lemma 2.6.2.

For any $k \in [k_1, k_2 - 1]$, using triangle inequality, we have

$$\begin{aligned}
\|x_{k+1}\|_c - \|x_k\|_c &\leq \|x_{k+1} - x_k\|_c \\
&= \alpha_k\|F(x_k, Y_k) - x_k + w_k\|_c \\
&\leq \alpha_k(\|F(x_k, Y_k)\|_c + \|x_k\|_c + \|w_k\|_c) \\
&\leq \alpha_k(A_1\|x_k\|_c + B_1 + \|x_k\|_c + A_2\|x_k\|_c + B_2). \\
&\qquad\qquad\qquad\qquad\qquad \text{(Assumptions 2.2.2 and 2.2.4)} \\
&\leq \alpha_k((A_1 + A_2 + 1)\|x_k\|_c + B_1 + B_2) \\
&= \alpha_k(A\|x_k\|_c + B). \qquad (2.11)
\end{aligned}$$

Note that the previous inequality is equivalent to

$$\|x_{k+1}\|_c + \frac{B}{A} \leq (1 + A\alpha_k) \left( \|x_k\|_c + \frac{B}{A} \right),$$

which implies for all $k \in [k_1, k_2]$:

$$\|x_k\|_c \leq \prod_{j=k_1}^{k-1} (1 + A\alpha_j) \left( \|x_{k_1}\|_c + \frac{B}{A} \right) - \frac{B}{A}.$$

Using the fact that $1 + x \leq e^x \leq 1 + 2x$ for all $x \in [0, 1/2]$, we have when $\alpha_{k_1,k_2-1} \leq \frac{1}{4A}$:

$$\prod_{j=k_1}^{k-1} (1 + A\alpha_j) \leq \exp\left(A\alpha_{k_1,k-1}\right) \leq 1 + 2A\alpha_{k_1,k-1}.$$

It follows that for all $k \in [k_1, k_2]$:

$$\|x_k\|_c \leq (1 + 2A\alpha_{k_1,k-1})\|x_{k_1}\|_c + 2B\alpha_{k_1,k-1}.$$

Using the previous inequality in Equation 2.11 and we have for any $k \in [k_1, k_2 - 1]$:

$$\begin{aligned} \|x_{k+1} - x_k\|_c &\leq \alpha_k(A\|x_k\|_c + B) \\ &\leq \alpha_k A(1 + 2A\alpha_{k_1,k-1})\|x_{k_1}\|_c + 2\alpha_k AB\alpha_{k_1,k-1} \\ &\leq 2\alpha_k(A\|x_{k_1}\|_c + B). && (\alpha_{k_1,k-1} \leq \tfrac{1}{4A}) \end{aligned}$$

Hence, we have for any $k \in [k_1, k_2]$:

$$\|x_k - x_{k_1}\|_c \leq \sum_{j=k_1}^{k-1} \|x_{j+1} - x_j\|_c \leq 2 \sum_{j=k_1}^{k-1} \alpha_j (A\|x_{k_1}\|_c + B) = 2\alpha_{k_1,k-1}(A\|x_{k_1}\|_c + B).$$

Since $\alpha_{k_1,k-1} \leq \alpha_{k_1,k_2-1}$ when $k \in [k_1, k_2]$, we obtain the first claimed inequality:

$$\|x_k - x_{k_1}\|_c \leq 2\alpha_{k_1,k_2-1}(A\|x_{k_1}\|_c + B), \quad \forall\, k \in [k_1, k_2].$$

Now for the second claimed inequality, since

$$\begin{aligned}
\|x_{k_2} - x_{k_1}\|_c &\leq 2\alpha_{k_1,k_2-1}(A\|x_{k_1}\|_c + B)\\
&\leq 2\alpha_{k_1,k_2-1}(A\|x_{k_1} - x_{k_2}\|_c + A\|x_{k_2}\|_c + B)\\
&\leq \frac{1}{2}\|x_{k_2} - x_{k_1}\|_c + 2\alpha_{k_1,k_2-1}(A\|x_{k_2}\|_c + B),
\end{aligned}$$

we have $\|x_{k_2} - x_{k_1}\|_c \leq 4\alpha_{k_1,k_2-1}(A\|x_{k_2}\|_c + B)$. Therefore, we have for any $k \in [k_1, k_2]$:

$$\begin{aligned}
\|x_k - x_{k_1}\|_c &\leq 2\alpha_{k_1,k_2-1}(A\|x_{k_1}\|_c + B)\\
&\leq 2\alpha_{k_1,k_2-1}(A\|x_{k_1} - x_{k_2}\|_c + A\|x_{k_2}\|_c + B)\\
&\leq 2\alpha_{k_1,k_2-1}(4A\alpha_{k_1,k_2-1}(A\|x_{k_2}\|_c + B) + A\|x_{k_2}\|_c + B)\\
&\leq 4\alpha_{k_1,k_2-1}(A\|x_{k_2}\|_c + B), \qquad\qquad (\alpha_{k_1,k_2-1} \leq \tfrac{1}{4A})
\end{aligned}$$

which is the second claimed inequality.

### 2.7.3 Proof of Lemma 2.6.3

(1) For the term $T_{31}$, using Hölder's inequality and we have

$$\begin{aligned}
T_{31} &= \langle \nabla M_f^{\theta,g}(x_k - x^*) - \nabla M_f^{\theta,g}(x_{k-t_k} - x^*), F(x_k, Y_k) - \bar{F}(x_k)\rangle\\
&\leq \|\nabla M_f^{\theta,g}(x_k - x^*) - \nabla M_f^{\theta,g}(x_{k-t_k} - x^*)\|_s^*\|F(x_k, Y_k) - \bar{F}(x_k)\|_s\\
&\leq \frac{1}{\ell_{cs}}\|\nabla M_f^{\theta,g}(x_k - x^*) - \nabla M_f^{\theta,g}(x_{k-t_k} - x^*)\|_s^*\|F(x_k, Y_k) - \bar{F}(x_k)\|_c,\\
&\qquad\qquad (\ell_{cs}\|\cdot\|_s \leq \|\cdot\|_c)
\end{aligned}$$

where $\|\cdot\|_s^*$ denotes the dual norm of $\|\cdot\|_s$. We first control the term $\|\nabla M_f^{\theta,g}(x_k - x^*) - \nabla M_f^{\theta,g}(x_{k-t_k} - x^*)\|_s^*$. Recall that an equivalent definition of a convex function $h(x)$ been $L$ – smooth with respect to some norm $\|\cdot\|$ is that

$$\|\nabla h(x_1) - \nabla h(x_2)\|_* \leq L\|x_1 - x_2\|, \quad \forall\, x_1, x_2,$$

where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$ [46]. Therefore, since $M_f^{\theta,g}(x)$ is $\frac{L}{\theta}$-smooth with respect to $\|\cdot\|_s$, we have

$$\begin{aligned}
&\|\nabla M_f^{\theta,g}(x_k - x^*) - \nabla M_f^{\theta,g}(x_{k-t_k} - x^*)\|_s^* \\
\leq\;& \frac{L}{\theta}\|x_k - x_{k-t_k}\|_s \\
\leq\;& \frac{L}{\theta \ell_{cs}}\|x_k - x_{k-t_k}\|_c \\
\leq\;& \frac{4L\alpha_{k-t_k,k-1}}{\theta \ell_{cs}}(A\|x_k - x^*\|_c + A\|x^*\|_c + B),
\end{aligned} \tag{2.12}$$

where the last line follows from Lemma 2.6.2 and triangle inequality.

We next control the term $\|F(x_k, Y_k) - \bar{F}(x_k)\|_c$. Using Assumptions 2.2.1, 2.2.2, and the fact that $\bar{F}(x^*) = x^*$, we have

$$\begin{aligned}
\|F(x_k, Y_k) - \bar{F}(x_k)\|_c &= \|F(x_k, Y_k) - \bar{F}(x_k) + \bar{F}(x^*) - x^*\|_c \\
&\leq \|F(x_k, Y_k)\|_c + \|\bar{F}(x_k) - \bar{F}(x^*)\|_c + \|x^*\|_c \\
&\leq A_1\|x_k\|_c + B_1 + \|x_k - x^*\|_c + \|x^*\|_c \\
&\leq (A_1 + 1)\|x_k - x^*\|_c + (A_1 + 1)\|x^*\|_c + B_1 \\
&\leq A\|x_k - x^*\|_c + A\|x^*\|_c + B.
\end{aligned}$$

It follows that

$$T_{31} \leq \frac{1}{\ell_{cs}}\|\nabla M_f^{\theta,g}(x_k - x^*) - \nabla M_f^{\theta,g}(x_{k-t_k} - x^*)\|_s^*\|F(x_k, Y_k) - \bar{F}(x_k)\|_c$$

$$\begin{aligned}
&\leq \frac{4L\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}(A\|x_k - x^*\|_c + A\|x^*\|_c + B)^2 \\
&\leq \frac{8L\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}A^2\|x_k - x^*\|_c^2 + \frac{8L\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}(A\|x^*\|_c + B)^2 \quad (a+b)^2 \leq 2(a^2+b^2) \\
&\leq \frac{16LA^2u_{cm}^2\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}M_f^{\theta,g}(x_k - x^*) + \frac{8L\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}(A\|x^*\|_c + B)^2.
\end{aligned}$$

(2) Consider the term $T_{32}$. Using Hölder's inequality and we have

$$\begin{aligned}
T_{32} &= \langle \nabla M_f^{\theta,g}(x_{k-t_k} - x^*), F(x_k, Y_k) - F(x_{k-t_k}, Y_k) + \bar{F}(x_{k-t_k}) - \bar{F}(x_k)\rangle \\
&\leq \|\nabla M_f^{\theta,g}(x_{k-t_k} - x^*)\|_s^* \|F(x_k, Y_k) - F(x_{k-t_k}, Y_k) + \bar{F}(x_{k-t_k}) - \bar{F}(x_k)\|_s \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(Hölder's inequality)} \\
&\leq \frac{1}{\ell_{cs}}\|\nabla M_f^{\theta,g}(x_{k-t_k} - x^*)\|_s^* \|F(x_k, Y_k) - F(x_{k-t_k}, Y_k) + \bar{F}(x_{k-t_k}) - \bar{F}(x_k)\|_c.
\end{aligned}$$

For the term $\|\nabla M_f^{\theta,g}(x_{k-t_k} - x^*)\|_s^*$, we have

$$\begin{aligned}
\|\nabla M_f^{\theta,g}(x_{k-t_k} - x^*)\|_s^* &= \|\nabla M_f^{\theta,g}(x_{k-t_k} - x^*) - \nabla M_f^{\theta,g}(x^* - x^*)\|_s^* \\
&\leq \frac{L}{\theta}\|x_{k-t_k} - x^*\|_s \qquad \text{(equivalent definition of smoothness)} \\
&\leq \frac{L}{\theta\ell_{cs}}\|x_{k-t_k} - x^*\|_c \\
&\leq \frac{L}{\theta\ell_{cs}}(\|x_{k-t_k} - x_k\|_c + \|x_k - x^*\|_c) \\
&\leq \frac{2L}{\theta\ell_{cs}}\left(\|x_k - x^*\|_c + \|x^*\|_c + \frac{B}{A}\right), \qquad (2.13)
\end{aligned}$$

where the last line follow from Corollary 2.6.1.

For the term $\|F(x_k, Y_k) - F(x_{k-t_k}, Y_k) + \bar{F}(x_{k-t_k}) - \bar{F}(x_k)\|_c$, using Assumptions 2.2.1 and 2.2.2 and we obtain

$$\begin{aligned}
&\|F(x_k, Y_k) - F(x_{k-t_k}, Y_k) + \bar{F}(x_{k-t_k}) - \bar{F}(x_k)\|_c \\
&\leq \|F(x_k, Y_k) - F(x_{k-t_k}, Y_k)\|_c + \|\bar{F}(x_{k-t_k}) - \bar{F}(x_k)\|_c \\
&\leq 2A_1\|x_k - x_{k-t_k}\|_c
\end{aligned}$$

$$
\begin{aligned}
&\leq 2A\|x_k - x_{k-t_k}\|_c \\
&\leq 8A\alpha_{k-t_k,k-1}(A\|x_k - x^*\|_c + A\|x^*\|_c + B),
\end{aligned}
$$

where in the last line we used Lemma 2.6.2. It follows that

$$
\begin{aligned}
T_{32} &\leq \frac{1}{\ell_{cs}}\|\nabla M_f^{\theta,g}(x_{k-t_k} - x^*)\|_s^*\|F(x_k, Y_k) - F(x_{k-t_k}, Y_k) + \bar{F}(x_{k-t_k}) - \bar{F}(x_k)\|_c \\
&\leq \frac{16L\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}(A\|x_k - x^*\|_c + A\|x^*\|_c + B)^2 \\
&\leq \frac{32LA^2\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}\|x_k - x^*\|_c^2 + \frac{32L\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}(A\|x^*\|_c + B)^2 \\
&\leq \frac{64LA^2u_{cm}^2\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}M_f^{\theta,g}(x_k - x^*) + \frac{32L\alpha_{k-t_k,k-1}}{\theta\ell_{cs}^2}(A\|x^*\|_c + B)^2.
\end{aligned}
$$

(3) Consider the term $T_{33}$. We first take expectation conditioning on $x_{k-t_k}$ and $Y_{k-t_k}$ to obtain

$$
\begin{aligned}
&\mathbb{E}[T_{33} \mid x_{k-t_k}, Y_{k-t_k}] \\
&= \langle\nabla M_f^{\theta,g}(x_{k-t_k} - x^*), \mathbb{E}[F(x_{k-t_k}, Y_k) \mid x_{k-t_k}, Y_{k-t_k}] - \bar{F}(x_{k-t_k})\rangle \\
&\leq \|\nabla M_f^{\theta,g}(x_{k-t_k} - x^*)\|_s^*\|\mathbb{E}[F(x_{k-t_k}, Y_k) \mid x_{k-t_k}, Y_{k-t_k}] - \bar{F}(x_{k-t_k})\|_s \\
&\leq \frac{1}{\ell_{cs}}\|\nabla M_f^{\theta,g}(x_{k-t_k} - x^*)\|_s^*\|\mathbb{E}[F(x_{k-t_k}, Y_k) \mid x_{k-t_k}, Y_{k-t_k}] - \bar{F}(x_{k-t_k})\|_c.
\end{aligned}
$$

For the term $\|\nabla M_f^{\theta,g}(x_{k-t_k} - x^*)\|_s^*$, we have from Equation 2.13 that

$$
\|\nabla M_f^{\theta,g}(x_{k-t_k} - x^*)\|_s^* \leq \frac{2L}{\theta\ell_{cs}}\left(\|x_k - x^*\|_c + \|x^*\|_c + \frac{B}{A}\right).
$$

For the term $\|\mathbb{E}[F(x_{k-t_k}, Y_k) \mid x_{k-t_k}, Y_{k-t_k}] - \bar{F}(x_{k-t_k})\|_c$, using the geometric mixing of the Markov chain $\{Y_k\}$ (cf. Assumption 2.2.3), we have

$$
\begin{aligned}
&\|\mathbb{E}[F(x_{k-t_k}, Y_k) \mid x_{k-t_k}, Y_{k-t_k}] - \bar{F}(x_{k-t_k})\|_c \\
= \;&\|\mathbb{E}[F(x_{k-t_k}, Y_k) \mid x_{k-t_k}, Y_{k-t_k}] - \mathbb{E}_{Y\sim\mu_Y}[F(x_{k-t_k}, Y)]\|_c
\end{aligned}
$$

$$
\begin{aligned}
&= \left\|\sum_{y\in\mathcal{Y}}\left(P_Y^{t_k}(Y_{k-t_k},y)-\mu_Y(y)\right)F(x_{k-t_k},y)\right\|_c\\
&\leq \sum_{y\in\mathcal{Y}}\left|P_Y^{t_k}(Y_{k-t_k},y)-\mu_Y(y)\right|\|F(x_{k-t_k},y)\|_c\\
&\leq 2\max_{y_0\in\mathcal{Y}}\|P_Y^{t_k}(y_0,\cdot)-\mu_Y(\cdot)\|_{\text{TV}}(A_1\|x_{k-t_k}\|_c+B_1)\\
&\leq 2C\sigma^{t_k}(A_1\|x_k-x_{k-t_k}\|_c+A_1\|x_k\|_c+B_1) && \text{(Assumption 2.2.3)}\\
&\leq 2\alpha_k(A_1(\|x_k\|_c+B/A)+A_1\|x_k\|_c+B_1) && \text{(Definition of } t_k \text{ and Corollary 2.6.1)}\\
&\leq 4\alpha_k(A\|x_k-x^*\|_c+A\|x^*\|_c+B).
\end{aligned}
$$

It follows that

$$
\begin{aligned}
&\mathbb{E}[T_{33}\mid x_{k-t_k},Y_{k-t_k}]\\
\leq\;& \frac{1}{\ell_{cs}}\|\nabla M_f^{\theta,g}(x_{k-t_k}-x^*)\|_s^*\|\mathbb{E}[F(x_{k-t_k},Y_k)\mid x_{k-t_k},Y_{k-t_k}]-\bar{F}(x_{k-t_k})\|_c\\
\leq\;& \frac{8L\alpha_k}{\theta\ell_{cs}^2}(A\|x_k-x^*\|_c+A\|x^*\|_c+B)^2\\
\leq\;& \frac{16L\alpha_k}{\theta\ell_{cs}^2}A^2\|x_k-x^*\|_c^2+\frac{16L\alpha_k}{\theta\ell_{cs}^2}(A\|x^*\|_c+B)^2\\
\leq\;& \frac{32LA^2u_{cm}^2\alpha_k}{\theta\ell_{cs}^2}M_f^{\theta,g}(x_k-x^*)+\frac{16L\alpha_k}{\theta\ell_{cs}^2}(A\|x^*\|_c+B)^2.
\end{aligned}
$$

Taking the total expectation on both sides of the previous inequality gives the desired result.

### 2.7.4 Proof of Lemma 2.6.5

Using Proposition 2.3.1 (2), Assumption 2.2.2, and Assumption 2.2.4, we have

$$
\begin{aligned}
T_4 &= \frac{L\alpha_k^2}{2\theta}\|F(x_k,Y_k)-x_k+w_k\|_s^2\\
&\leq \frac{L\alpha_k^2}{2\theta\ell_{cs}^2}\|F(x_k,Y_k)-x_k+w_k\|_c^2 && \text{(Proposition 2.3.1 (3))}\\
&\leq \frac{L\alpha_k^2}{2\theta\ell_{cs}^2}(\|F(x_k,Y_k)\|_c+\|x_k\|_c+\|w_k\|_c)^2
\end{aligned}
$$

$$
\begin{aligned}
&\leq \frac{L\alpha_k^2}{2\theta\ell_{cs}^2}(A\|x_k\|_c + B)^2 \qquad\qquad \text{(Assumptions 2.2.4 and 2.2.2)}\\
&\leq \frac{L\alpha_k^2}{2\theta\ell_{cs}^2}(A\|x_k - x^*\|_c + A\|x^*\|_c + B)^2\\
&\leq \frac{L\alpha_k^2}{\theta\ell_{cs}^2}A^2\|x_k - x^*\|_c^2 + \frac{L\alpha_k^2}{\theta\ell_{cs}^2}(A\|x^*\|_c + B)^2\\
&\leq \frac{2LA^2u_{cm}^2\alpha_k^2}{\theta\ell_{cs}^2}M_f^{\theta,g}(x_k - x^*) + \frac{L\alpha_k^2}{\theta\ell_{cs}^2}(A\|x^*\|_c + B)^2.
\end{aligned}
$$

### 2.7.5 Proof of Lemma 2.6.6

We first simplify the RHS of Equation 2.8 using $\alpha_k = \frac{\alpha}{k+h}$. Since we have chosen $h$ such that $\alpha_{k-t_k,k-1} \leq 2\alpha_k$ for any $k \geq t_k$, Equation 2.8 implies

$$
\begin{aligned}
\mathbb{E}[\|x_k - x^*\|_c^2] &\leq \varphi_1 c_1 \prod_{j=K}^{k-1}(1-\varphi_2\alpha_j) + \varphi_3 c_2 \sum_{i=K}^{k-1}\alpha_i\alpha_{i-t_i,i-1}\prod_{j=i+1}^{k-1}(1-\varphi_2\alpha_j)\\
&\leq \varphi_1 c_1 \prod_{j=K}^{k-1}(1-\varphi_2\alpha_j) + 2\varphi_3 c_2 \sum_{i=K}^{k-1}\alpha_i^2 t_i\prod_{j=i+1}^{k-1}(1-\varphi_2\alpha_j)\\
&= \varphi_1 c_1 \underbrace{\prod_{j=K}^{k-1}\left(1-\frac{\varphi_2\alpha}{j+h}\right)}_{E_1} + 2\varphi_3 c_2 t_k \underbrace{\sum_{i=K}^{k-1}\frac{\alpha^2}{(i+h)^2}\prod_{j=i+1}^{k-1}\left(1-\frac{\varphi_2\alpha}{j+h}\right)}_{E_2}
\end{aligned}
\tag{2.14}
$$

For the term $E_1$, we have

$$
E_1 \leq \exp\left(-\varphi_2\alpha\sum_{j=K}^{k-1}\frac{1}{j+h}\right) \leq \exp\left(-\varphi_2\alpha\int_K^k \frac{1}{x+h}dx\right) = \left(\frac{K+h}{k+h}\right)^{\varphi_2\alpha}.
$$

Now consider the term $E_2$. Similarly we have

$$
\begin{aligned}
E_2 &= \sum_{i=K}^{k-1}\frac{\alpha^2}{(i+h)^2}\prod_{j=i+1}^{k-1}\left(1-\frac{\varphi_2\alpha}{j+h}\right)\\
&\leq \sum_{i=K}^{k-1}\frac{\alpha^2}{(i+h)^2}\left(\frac{i+1+h}{k+h}\right)^{\varphi_2\alpha}
\end{aligned}
$$

$$
\begin{aligned}
&\leq \frac{4\alpha^2}{(k+h)^{\varphi_2\alpha}} \sum_{i=K}^{k-1} \frac{1}{(i+1+h)^{2-\varphi_2\alpha}} \\
&\leq \begin{cases} \dfrac{4\alpha^2}{1-\varphi_2\alpha} \dfrac{1}{(k+h)^{\varphi_2\alpha}}, & \varphi_2\alpha \in (0,1), \\ \dfrac{4\alpha^2 \log(k+h)}{k+h}, & \varphi_2\alpha = 1, \\ \dfrac{4e\alpha^2}{\varphi_2\alpha - 1} \dfrac{1}{k+h}, & \varphi_2\alpha \in (1,\infty). \end{cases}
\end{aligned}
$$

The result then follows from using the upper bounds we obtained for the terms $E_1$ and $E_2$ in Equation 2.14.

### 2.7.6 Proof of Lemma 2.6.7

When $\alpha_k = \frac{\alpha}{(k+h)^\xi}$, similarly we have from Equation 2.8 that

$$
\mathbb{E}[\|x_k - x^*\|_c^2] \leq \varphi_1 c_1 \underbrace{\prod_{j=K}^{k-1} \left(1 - \frac{\varphi_2\alpha}{(j+h)^\xi}\right)}_{E_1} + 2\varphi_3 c_2 t_k \underbrace{\sum_{i=K}^{k-1} \frac{\alpha^2}{(i+h)^{2\xi}} \prod_{j=i+1}^{k-1} \left(1 - \frac{\varphi_2\alpha}{(j+h)^\xi}\right)}_{E_2} \tag{2.15}
$$

The term $E_1$ can be controlled in the following way:

$$
\begin{aligned}
E_1 &= \prod_{j=K}^{k-1} \left(1 - \frac{\varphi_2\alpha}{(j+h)^\xi}\right) \\
&\leq \exp\left(-\varphi_2\alpha \sum_{j=K}^{k-1} \frac{1}{(j+h)^\xi}\right) \\
&\leq \exp\left(-\varphi_2\alpha \int_K^k \frac{1}{(x+h)^\xi} dx\right) \\
&= \exp\left[-\frac{\varphi_2\alpha}{1-\xi} \left((k+h)^{1-\xi} - (K+h)^{1-\xi}\right)\right].
\end{aligned}
$$

As for the term $E_2$, we will show by induction that $E_2 \leq \frac{2\alpha}{\varphi_2} \frac{1}{(k+h)^\xi}$ for all $k \geq 0$.

Consider a sequence $\{u_k\}_{k\geq 0}$ (with $u_0 = 0$) defined by

$$u_{k+1} = \left(1 - \varphi_2 \frac{\alpha}{(k+h)^\xi}\right) u_k + \frac{\alpha^2}{(k+h)^{2\xi}}, \quad \forall\, k \geq 0.$$

It can be easily verified that $u_k = E_2$. Since $u_0 = 0 \leq \frac{2\alpha}{\varphi_2}\frac{1}{h^\xi}$, we have the base case. Now suppose $u_k \leq \frac{2\alpha}{\varphi_2}\frac{1}{(k+h)^\xi}$ for some $k > 0$. Consider $u_{k+1}$, and we have

$$\begin{aligned}
\frac{2\alpha}{\varphi_2}\frac{1}{(k+1+h)^\xi} - u_{k+1} =& \frac{2\alpha}{\varphi_2}\frac{1}{(k+1+h)^\xi} - \left(1 - \varphi_2 \frac{\alpha}{(k+h)^\xi}\right) u_k + \frac{\alpha^2}{(k+h)^{2\xi}} \\
\geq& \frac{2\alpha}{\varphi_2}\frac{1}{(k+1+h)^\xi} - \left(1 - \frac{\varphi_2\alpha}{(k+h)^\xi}\right)\frac{2\alpha}{\varphi_2}\frac{1}{(k+h)^\xi} - \frac{\alpha^2}{(k+h)^{2\xi}} \\
=& \frac{2\alpha}{\varphi_2}\left[\frac{1}{(k+1+h)^\xi} - \frac{1}{(k+h)^\xi} + \frac{\varphi_2\alpha}{2}\frac{1}{(k+h)^{2\xi}}\right] \\
=& \frac{2\alpha}{\varphi_2}\frac{1}{(k+h)^{2\xi}}\left[\frac{\varphi_2\alpha}{2} - (k+h)^\xi\left(1 - \left(\frac{k+h}{k+1+h}\right)^\xi\right)\right].
\end{aligned}$$

Note that

$$\left(\frac{k+h}{k+1+h}\right)^\xi = \left[\left(1 + \frac{1}{k+h}\right)^{k+h}\right]^{-\frac{\xi}{k+h}} \geq \exp\left(-\frac{\xi}{k+h}\right) \geq 1 - \frac{\xi}{k+h},$$

where we used $(1+\frac{1}{x})^x < e$ for all $x > 0$ and $e^x \geq 1 + x$ for all $x \in \mathbb{R}$. Therefore, we obtain

$$\begin{aligned}
\frac{2\alpha}{\varphi_2}\frac{1}{(k+1+h)^\xi} - u_{k+1} &\geq \frac{2\alpha}{\varphi_2}\frac{1}{(k+h)^{2\xi}}\left[\frac{\varphi_2\alpha}{2} - (k+h)^\xi\left(1 - \left(\frac{k+h}{k+1+h}\right)^\xi\right)\right] \\
&\geq \frac{2\alpha}{\varphi_2}\frac{1}{(k+h)^{2\xi}}\left[\frac{\varphi_2\alpha}{2} - \frac{\xi}{(k+h)^{1-\xi}}\right] \\
&\geq 0,
\end{aligned}$$

where the last line follows from $h \geq \bar{h} \geq [2\xi/(\varphi_2\alpha)]^{1/(1-\xi)}$. The induction is now complete, and we have $E_2 \leq \frac{2\alpha}{\varphi_2}\frac{1}{(k+h)^\xi}$ for all $k \geq 0$. Using the upper bounds we obtained for the terms $E_1$ and $E_2$ in Equation 2.15 and we have the desired result.

### 2.7.7 Proof of Corollary 2.5.1

(1) When $\|\cdot\|_c = \|\cdot\|_2$, we choose $\theta = 1$ and $g(x) = \frac{1}{2}\|x\|_2^2$. It follows that $L = 1$ and $u_{cs} = \ell_{cs} = 1$. Therefore, we have $\varphi_1 = 1$, $\varphi_2 = 1 - \beta$, and $\varphi_3 = 228$.

(2) Recall the definition of $\{\varphi_i\}_{1\leq i\leq 3}$ in the beginning of Section 2.4. When $\|\cdot\|_c = \|\cdot\|_\infty$, we choose $\theta = \left(\frac{1+\beta}{2\beta}\right)^2 - 1$ and $g(x) = \frac{1}{2}\|x\|_p^2$ with $p = 2\log(d)$, where $d$ is the dimension of the iterates $\{x_k\}$. It follows that $L = p - 1 \leq 2\log(d)$ [46], $u_{cs} = 1$, and $\ell_{cs} = 1/d^{1/p} = 1/\sqrt{e}$. Therefore, we have

$$\begin{aligned}
\varphi_1 &= \frac{1+\theta u_{cs}^2}{1+\theta\ell_{cs}^2} = \frac{1+\theta}{1+\theta/\sqrt{e}} \leq \sqrt{e} \leq 3,\\
\varphi_2 &= 1 - \beta\varphi_1^{1/2} \geq 1 - \beta\frac{1+\beta}{2\beta} = \frac{1-\beta}{2},\\
\varphi_3 &= \frac{114L(1+\theta u_{cs}^2)}{\theta\ell_{cs}^2} \leq \frac{228e\log(d)(1+\theta)}{\theta} \leq \frac{456e\log(d)}{1-\beta}.
\end{aligned}$$

## 2.8 Conclusion and Future Work

In this chapter, we have established finite-sample bounds for Markovian SA algorithms involving contraction operators with respect arbitrary norms. We prove this result using a novel Lyapunov function. Such a a smooth Lyapunov function is constructed using the generalized Moreau envelope, which involves the infimal convolution with respect to the square of some other suitable norm.

Beyond mean-square bounds, sometimes high probability (concentration) bounds are more preferred for practical applications, which is our immediate future direction of this line of work.

# CHAPTER 3
# STOCHASTIC APPROXIMATION UNDER A STRONGLY PSEUDO-MONOTONE OPERATOR

In the previous chapter we studied Markovian SA algorithms under contractive operators. In this chapter, we consider a different Markovian SA algorithm, where instead of having contractive operators, we have strongly pseudo-monotone operators. The results in this chapter will be used to design and analyze convergent RL algorithms in the presence of the deadly triad in Part III.

## 3.1 Problem Setting

Consider the problem of solving for $x^* \in \mathbb{R}^d$ in the equation

$$\bar{G}(x) = \mathbb{E}_{\mu_Y}[G(x, Y)] = 0, \tag{3.1}$$

where $Y \in \mathcal{Y}$ is a random vector with distribution $\mu_Y$, and $G : \mathbb{R}^d \times \mathcal{Y} \mapsto \mathbb{R}^d$ is a general nonlinear operator. Similarly as in Chapter 2, we assume the set $\mathcal{Y}$ is finite.

When the distribution $\mu_Y$ is unknown, Equation 3.1 cannot be solved analytically. Therefore, we consider solving the equation using the SA method presented in the following. Here in Algorithm 2, $\{Y_k\}$ is a uniformly ergodic Markov chain with stationary distribution $\mu_Y$, $\{w_k\}$ represents the additive martingale difference noise that possibly depends on $\{x_k\}$, and $\{\alpha_k\}$ is the stepsize sequence.

We next state our assumptions to study Algorithm 2. Before that, the following definition is needed. Recall that we use $\|\cdot\|_2$ for the $\ell_2$-norm for vectors.

**Definition 3.1.1.** An operator $F : \mathbb{R}^d \mapsto \mathbb{R}^d$ is said to be *strongly pseudo-monotone* with

**Algorithm 2** SA under a Strongly Pseudo-Monotone Operator

```
Input: Integer $k'$, and initialization $x_0 \in \mathbb{R}^d$
for $k = 0, 1, \cdots, k' - 1$ do
    $x_{k+1} = x_k + \alpha_k(G(x_k, Y_k) + w_k)$
end for
Output: $x_{k'}$
```

respect to $\bar{x} \in \mathbb{R}^d$ if there exists $c_0 > 0$ such that

$$(x - \bar{x})^\top (F(x) - F(\bar{x})) \geq c_0 \|x - \bar{x}\|_2^2, \ \forall \, x \in \mathbb{R}^d.$$

*Remark.* Recall that an operator $F : \mathbb{R}^d \mapsto \mathbb{R}^d$ being strongly monotone if and only if there exists $c_0' > 0$ such that $(x - y)^\top (F(x) - F(y)) \geq c_0' \|x - y\|_2^2$ for all $x, y \in \mathbb{R}^d$. Therefore, an operator being strongly monotone implies it being strongly pseudo-monotone with respect to any point.

**Assumption 3.1.1.** The target equation $\bar{G}(x) = 0$ has a unique solution, which we denote by $x^*$. In addition, the operator $-\bar{G}(\cdot)$ is strongly pseudo-monotone with respect to $x^*$, i.e., there exists $\kappa > 0$ such that

$$(x - x^*)^\top (\bar{G}(x) - \bar{G}(x^*)) \leq -\kappa \|x - x^*\|_2^2, \ \forall \, x \in \mathbb{R}^d.$$

In the SGD setting (i.e., $G(x, y) = -\nabla J(x) + y$ for some cost function $J(\cdot)$), Assumption 3.1.1 is satisfied when the objective function $J(\cdot)$ is strongly convex. Moreover, Assumption 3.1.1 can be viewed as an exponential dissipativeness property of the following ODE

$$\dot{x}(t) = \bar{G}(x(t)), \tag{3.2}$$

which is the ODE associated with Algorithm 2 [31]. In fact, this assumption guarantees that $x^*$ is the unique exponentially stable equilibrium point of the ODE. To see this, let

$W(x) = \|x - x^*\|_2^2$ be a candidate Lyapunov function. Then we have by Assumption 3.1.1 that

$$\frac{d}{dt}W(x(t)) = 2(x(t) - x^*)^\top \dot{x}(t) \leq -2\kappa W(x(t)), \tag{3.3}$$

which implies that $W(x(t)) \leq W(x(0))e^{-2\kappa t}$ for all $t \geq 0$. The parameter $\kappa$ is called the *negative drift*, and we see that the larger $\kappa$ is, the faster $x(t)$ converges.

**Assumption 3.1.2.** The following statements hold.

(1) *Lipchitz Continuity.* There exists constant $L_1 > 0$ such that

 (a) $\|G(x_1, y) - G(x_2, y)\| \leq L_1\|x_1 - x_2\|$ for all $x_1, x_2 \in \mathbb{R}^d$ and $y \in \mathcal{Y}$,

 (b) $\|G(\mathbf{0}, y)\| \leq L_1$ for all $y \in \mathcal{Y}$.

(2) *Uniform Ergodicity.* The Markov chain $\{Y_k\}$ is uniformly geometrically ergodic with unique stationary distribution $\mu_Y$.

(3) *Additive Martingale Difference Noise.* The random process $\{w_k\}$ satisfies

 (a) $\mathbb{E}[w_k \mid \mathcal{F}_k] = 0$ for all $k \geq 0$, where $\mathcal{F}_k$ be the Sigma-algebra generated by $\{x_i, Y_i, w_i\}_{0 \leq i \leq k-1} \cup \{x_k\}$,

 (b) $\|w_k\|_2 \leq L_2(\|x_k\|_2 + 1)$ for all $k \geq 0$, where $L_2 > 0$ is a constant.

Assumption 3.1.2 is analogous to Assumptions 2.2.2, 2.2.3, and 2.2.4 given in Chapter 2. Specifically, Assumption 3.1.2 (1) states that the operator $G(x, y)$ is $L_1$-Lipschitz continuous with respect to $x$ uniformly in $y$. In the special case where $G(x, y)$ is a linear function of $x$ as considered in [40, 12], i.e., $G(x, y) = A(y)x + b(y)$, Assumption 3.1.2 (1) is automatically satisfied. Assumption 3.1.2 (2) is made to control the Markovian noise in Algorithm 2, and implies that there exist $C \geq 1$ and $\sigma \in (0, 1)$ such that $\max_{y \in \mathcal{Y}} \|P_Y^k(y, \cdot) - \mu_Y(\cdot)\|_{\text{TV}} \leq C\sigma^k$ for all $k \geq 0$. Using the definition of mixing time (cf. Definition 2.2.1), Assumption 3.1.2 (2) implies $t_\delta \leq L_3(\log(1/\delta) + 1)$, where

$L_3 = \frac{\log(C/\sigma)}{\log(1/\sigma)}$. Assumption 3.1.2 (3) states that $\{w_k\}$ is a martingale difference sequence, and $\|w_k\|_2$ at most scales affinely with respect to the latest iterate $\|x_k\|_2$.

Finally we state the requirement for choosing the stepsize sequence. Recall that we denote $t_k = t_{\alpha_k}$ and $\alpha_{i,j} = \sum_{k=i}^{j} \alpha_k$. Let $L = L_1 + L_2$, and assume without loss of generality that $L \geq 1$.

**Condition 3.1.1.** The stepsize sequence $\{\alpha_k\}$ satisfies the following conditions:

(1) $\{\alpha_k\}$ is non-increasing and $\alpha_0 \in (0, 1)$,

(2) it holds that $\alpha_{k-t_k,k-1} < \frac{\kappa}{130L^2}$ for all $k \geq t_k$.

The reason we impose Condition 3.1.1 on the stepsize sequence is the following. Recall that a key step in deriving the convergence rate of the ODE given in Equation 3.2 is to establish the negative drift (cf. Equation 3.3). Similarly, when deriving finite-sample bounds for Algorithm 2, there will also be a negative drift term. In addition, there are error terms that arise because of the discretization and the stochastic noise. Using small stepsize helps suppressing these error terms and hence ensures that the negative drift is the dominant term in our analysis.

Suppose we use constant stepsize, i.e., $\alpha_k = \alpha$ for all $k \geq 0$. Since in this case we have $\alpha_{k-t_k,k-1} = \alpha t_\alpha$, and $\lim_{\alpha \to 0} \alpha t_\alpha = 0$, Condition 3.1.1 is satisfied when $\alpha$ is small enough. In addition to constant stepsize, when using polynomially diminishing stepsizes of the form $\alpha_k = \alpha/(k+h)^\xi$, Condition 3.1.1 is satisfied for any $\alpha > 0$ and $\xi \in (0, 1]$, provided that $h$ is appropriately chosen.

### 3.2 Finite-Sample Convergence Guarantees

In this section, we present the finite-sample bounds of Algorithm 2.

**Theorem 3.2.1.** *Consider $\{x_k\}$ of Algorithm 2. Suppose that Assumptions 3.1.1 and 3.1.2 are satisfied, and $\{\alpha_k\}$ satisfies Condition 3.1.1. Let $K = \min\{k : k \geq t_k\}$. Then we have* *for all $k \geq K$:*

$$\mathbb{E}[\|x_k - x^*\|_2^2] \leq c_1 \prod_{j=K}^{k-1}(1-\kappa\alpha_j) + c_2 \sum_{i=K}^{k-1} \hat{\alpha}_i \prod_{j=i+1}^{k-1}(1-\kappa\alpha_j), \tag{3.4}$$

*where $c_1 = (\|x_0\|_2 + \|x_0 - x^*\|_2 + 1)^2$, $c_2 = 130L^2(\|x^*\|_2 + 1)^2$, and $\hat{\alpha}_i = \alpha_i \alpha_{i-t_i,i-1}$.*

*Remark.* Although the parameter $K$ is defined as $K = \min\{k : k \geq t_k\}$, we indeed have $K = t_K$. To see this, suppose that $K > t_K$. Since both $K$ and $t_K$ are integers, we must have $K - 1 \geq t_K \geq t_{K-1}$, where the second inequality follows from the fact that $t_k = t_{\alpha_k}$ is an increasing function of $k$. This contradict to the definition of $K$ and hence we have $K = t_K$.

On the RHS of Equation 3.4, the first term represents the bias due to the initial guess $x_0$, and the second term captures the variance due to the noise. After establishing the finite-sample bounds of Algorithm 2 in its general form, we next consider using stepsizes of the form $\alpha_k = \frac{\alpha}{(k+h)^\xi}$ and see more explicitly the corresponding convergence rates.

**Corollary 3.2.1.** *We have the following finite-sample guarantees.*

*(1) Constant Stepsize. When $\xi = 0$ and $\alpha$ is chosen such that $\alpha t_\alpha \leq \frac{\kappa}{130L^2}$, we have*

$$\mathbb{E}[\|x_k - x^*\|_2^2] \leq c_1(1-\kappa\alpha)^{k-t_\alpha} + c_2\frac{\alpha t_\alpha}{\kappa}, \ \forall\, k \geq t_\alpha.$$

*(2) Linearly Diminishing Stepsizes. When $\xi = 1$ and $\alpha > 1/\kappa$, we have for all $k \geq K$:*

$$\mathbb{E}[\|x_k - x^*\|_2^2] \leq c_1 \left(\frac{K+h}{k+h}\right)^{\kappa\alpha} + \frac{8ec_2\alpha^2 L_3}{\kappa\alpha - 1}\frac{\left[\log\left(\frac{k+h}{\alpha}\right)+1\right]}{k+h}.$$

*(3) Polynomialy Diminishing Stepsizes. When $\xi \in (0,1)$ and $\alpha > 0$, assume without loss of generality that $K \geq [2\xi/(\kappa\alpha)]^{1/(1-\xi)}$, then we have for all $k \geq K$:*

$$\mathbb{E}[\|x_k - x^*\|_2^2] \leq c_1 e^{-\frac{\kappa\alpha}{1-\xi}\left((k+h)^{1-\xi}-(K+h)^{1-\xi}\right)} + \frac{4c_2\alpha L_3}{\kappa}\frac{\left[\log\left(\frac{k+h}{\alpha}\right)+1\right]}{(k+h)^\xi}.$$

Corollary 3.2.1 is qualitatively similar to Theorem 2.5.1 in Chapter 2. Specifically, using constant stepsize results in constant variance and geometric decaying bias, and using diminishing stepsizes results in both decaying variance and decaying bias (albeit at a slower decay rate).

Unlike almost sure convergence, where the usual requirement for stepsizes are $\sum_{k=0}^{\infty} \alpha_k = \infty$ and $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$ (which corresponds to $\xi \in (1/2, 1]$ in our case), we have convergence in the mean-square sense for all $\xi \in (0, 1]$. The same phenomenon has been observed in [40], where they study linear SA and nonlinear SA with martingale difference noise.

## 3.3 Proof of All Theoretical Results

In this section, we present the proofs of Theorem 3.2.1 and Corollary 3.2.1.

### 3.3.1 Proof of Theorem 3.2.1

We prove Theorem 3.2.1 using a Lyapunov approach with $W(x) = \|x - x^*\|_2^2$ being the Lyapunov function. Using the update equation of Algorithm 2 and we have for all $k \geq 0$ that

$$\begin{aligned} \|x_{k+1} - x^*\|_2^2 - \|x_k - x^*\|_2^2 &= 2(x_k - x^*)^\top (x_{k+1} - x_k) + \|x_{k+1} - x_k\|_2^2 \\ &= \underbrace{2\alpha_k (x_k - x^*)^\top \bar{G}(x_k)}_{T_1} + \underbrace{2\alpha_k (x_k - x^*)^\top w_k}_{T_2} \\ &\quad + \underbrace{2\alpha_k (x_k - x^*)^\top (G(x_k, Y_k) - \bar{G}(x_k))}_{T_3} \\ &\quad + \underbrace{\alpha_k^2 \|G(x_k, Y_k) + w_k\|_2^2]}_{T_4}. \end{aligned} \tag{3.5}$$

The term $T_1$ corresponds to the negative drift of the ODE given in Equation 3.2, and we have by Assumption 3.1.1 that

$$T_1 \leq -2\kappa\alpha_k\|x_k - x^*\|_2^2.$$

The term $T_2$ corresponds to the error due to martingale difference noise $\{w_k\}$. Using the tower property of conditional expectation and we have $T_2 = 0$.

The term $T_3$ corresponds to the error due to the Markovian noise $\{Y_k\}$, and the term $T_4$ arises mainly because of the error due to discretization. We next control the terms $T_3$ and $T_4$ in the following sequence of lemmas. Their proofs are identical to that of Lemmas 2.6.2, 2.6.4 and 2.6.5 in Chapter 2 and hence is omitted.

**Lemma 3.3.1.** *For any $k_1 < k_2$ satisfying $\alpha_{k_1,k_2-1} \leq \frac{1}{4L}$, the following two inequalities hold:*

*(1)* $\|x_{k_2} - x_{k_1}\|_2 \leq 2L\alpha_{k_1,k_2-1}(\|x_{k_1}\|_2 + 1)$,

*(2)* $\|x_{k_2} - x_{k_1}\|_2 \leq 4L\alpha_{k_1,k_2-1}(\|x_{k_2}\|_2 + 1)$.

**Lemma 3.3.2.** *The following inequality holds for all $k$ such that $\alpha_{k-t_k,k-1} \leq \frac{1}{4L}$ (where we recall that $\alpha_{k-t_k,k-1} = \sum_{i=k-t_k}^{k-1} \alpha_i$):*

$$\mathbb{E}[T_3] \leq 128L^2\alpha_k\alpha_{k-t_k,k-1}\left[\mathbb{E}[\|x_k - x^*\|_2^2] + (\|x^*\|_2 + 1)^2\right].$$

**Lemma 3.3.3.** *The following inequality holds for all $k \geq t_k$:*

$$T_4 \leq 2L^2\alpha_k^2\left[\|x_k - x^*\|_2^2 + (\|x^*\|_2 + 1)^2\right].$$

Substituting the upper bounds we obtained for the terms $T_1$ to $T_4$ into Equation 3.5, we have the following recursive bound.

**Lemma 3.3.4.** *It holds for all $k$ satisfying $\alpha_{k-t_k,k-1} \leq \frac{1}{4L}$ that:*

$$\mathbb{E}[\|x_{k+1} - x^*\|_2^2] \leq (1 - 2\kappa\alpha_k + 130L^2\hat{\alpha}_k)\mathbb{E}[\|x_k - x^*\|_2^2] + 130L^2\hat{\alpha}_k(\|x^*\|_2 + 1)^2, \tag{3.6}$$

*where we recall that $\hat{\alpha}_k = \alpha_k\alpha_{k-t_k,k-1}$.*

In view of Equation 3.6, as long as the drift term dominates the error terms, i.e., $2\kappa\alpha_k > 130L^2\hat{\alpha}_k$, we can repeatedly use Equation 3.6 to derive finite-sample error bounds of Algorithm 2. In particular, when Condition 3.1.1 is satisfied and $k \geq K$ (see Theorem 3.2.1 for the definition of $K$), we have by Equation 3.6 that

$$\mathbb{E}[\|x_{k+1} - x^*\|_2^2] \leq (1 - \kappa\alpha_k)\mathbb{E}[\|x_k - x^*\|_2^2] + c_2\hat{\alpha}_k,$$

where $c_2$ is defined in Theorem 3.2.1. Repeatedly using the preceding inequality starting from $K$, we obtain

$$\mathbb{E}[\|x_k - x^*\|_2^2] \leq \mathbb{E}[\|x_K - x^*\|_2^2] \prod_{j=K}^{k-1}(1 - \kappa\alpha_j) + c_2\sum_{i=K}^{k-1}\hat{\alpha}_i \prod_{j=i+1}^{k-1}(1 - \kappa\alpha_j).$$

To bound $\mathbb{E}[\|x_K - x^*\|_2^2]$, we use Lemma 3.3.1 and $\alpha_{K-t_K,K-1} = \alpha_{0,K-1} \leq \frac{1}{4L}$ to obtain

$$\mathbb{E}[\|x_K - x^*\|_2^2] \leq \mathbb{E}[(\|x_K - x_0\|_2 + \|x^* - x_0\|_2)^2] \leq c_1.$$

This completes the proof.

### 3.3.2 Proof of Corollary 3.2.1

The result is obtained by evaluating the RHS of Equation 3.4 when the stepsize sequence $\{\alpha_k\}$ is explicitly chosen. The proof is identical to that of Theorem 2.5.1 after Equation 2.8, and hence is omitted.

## 3.4 Conclusion

In this chapter we have established finite-sample convergence guarantees for Markovian SA algorithms under strongly pseudo-monotone operators. Specifically, we have shown that the optimal convergence rate is $\mathcal{O}(1/k)$ with appropriately chosen diminishing stepsizes. The rate matches with the results in Chapter 2 for contractive SA, and that of SGD with a smooth and strongly convex objective.

The results in this chapter will be frequently used in Part III of the thesis to study RL with linear function approximation.

# CHAPTER 4
# STATIONARY BEHAVIOR OF STOCHASTIC APPROXIMATION ALGORITHMS

In the previous two chapters, we have characterized the finite-sample behavior of SA algorithms under contractive operators and under strongly pseudo-monotone operators. In particular, we have shown that using constant stepsize leads to geometric convergence (in the mean-square sense) to a ball centered at the desired limit point, and using diminishing stepsize leads to asymptotic convergence at a polynomial rate. In this chapter, we switch our focus to the asymptotic region of constant stepsize SA algorithms, and characterize the stationary distribution of properly scaled iterates as the constant stepsize goes to zero.

## 4.1 Introduction

As we have shown in the previous two chapters, theoretically, to achieve asymptotic convergence, we should use diminishing stepsizes with proper decay rate [28, 11, 33]. However, constant stepsize SA algorithms are preferred in practice due to their faster convergence. In that case, instead of converging asymptotically to the desired solution, the iterates of constant stepsize SA algorithms have a stationary distribution. Although such weak convergence to a stationary distribution was established in the literature [55], it is not possible to fully characterize the limiting distribution. The reason is that, when constant stepsize is used, the distribution of the noise sequence within the SA algorithm plays an important role in the stationary distribution of the iterates. Since the distribution of the noise is in general unknown, the stationary distribution cannot be analytically characterized. In this chapter, building upon the works on stationary distribution of constant stepsize SA algorithms, we aim at understanding the limiting behavior of the properly scaled stationary distribution as the constant stepsize goes to zero.

More formally, with initialization $X_0^{(\alpha)} \in \mathbb{R}^d$, consider the SA algorithm

$$X_{k+1}^{(\alpha)} = X_k^{(\alpha)} + \alpha \left( F(X_k^{(\alpha)}) + w_k \right), \tag{4.1}$$

where $F : \mathbb{R}^d \mapsto \mathbb{R}^d$ is a general nonlinear operator, $\alpha$ is the constant stepsize, and $\{w_k\}$ is the noise sequence. Observe that Equation 4.1 can be viewed as an iterative algorithm for solving the equation $F(x) = 0$ in the presence of noise [10]. A typical example is when $F(x) = -c\nabla f(x)$ (where $c > 0$ is a constant) for some objective function $f(\cdot)$. In this case Equation 4.1 becomes the popular SGD algorithm for minimizing $f(\cdot)$ [23, 43]. Another example lies in the context of RL, where $F(x) = \mathcal{T}(x) - x$, and $\mathcal{T}(\cdot)$ is the Bellman operator [1]. In this case, Equation 4.1 is closely related popular RL algorithms such as TD-learning [56] and $Q$-learning [17].

Under some mild conditions on the operator $F(\cdot)$, it was shown in the literature that the sequence $\{X_k^{(\alpha)}\}$ converges weakly to some random variable $X^{(\alpha)}$ [57, 58, 55, 59]. However, for a fixed $\alpha$, it is not possible to fully characterize the distribution of $X^{(\alpha)}$ because it depends on the distribution of the noise sequence $\{w_k\}$, which is usually unknown. In this chapter, we further consider letting $\alpha$ go to zero, and study the distribution of a properly centered and scaled iterate. Specifically, let $Y_k^{(\alpha)} := (X_k^{(\alpha)} - x^*)/g(\alpha)$, where $x^*$ is the solution of $F(x) = 0$ (provided that it exists and is unique), and $g : \mathbb{R} \mapsto \mathbb{R}$ is a properly chosen scaling function[1]. When $k$ goes to infinity, we expect that $Y_k^{(\alpha)}$ converges weakly to some random variable $Y^{(\alpha)}$. Then we let $\alpha$ go to zero, and our goal is to further characterize the weak limit of $Y^{(\alpha)}$. Notice that proper scaling of the iterates is essential for raveling its fine grade behavior because otherwise the limiting distribution of the un-scaled iterates will converge to a singleton as the stepsize $\alpha$ goes to zero, which is analogous to the almost sure convergence results for using diminishing stepsizes in SA algorithms [10].

To summarize, we want to find a suitable scaling function $g(\cdot)$ and to characterize the following two-step weak convergence of the centered scaled iterate $Y_k^{(\alpha)} = (X_k^{(\alpha)} -$

[1] The scaling function is unique up to a numerical factor

$x^*)/g(\alpha)$:

$$Y_k^{(\alpha)} \overset{k\to\infty}{\Longrightarrow} Y^{(\alpha)} \overset{\alpha\to 0}{\Longrightarrow} Y, \tag{4.2}$$

where we use the notation ⇒ for weak convergence (or convergence in distribution).

### 4.1.1 Main Contributions

In this subsection, we present the main contributions of this chapter.

**Characterizing the Distribution of $\boldsymbol{Y}$.** We propose a general framework for characterizing the distribution of $Y$ in the following 3 cases: (1) SGD with a smooth and strongly convex objective, (2) linear SA with a Hurwitz matrix, and (3) SA involving a contractive operator. In particular, we show that in all three cases above the correct scaling function is $g(\alpha) = \sqrt{\alpha}$, and the distribution of $Y$ is Gaussian with mean zero and covariance matrix being the unique solution of an appropriate Lyapunov equation. Our proof is to use the characteristic function as a test function to obtain an implicit equation of the distribution of $Y$, and then show that the desired Gaussian distribution solves the implicit equation.

**Determining the Suitable Scaling Function.** For more general SA algorithms, we show empirically that the scaling function need not be $g(\alpha) = \sqrt{\alpha}$ and the distribution of $Y$ need not be Gaussian. Inspired by this observation, we propose a method to find the the correct scaling function for general SA algorithms. In particular, our results indicate that the scaling function $g(\alpha)$ should be chosen such that (1) $\lim_{\alpha\to 0} \frac{\alpha}{g(\alpha)} = 0$ and $\lim_{k\to\infty} g(\alpha) = 0$, and (2) the function $\tilde{F}(\cdot)$ defined by $\tilde{F}(y) = \lim_{\alpha\to 0} \frac{g(\alpha)F(yg(\alpha)+x^*)}{\alpha}$ is non-trivial in the sense that it is not identically zero or infinity. Our proposed condition is verified in numerical experiments. Moreover, we make an insightful connection between the choice of the scaling function $g(\alpha)$ and the Euler-Maruyama discretization scheme for approximating stochastic differential equations (SDEs) – Langevin diffusion [60].

### 4.1.2 An Illustrative Example

We next provide an example to illustrate the problem we are going to study. Consider Equation 4.1. Suppose that $F(x) = -x$ is a scalar-valued function, and $\{w_k\}$ is a sequence of i.i.d. standard normal random variables. We make such noise assumption here only for ease of exposition, and it will be relaxed in later sections of this chapter. In this case, Equation 4.1 becomes

$$X_{k+1}^{(\alpha)} = (1-\alpha)X_k^{(\alpha)} + \alpha w_k. \tag{4.3}$$

This algorithm has the following two interpretations: (1) it can be viewed as the SGD algorithm for minimizing the quadratic objective function $f(x) = x^2/2$, which has a unique minimizer at $x^* = 0$, and (2) it can also be viewed as an SA algorithm for solving the fixed-point equation $\mathcal{T}(x) = x$ with $\mathcal{T}(x)$ being identically equal to zero, therefore $x^* = 0$ is the unique fixed-point.

Let $Y_k^{(\alpha)} = X_k^{(\alpha)}/\sqrt{\alpha}$ be the centered scaled iterate. To obtain an update equation for $Y_k^{(\alpha)}$, dividing both sides of Equation 4.3 by $\sqrt{\alpha}$ and we obtain for all $k \geq 0$:

$$\begin{aligned}
Y_k^{(\alpha)} &= (1-\alpha)Y_{k-1}^{(\alpha)} + \sqrt{\alpha}w_{k-1} \\
&= (1-\alpha)^2 Y_{k-2}^{(\alpha)} + (1-\alpha)\sqrt{\alpha}w_{k-2} + \sqrt{\alpha}w_{k-1} \\
&= \cdots \\
&= (1-\alpha)^k Y_0^{(\alpha)} + \sum_{i=0}^{k-1}(1-\alpha)^{k-1-i}\sqrt{\alpha}w_i.
\end{aligned}$$

Since $Y_k^{(\alpha)}$ is a linear combination of mutually independent Gaussian random variables, $Y_k^{(\alpha)}$ itself is also a Gaussian random variable. Therefore, the distribution of $Y_k^{(\alpha)}$ is uniquely determined by its mean and variance. Using the fact that $\{w_k\}$ is an i.i.d. sequence of standard normal random variables, we have

$$\mathbb{E}[Y_k^{(\alpha)}] = (1-\alpha)^k Y_0^{(\alpha)} + \sum_{i=0}^{k-1} (1-\alpha)^{k-1-i} \sqrt{\alpha} \mathbb{E}[w_i] = (1-\alpha)^k Y_0^{(\alpha)},$$

and

$$\begin{aligned}
\mathbb{V}[Y_k^{(\alpha)}] &= \mathbb{V}\left[(1-\alpha)^k Y_0^{(\alpha)} + \sum_{i=0}^{k-1} (1-\alpha)^{k-1-i} \sqrt{\alpha} w_i\right] \\
&= \alpha \sum_{i=0}^{k-1} (1-\alpha)^{2i} \\
&= \frac{1}{2-\alpha} \left(1 - (1-\alpha)^{2k}\right),
\end{aligned}$$

where $\mathbb{V}(\cdot)$ represents the variance of a random variable. It follows that $\lim_{k\to\infty} \mathbb{E}[Y_k^{(\alpha)}] = 0$ and $\lim_{k\to\infty} \mathbb{V}[Y_k^{(\alpha)}] = \frac{1}{2-\alpha}$. Therefore, the sequence $Y_k^{(\alpha)}$ converges weakly to a random variable $Y^{(\alpha)}$, whose distribution is $\mathcal{N}(0, \frac{1}{2-\alpha})$. In this case, we are able to analytically characterize the distribution of $Y^{(\alpha)}$ for a fixed $\alpha$ because of the simplicity of Equation 4.3 and the noise sequence $\{w_k\}$ being i.i.d. standard normal. For Equation 4.1 with limited information on the noise sequence $\{w_k\}$, it is in general not possible to fully characterize the distribution of $Y^{(\alpha)}$.

Now that we have characterized the first weak convergence in Equation 4.2, consider the second weak convergence. Note that we have already shown $Y^{(\alpha)} \sim \mathcal{N}(0, \frac{1}{2-\alpha})$. As $\alpha$ goes to zero, we have that $Y^{(\alpha)}$ converges weakly to a random variable $Y$, whose distribution is $\mathcal{N}(0, \frac{1}{2})$. As opposed to the first weak convergence in Equation 4.2, where the distribution of $Y^{(\alpha)}$ in general cannot be fully characterized, we are able to characterize (in later sections) the distribution of $Y$ for more general SA algorithms under more general noise assumptions. Intuitively, the reason is that as the constant stepsize decreases, the effect of the entire distribution of the noise $\{w_k\}$ on the distribution of $Y^\alpha$ is weakened. This is analogous to central limit theorem type of results.

To summarize, we have shown in the special case of Equation 4.3 that the correct scaling function is $g(\alpha) = \sqrt{\alpha}$, and the distribution of the limiting random variable $Y$ is a Gaussian distribution with mean zero and variance $1/\sqrt{2}$. In later sections, we extend this result to more general SA algorithms with weaker noise assumptions.

### 4.1.3 Related Literature

Since proposed in [10], SA has been popular for solving large scale optimization problems [43, 23]. Although in principle it requires using diminishing stepsizes to achieve asymptotic convergence, constant stepsize is preferred in practice [61]. Although there are many existing papers studying SA algorithms with both constant and diminishing stepsizes [62, 63, 64, 65, 66, 67, 68, 12], the focus of this chapter is fundamentally different from them. In particular, we are interested in the stationary distribution of the centered scaled iterate (scaled by some function of the constant stepsize), while most of the existing papers study the convergence or convergence rate of the original iterates, and do not study the stationary distribution.

**Constant Stepsize SGD.** In contrast to the success in machine learning practice, there is little discussion about the stationary distribution of constant stepsize SGD. Among existing literature [57, 58, 55], [57] introduced Markov chain theory in the study of constant stepsize SGD algorithm under the strong convexity assumption. They utilized the property that the sequence of iterates is an homogeneous Markov chain to provide an explicit asymptotic expansion of the moments of the averaged SGD iterates. [55] generalized the results in [57] to the setting where the objective function is neither strongly convex nor smooth but satisfies a dissipativity assumption. Under the dissipativity assumption, the authors of [57] established an asymptotic normality result for the constant stepsize SGD algorithm. [58] studied the asymptotic behavior of constant stepsize SGD with a nonconvex, nonsmooth, but locally Lipchitz objective function. It was shown that in a small stepsize regime, the interpolated trajectory of the algorithm converges in probability towards the solutions of the differential inclusion $\dot{x} = \partial F(x)$ and the invariant distribution of the corresponding

Markov chain converges weakly to the set of invariant distributions of the differential inclusion.

The work mentioned before establish the existence of the stationary distribution for constant stepsize SGD type of algorithms. However, it is in general not possible to fully characterize such stationary distribution unless the update rule and the noise sequence are extremely simple (see Subsection 4.1.2). Therefore, we propose studying the limit of such stationary distribution as the constant stepsize goes to zero. Since the SGD iterates will converge to a singleton as the constant stepsize goes to zero, without proper scaling, it is not possible to provide meaningful results regarding the distribution of iterates. Hence none of the previously mentioned work can be applied to study the limiting behavior of SA algorithms in our setting. A concurrent work [59] studied the constant stepsize SA algorithms on a Riemannian manifold and established the limiting distribution of the $\sqrt{\alpha}$ scaled iterate (cf. [59, Theorem 7]). However, for general SA algorithms, the scaling function need not be $g(\alpha) = \sqrt{\alpha}$ and the corresponding distribution need not be Gaussian. In this case, a proper method to determine the correct scaling function is of vital importance.

**Diminishing Stepsize SA.** A set of closely related literature is on studying the asymptotic normality of SA algorithms using diminishing stepsizes, in particular $O(1/k)$ stepsize [69, 70, 71, 72]. Note that in order to have a meaningful distribution, $O(\sqrt{k})$ scaling of the iterates is also required. This type of results can be viewed as an extension of the central limit theorem (which considers only the average of random variables) to the more general SA setting. In contrast, we study constant stepsize SA, which is more preferable in practice due to its fast convergence. In addition, since we have a two-step weak convergence (see Equation 4.2), the analysis is fundamentally different. Moreover, for general SA algorithms, we are the first to demonstrate that the scaling function need not be $\sqrt{\alpha}$, and the limiting stationary distribution need not be Gaussian, see Section 4.3.

**Diffusion Approximation of SGD.** Another set of related literature is on the diffusion approximation of SGD [73, 74, 75, 76, 77], where the authors aim to approximate the trajectory of SGD by a diffusion process which solves the corresponding SDE. Notice that they also study the scaled version of the diffusion limit of SGD. However, different from our approach, their scale is in temporal domain and cannot be applied to our research.

**Heavy Traffic Analysis in Stochastic Networks.** The Markov chain perspective of studying SGD iterates when the constant stepsize goes to zero [57] is qualitatively related to the heavy traffic analysis for stochastic networks [78]. It has been studied in the literature using fluid and diffusion limits [79, 80, 81, 82, 83, 84], where the interchange of limit is usually problematic [78]. An alternative approach in studying stochastic networks is based on a Lyapunov drift argument introduced by [78] and further generalized by [85, 86, 87, 88, 89]. We adopt similar techniques in quantifying the limiting distribution of the scaled SGD iterates. Notice that in stochastic networks, people mainly focus on finite (or countable) state-space Markov chains. However, when it comes to the SA iterates, the state-space is continuous and thus more challenging.

## 4.2 Characterizing the Asymptotic Stationary Distribution

Through out this section, we make the following assumption regarding the noise $\{w_k\}$.

**Assumption 4.2.1.** The noise sequence $\{w_k\}$ is independent and identically distributed with mean zero and a positive definite covariance matrix $\Sigma \in \mathbb{R}^{d\times d}$.

Note that Assumption 4.2.1 is much weaker than the assumption used in Subsection 4.1.2, where the noise is assumed to obey the standard normal distribution. That being said, extending our results to the more general noise setting (e.g. martingale difference noise, and Markovian noise, etc) is one of our future directions.

### 4.2.1 SGD for Minimizing a Smooth and Strongly Convex Objective

Suppose that $F(x) = -\nabla f(x)$, where $f(\cdot)$ is an objective function. Then Equation 4.1 becomes

$$X_{k+1}^{(\alpha)} = X_k^{(\alpha)} + \alpha\left(-\nabla f(X_k^{(\alpha)}) + w_k\right), \tag{4.4}$$

which is the well-known SGD algorithm for minimizing $f(\cdot)$.

To characterize the asymptotic behavior of Equation 4.4, we make the following assumption.

**Assumption 4.2.2.** The objective function $f : \mathbb{R}^d \mapsto \mathbb{R}$ is twice differentiable, and is both $L$ – smooth and $\sigma$ – strongly convex.

Assumption 4.2.2 implies that

$$\begin{aligned} & f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|x-y\|_2^2, && (L\text{ – smooth}) \\ \text{and}\quad & f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\sigma}{2}\|x-y\|_2^2 && (\sigma\text{-convex}) \end{aligned}$$

for all $x, y \in \mathbb{R}^d$. In addition, the function $f(x)$ has a unique minimizer (or $F(x) = 0$ has a unique solution), which we have denoted by $x^*$. To proceed, let $Y_k^{(\alpha)} = (X_k^{(\alpha)} - x^*)/\sqrt{\alpha}$ be the centered scaled iterate. We first derive the corresponding update equation of $Y_k^{(\alpha)}$ in following:

$$Y_{k+1}^{(\alpha)} = Y_k^{(\alpha)} - \sqrt{\alpha}\nabla f\left(\sqrt{\alpha}Y_k^{(\alpha)} + x^*\right) + \sqrt{\alpha}w_k, \tag{4.5}$$

which is obtained by subtracting $x^*$ from both sides of Equation 4.4 and then dividing by $\sqrt{\alpha}$.

We next characterize the two-step weak convergence (cf. Equation 4.2) of $\{Y_k^{(\alpha)}\}$ in the following theorem. Let $H_f \in \mathbb{R}^{d\times d}$ be the Hessian matrix of the objective function $f(\cdot)$

evaluated at $x^*$, which is well-defined because $f(\cdot)$ is twice differentiable.

**Theorem 4.2.1.** *Consider the iterates $\{Y_k^{(\alpha)}\}$ generated by Equation 4.5. Suppose that Assumptions 4.2.1 and 4.2.2 are satisfied, then the following statements hold.*

*(1) There exists a threshold $\bar{\alpha} > 0$ such that for all $\alpha \in (0, \bar{\alpha})$, the sequence of random variables $\{Y_k^{(\alpha)}\}$ converges weakly to some random variable $Y^{(\alpha)}$, which satisfies $\mathbb{E}[\|Y^{(\alpha)}\|_2^2] < \infty$.*

*(2) For any positive sequence $\{\alpha_k\}$ satisfying $\alpha_k \in (0, \bar{\alpha})$ for all $k \geq 0$ and $\lim_{k\to\infty} \alpha_k = 0$, the sequence $\{Y^{(\alpha_k)}\}$ converges weakly to a random variable $Y$, which satisfies the following equation*

$$\mathbb{E}\left[\left(t^\top \Sigma t + 2it^\top H_f Y\right) e^{it^\top Y}\right] = 0, \ \forall\, t \in \mathbb{R}^d, \tag{4.6}$$

*where $i$ is the imaginary unit. In addition, suppose that Equation 4.6 has a unique solution (in terms of the distribution of $Y$), then the distribution of $Y$ is the multivariate normal distribution with mean zero and covariance matrix $\Sigma_Y$ being the unique solution of the Lyapunov equation*

$$H_f \Sigma_Y + \Sigma_Y H_f^\top = \Sigma. \tag{4.7}$$

*Remark.* To establish Theorem 4.2.1 (2), we require Equation 4.6 to have a unique solution in terms of the distribution of $Y$. Such uniqueness assumption will be discussed and relaxed to some extent in Subsection 4.2.4.

Since $\Sigma$ is positive definite, and $H_f$ is also positive definite under strong convexity, it is well established in the literature that the Lyapunov equation $H_f \Sigma_Y + \Sigma_Y H_f^\top = \Sigma$ has a unique solution [90]. One way of writing the solution $\Sigma_Y$ is given by

$$\Sigma_Y = \int_0^\infty e^{-H_f u} \Sigma e^{-H_f^\top u} du.$$

See [91] for an alternative approach of solving Lyapunov equations using Kronecker product.

To better understand Theorem 4.2.1, consider the scalar setting where $f(x) = x^2/2$ and $\Sigma = 1$. In this case we have $H_f = 1$ and hence $\Sigma_Y = 1/2$ by the Lyapunov equation (cf. Equation 4.7). As a result, the distribution of the limiting random variable $Y$ is a Gaussian distribution with mean zero and variance $1/2$. This agrees with the illustrative example presented in Subsection 4.1.2.

From Theorem 4.2.1, we see that the distribution of $Y$ only depends on the Hessian of $f(\cdot)$ at $x^*$. This makes intuitive sense because we are studying the asymptotic behavior of Equation 4.4, and only the properties of $f(\cdot)$ around $x^*$ should play a role in characterizing the stationary distribution.

### 4.2.2 Linear Stochastic Approximation

Suppose that $F(x) = Ax + b$, where $A \in \mathbb{R}^{d\times d}$ and $b \in \mathbb{R}^d$. Then Equation 4.1 becomes

$$X_{k+1}^{(\alpha)} = X_k^{(\alpha)} + \alpha \left( AX_k^{(\alpha)} + b + w_k \right), \tag{4.8}$$

which aims at iteratively solving the linear system of equations $Ax + b = 0$. Note that since the matrix $A$ is not necessarily symmetric, $F(x) = Ax + b$ need not be the gradient of any objective function. Such linear SA algorithms arise in many realistic applications. One typical example is TD-learning (with linear function approximation) for solving the policy evaluation problem in RL, where the goal is to solve a linear Bellman equation. See [11, 92, 12, 40] for more details about TD-learning as a linear SA algorithm.

To study the asymptotic behavior of Equation 4.8, we make the following assumption regarding the matrix $A$.

**Assumption 4.2.3.** The matrix $A$ is Hurwitz., i.e., all eigenvalues of $A$ have strict negative real parts.

*Remark.* Since $A$ being Hurwitz implies $A$ being non-singular, Assumption 4.2.3 ensures that the target equation $Ax + b = 0$ has a unique solution, which we have denoted by $x^*$.

Assumption 4.2.3 is standard in studying linear SA algorithms. In particular, it was shown in the literature that under Assumption 4.2.3 and some mild conditions on the noise $\{w_k\}$, Equation 4.8 converges in the mean square sense to a neighborhood around $x^*$ [12].

To study the asymptotic distribution, for a fixed stepsize $\alpha$, we define the centered scaled iterate $Y_k^{(\alpha)}$ by $Y_k^{(\alpha)} = (X_k^{(\alpha)} - x^*)/\sqrt{\alpha}$ for all $k \geq 0$. To find the corresponding update equation for $Y_k^{(\alpha)}$, we first subtract $x^*$ from both sides of Equation 4.8 to obtain

$$X_{k+1}^{(\alpha)} - x^* = X_k^{(\alpha)} - x^* + \alpha \left( A(X_k^{(\alpha)} - x^*) + w_k \right),$$

where we used $Ax^* + b = 0$. Then we divide both sides of the previous inequality by $\sqrt{\alpha}$ to obtain:

$$Y_{k+1}^{(\alpha)} = (I + \alpha A) Y_k^{(\alpha)} + \sqrt{\alpha} w_k. \tag{4.9}$$

The full characterization of the two-step weak convergence (cf. Equation 4.2) of the random process $\{Y_k^{(\alpha)}\}$ is captured by the following theorem.

**Theorem 4.2.2.** *Consider the iterates $\{Y_k^{(\alpha)}\}$ generated by Equation 4.9. Suppose that Assumptions 4.2.1 and 4.2.3 are satisfied, then the following statements hold.*

*(1) There exists a threshold $\bar{\alpha}' > 0$ such that for all $\alpha \in (0, \bar{\alpha}')$, the sequence of random variables $\{Y_k^{(\alpha)}\}$ converges weakly to some random variable $Y^{(\alpha)}$, which satisfies $\mathbb{E}[\|Y^{(\alpha)}\|_2^2] < \infty$.*

*(2) For any positive sequence $\{\alpha_k\}$ satisfying $\alpha_k \in (0, \bar{\alpha}')$ for all $k \geq 0$ and $\lim_{k\to\infty} \alpha_k = 0$, the sequence of random variables $\{Y^{(\alpha_k)}\}$ converges weakly to a random variable*

*$Y$, which satisfies the following equation*

$$\mathbb{E}\left[\left(t^\top \Sigma t - 2it^\top AY\right) e^{it^\top Y}\right] = 0, \quad \forall\, t \in \mathbb{R}^d. \tag{4.10}$$

*In addition, suppose that Equation 4.10 has a unique solution in terms of the distribution of $Y$, then $Y$ obeys the multivariate normal distribution with mean zero and covariance matrix being the unique solution $\Sigma_Y$ of the Lyapunov equation.*

$$A\Sigma_Y + \Sigma_Y A^\top + \Sigma = 0. \tag{4.11}$$

Since the matrix $A$ is Hurwitz, and the matrix $\Sigma$ is positive definite, the existence and uniqueness of a positive definition solution to the Lyapunov equation (cf. Equation 4.11) are guaranteed [91].

Lyapunov equations were used extensively in the stability analysis of ODEs. For example, the ODE associated with Equation 4.8 is given by $\dot{x}(t) = Ax(t) + b$ [33], and the function $W(x) = (x - x^*)^\top \Sigma_Y (x - x^*)$ is a valid Lyapunov function for showing the global geometric stability of this ODE [91]. Interestingly, according to Theorem 4.2.2, the solution $\Sigma_Y$ also plays an important role in characterizing the limit distribution of centered scaled iterates of linear SA algorithm (cf. Equation 4.8), which can viewed as a discrete and stochastic counterpart of the ODE $\dot{x}(t) = Ax(t) + b$.

### 4.2.3 Stochastic Approximation under Contraction Assumption

Suppose that $F(x) = \mathcal{T}(x) - x$, where $\mathcal{T} : \mathbb{R}^d \times \mathbb{R}^d$ is a general nonlinear operator. In this case, Equation 4.1 becomes

$$X_{k+1}^{(\alpha)} = X_k^{(\alpha)} + \alpha\left(\mathcal{T}\left(X_k^{(\alpha)}\right) - X_k^{(\alpha)} + w_k\right), \tag{4.12}$$

which can be interpreted as an SA algorithm for finding the fixed-point of the operator $\mathcal{T}(\cdot)$. These type of algorithms arise in the context of RL. Specifically, many popular RL algorithms such as $Q$-learning [17] and TD-learning [56] are SA algorithms for solving fixed-point equations (i.e., Bellman equations), where the fixed-point operators (i.e., Bellman operators) are contraction mappings. Therefore, our result is closely related to those RL algorithms. In fact, one of our immediate future directions is to actually extend our study to constant-stepsize RL algorithms and provide theoretical insights about their stationary distributions.

To proceed and study Equation 4.12, we need the following definition.

**Definition 4.2.1.** Let $\nu_i$, $1 \leq i \leq d$ be positive real numbers. Then the weighted $\ell_2$-norm $\|\cdot\|_\nu$ with weights $\{\nu_i\}_{1\leq i\leq d}$ is defined by $\|x\|_\nu = (\sum_{i=1}^d \nu_i x_i^2)^{1/2}$ for all $x \in \mathbb{R}^d$.

Next, we state our assumption regarding the operator $\mathcal{T}(\cdot)$.

**Assumption 4.2.4.** The operator $\mathcal{T}(\cdot)$ is continuously differentiable, and there exists $\beta \in (0,1)$ such that $\|\mathcal{T}(x_1) - \mathcal{T}(x_2)\|_\mu \leq \beta\|x_1 - x_2\|_\mu$ for any $x_1, x_2 \in \mathbb{R}^d$, where $\|\cdot\|_\mu$ is some weighted $\ell_2$-norm with weights $\{\mu_i\}_{1\leq i\leq d}$.

Assumption 4.2.4 essentially states that the operator $\mathcal{T}(\cdot)$ is a contraction mapping with respect to the weighted $\ell_2$-norm $\|\cdot\|_\mu$. By Banach fixed-point theorem [47], the operator $\mathcal{T}(\cdot)$ has a unique fixed-point $x^*$.

To proceed, we derive the update equation of the centered scaled iterate $Y_k^{(\alpha)} = (X_k^{(\alpha)} - x^*)/\sqrt{\alpha}$ in the following:

$$Y_{k+1}^{(\alpha)} = Y_k^{(\alpha)} + \sqrt{\alpha}\left(\mathcal{T}\left(\sqrt{\alpha}Y_k^{(\alpha)} + x^*\right) - \left(\sqrt{\alpha}Y_k^{(\alpha)} + x^*\right)\right) + \sqrt{\alpha}w_k. \tag{4.13}$$

To characterize the distribution of the limiting random vector $Y$ (cf. Equation 4.2), let $J \in \mathbb{R}^{d\times d}$ be the Jacobian matrix of the operator $\mathcal{T}(\cdot)$ evaluated at $x^*$, which is well defined because $\mathcal{T}(\cdot)$ is continuously differentiable. We first show that all eigenvalues of the matrix

$J$ are contained in the open unit ball of the complex plane. This result is important for us to later describe the covariance matrix of the limiting random vector $Y$.

**Lemma 4.2.1.** *The spectral radius $r(J) := \max_{1\leq i\leq d} |\lambda_i(J)|$ of the matrix $J$ is strictly less than* $1$.

*Proof of Lemma 4.2.1.* We first show that $r(J) \leq \|J\|$ for any induced matrix norm $\|\cdot\|$. Let $(\lambda_i, v_i)$ be an eigenvalue-eigenvector pair of the matrix $J$. Then we have $\|Jv_i\| = \|\lambda_i v_i\| = |\lambda_i|\|v_i\|$, which implies

$$\|J\| := \max_{x\neq \mathbf{0}} \frac{\|Jx\|}{\|x\|} \geq \frac{\|Jv_i\|}{\|v_i\|} = |\lambda_i|.$$

Since the previous inequality holds for any eigenvalue $\lambda_i$ of the matrix $J$, we have $r(J) \leq \|J\|$. The rest of the proof follows by showing $\|J\|_\mu \leq \beta$ under the contraction assumption, which can be found on standard analysis textbooks. □

The next theorem characterizes the distribution of the two-step limiting random vector $Y$ of the centered scaled iterates $\{Y_k^{(\alpha)}\}$ of Equation 4.12.

**Theorem 4.2.3.** *Consider the iterates $\{Y_k^{(\alpha)}\}$ generated by Equation 4.13. Suppose that Assumptions 4.2.1 and 4.2.4 are satisfied, then the following statements hold.*

*(1) There exists a threshold $\bar{\alpha}'' > 0$ such that for all $\alpha \in (0, \bar{\alpha}'')$, the sequence of random variables $\{Y_k^{(\alpha)}\}$ converges weakly to some random variable $Y^{(\alpha)}$, which satisfies $\mathbb{E}[\|Y^{(\alpha)}\|_2^2] < \infty$.*

*(2) For any positive sequence $\{\alpha_k\}$ satisfying $\alpha_k \in (0, \bar{\alpha}'')$ for all $k \geq 0$ and $\lim_{k\to\infty} \alpha_k = 0$, the sequence of random variables $\{Y^{(\alpha_k)}\}$ converges weakly to a random variable $Y$, which satisfies the following equation*

$$\mathbb{E}\left[\left(t^\top \Sigma t - 2it^\top (J - I)Y\right) e^{it^\top Y}\right] = 0, \quad \forall\, t \in \mathbb{R}^d. \tag{4.14}$$

*In addition, suppose that Equation 4.14 has a unique solution, then $Y$ obeys a multivariate normal distribution with mean zero and covariance matrix being the unique solution of the Lyapunov equation $(J-I)\Sigma_Y + \Sigma_Y(J-I)^\top + \Sigma = 0$.*

Under the contraction assumption, the spectral radius of the Jacobian matrix $J$ is strictly less than one (cf. Lemma 4.2.1). Therefore, all eigenvalues of the matrix $J-I$ belong to the open-left half of the complex plane. As a result, the matrix $J-I$ is Hurwitz and hence the Lyapunov equation $(J-I)\Sigma_Y + \Sigma_Y(J-I)^\top + \Sigma = 0$ has a unique positive definite solution $\Sigma_Y$ [91].

### 4.2.4 The Uniqueness Assumption

In Theorem 4.2.1, Theorem 4.2.2, and Theorem 4.2.3, after obtaining the implicit equations (i.e., Equation 4.6, Equation 4.10, and Equation 4.14), to conclude that the distribution of $Y$ is Gaussian, we need to assume that the equation has a unique solution. In this subsection, we show that such uniqueness assumption can be relaxed to some extend.

*Uni-Dimensional Setting*

Suppose that we are in the uni-dimensional setting, i.e., $d = 1$. Then Equation 4.6, Equation 4.10, and Equation 4.14 all reduce to an equation of the following form: $\mathbb{E}[(at + 2biY)e^{itY}] = 0$ for all $t \in \mathbb{R}$, where $a$ and $b$ are positive constants. Let $\phi_Y(t) = \mathbb{E}[e^{itY}]$ be the characteristic function of the random variable $Y$. Then we can rewrite the previous equation as

$$at\phi_Y(t) + 2b\frac{d\phi_Y(t)}{dt} = 0, \tag{4.15}$$

where the interchange of integral and differentiation is justified [93]. Now Equation 4.15 is an ODE, which has solutions of the form

$$\phi_Y(t) = C \exp\left(-\frac{a}{4b}t^2\right),$$

where $C$ is a constant. Since $\phi_Y(t)$ as a characteristic function, hence satisfies $\phi_Y(0) = 1$, we have $C = 1$. It follows that $\phi_Y(t) = \exp(-\frac{a}{4b}t^2)$, which is the characteristic function for a Gaussian random variable with mean zero and covariance $\sqrt{a/(2b)}$.

Based on the previous analysis, the uniqueness assumption about Equation 4.6, Equation 4.10, and Equation 4.14 can be removed in the uni-dimensional setting.

*Multi-Dimensional Setting*

Moving to the multi-dimensional setting, consider Equation 4.6 of Theorem 4.2.1 as a representative example. To reproduce Theorem 4.2.1 (2) without imposing the uniqueness assumption, we consider the setting where (1) the Hessian matrix $H_f$ of the objective function $f(\cdot)$ evaluated at $x^*$ is the identity matrix, and (2) the covariance matrix of the noise $w_k$ is also an identity matrix. Extending the result to the more general setting where $H_f$ and $\Sigma$ can be any positive definite matrices is a future research direction.

Similarly let $\phi_Y(t) = \mathbb{E}[e^{it^\top Y}]$ be the characteristic function of the random vector $Y$. Then in this case Equation 4.6 becomes $t^\top t\phi_Y(t) + 2t^\top \nabla\phi_Y(t) = 0$, which is equivalently to

$$\begin{aligned} 0 &= t^\top t + 2t^\top \frac{\nabla\phi_Y(t)}{\phi_Y(t)} \\ &= t^\top t + 2t^\top \nabla\psi_Y(t), \end{aligned} \tag{4.16}$$

where $\psi_Y(t) := \log(\phi_Y(t))$. To solve the partial differential equation (PDE) (i.e., Equation 4.24), we will first convert the PDE from Cartesian coordinates to spherical coordinates, which then becomes directly solvable.

The $d$-dimensional spherical coordinate system consists of a radial coordinate $\rho$, and $d-1$ angular coordinates $\{\theta_i\}_{1\leq i\leq d-1}$. The relation between the Cartesian coordinates $(t_1,\cdots,t_d)$ and the spherical coordinates $(\rho,\theta_1,\cdots,\theta_{d-1})$ is given by

$$\begin{aligned}
t_1 &= \rho sin(\theta_1)sin(\theta_2)\cdots sin(\theta_{d-2})sin(\theta_{d-1}),\\
t_2 &= \rho cos(\theta_1)sin(\theta_2)\cdots sin(\theta_{d-2})sin(\theta_{d-1}),\\
t_3 &= \rho cos(\theta_2)sin(\theta_3)\cdots sin(\theta_{d-2})sin(\theta_{d-1}),\\
&\vdots\\
t_{d-1} &= \rho cos(\theta_{d-2})sin(\theta_{d-1}),\\
t_d &= \rho cos(\theta_{d-1}),
\end{aligned}$$

where $\theta_1 \in [0,2\pi)$ and $\theta_i \in [0,\pi]$ for all $i = 2,3,\cdots,d-1$. To proceed, we first compute the Jacobian matrix $J_d$ of the transformation based on the formula presented in [94]. Specifically, we have

$$\begin{aligned}
J_d &= \frac{\partial(t_1,t_2,\cdots,t_d)}{\partial(\rho,\theta_1,\theta_2,\cdots,\theta_{d-1})}\\
&= \begin{bmatrix}
\frac{t_1}{\rho} & t_1cot(\theta_1) & t_1cot(\theta_2) & \cdots & t_1cot(\theta_{d-2}) & t_1cot(\theta_{d-1})\\
\frac{t_2}{\rho} & -t_2tan(\theta_1) & t_2cot(\theta_2) & \cdots & t_2cot(\theta_{d-2}) & t_2cot(\theta_{d-1})\\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\
\frac{t_{d-1}}{\rho} & 0 & 0 & \cdots & -t_{d-1}tan(\theta_{d-2}) & t_{d-1}cot(\theta_{d-1})\\
\frac{t_d}{\rho} & 0 & 0 & \cdots & 0 & -t_dtan(\theta_{d-1})
\end{bmatrix}.
\end{aligned}$$

Using the spherical coordinate system, Equation 4.24 can written as

$$\rho^2 + 2t^\top J_d^{-1}\nabla\psi_Y(\rho,\theta_1,\cdots,\theta_{d-1}).$$

which by direct computation simplifies to

$$\rho + 2\frac{\partial \psi_Y(\rho, \theta_1, \cdots, \theta_{d-1})}{\partial \rho} = 0.$$

This implies that $\psi_Y(\rho, \theta_1, \cdots, \theta_{d-1}) = -\frac{\rho^2}{4} + C(\theta_1, \cdots, \theta_{d-1})$. Using the initial condition that $\psi_Y(0, \theta_1, \cdots, \theta_{d-1}) = \log(\phi_Y(0)) = \log(1) = 0$ for any $\theta_1, \cdots, \theta_{d-1}$, we see that $C(\theta_1, \cdots, \theta_{d-1}) = 0$ and hence $\phi_Y(\rho, \theta_1, \cdots, \theta_{d-1}) = \frac{\rho^2}{4}$. Therefore, we have that $\psi_Y(t) = -\frac{t^\top t}{4}$, which implies $\phi_Y(t) = \exp(-\frac{t^\top t}{4})$. It follows that the distribution of $Y$ is the multinormal distribution with mean zero and covariance matrix being $I_d/\sqrt{2}$. This agrees with Theorem 4.2.1 (2) when $H_f = \Sigma = I$, but the uniqueness assumption is not required to establish the result.

## 4.3 Identifying the Suitable Scaling Function for More General Stochastic Approximation Algorithms

In the previous section, we have shown that for several particular SA algorithms (e.g. SGD, linear SA, and contractive SA), the scaling function is $g(\alpha) = \sqrt{\alpha}$ and distribution of the limiting random variable $Y$ is a Gaussian distribution. In this section, we consider more general SA algorithms. We first show impirically in the following subsection that in general the scaling function need not be $g(\alpha) = \sqrt{\alpha}$, and the distribution of $Y$ need not be Gaussian.

### 4.3.1 Numerical Experiments

Suppose that Equation 4.1 is the SGD algorithm for minimizing the scalar objective $f(x) = x^4/4$. That is:

$$X_{k+1}^{(\alpha)} = X_k^{(\alpha)} + \alpha\left(-(X_k^{(\alpha)})^3 + w_k\right). \tag{4.17}$$

Note that $f(\cdot)$ in this case is neither smooth nor strongly convex. It is clear that the unique minimizer of $f(\cdot)$ is zero. Let the centered scaled iterate $Y_k^{(\alpha)}$ be defined by $Y_k^{(\alpha)} = X_k^{(\alpha)}/g(\alpha)$. We next use numerical simulation to show that the correct scaling function in this case should be $g(\alpha) = \alpha^{1/4}$ instead of $g(\alpha) = \sqrt{\alpha}$.

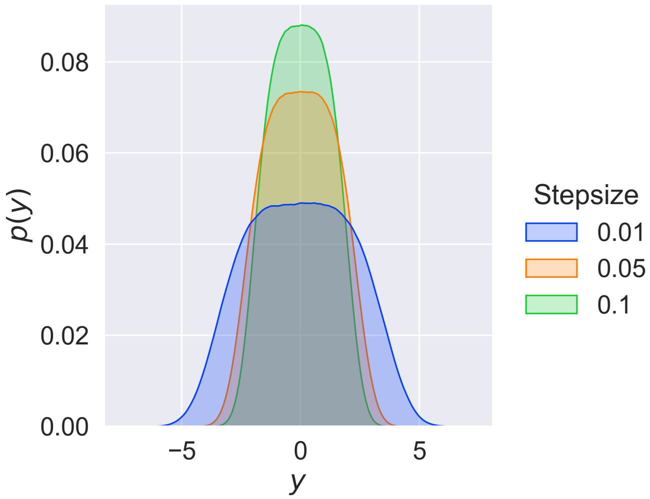

Figure 4.1: Estimated Density Functions When Choosing $g(\alpha) = \alpha^{1/2}$

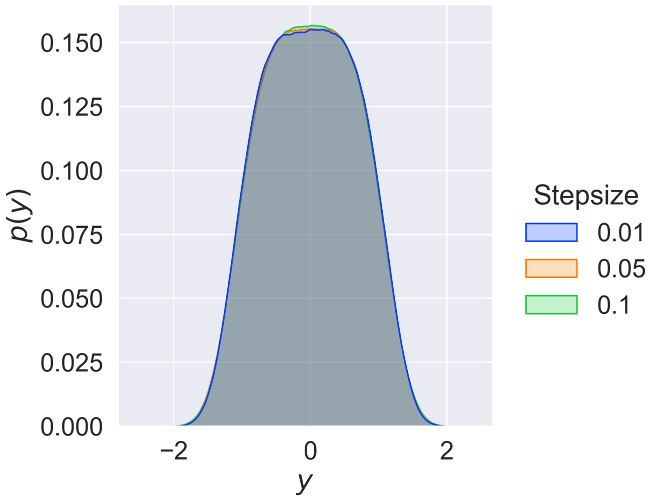

Figure 4.2: Estimated Density Functions When Choosing $g(\alpha) = \alpha^{1/4}$

In Figure 11.1 and Figure 11.2, we plot the empirical density function of $Y^{(\alpha)}$ for different $\alpha$. For the right scaling function, we expect the density function to converge

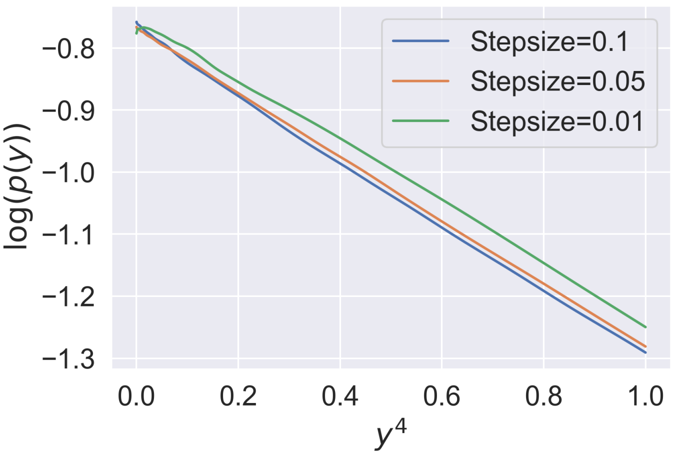

Figure 4.3: $\log(p_Y(y))$ as a Function of $y^4$

as $\alpha$ decreases, while for the wrong scaling function, we expect the density function to change drastically for order-wise different $\alpha$. As we see, it is clear that $g(\alpha) = \sqrt{\alpha}$ is not suitable in this case, and $g(\alpha) = \alpha^{1/4}$ seems to be the right scaling.

To further verify this result, we plot the logarithmic empirical density function as a function of $y^4$ in Figure 11.3. We observe linear growth in Figure 11.3. This indicates that the density function $p_Y(y)$ is proportional to $e^{\beta y^4}$, where $\beta$ is some numerical constant. Therefore, numerical experiments suggest that the distribution of $Y$ is not Gaussian but super Gaussian in this problem.

### 4.3.2 A Method to Determine the Suitable Scaling Function

Inspired by the numerical simulations provided in the previous section, we here provide a method to determine the correct scaling function for general SA algorithms.

To gain intuition, we consider the centered scaled iterates $Y_k^{(\alpha)} = X_k^{(\alpha)}/\alpha^{1/4}$ for Equation 4.17. The update equation of $Y_k^{(\alpha)}$ is given by

$$Y_{k+1}^{(\alpha)} = Y_k^{(\alpha)} - \alpha^{3/2}(Y_k^{(\alpha)})^3 + \alpha^{3/4} w_k.$$

Notably, the factor in terms of the stepsize $\alpha$ in front of the term $(Y_k^{(\alpha)})^3$ is $\alpha^{3/2}$, which is equal to the square of the factor $\alpha^{3/4}$ in front of the noise term $w_k$.

Now for the general SA algorithm presented in Equation 4.1, by rewriting Equation 4.1 in terms of the centered scaled iterate $Y_k^{(\alpha)} = (X_k^{(\alpha)} - x^*)/g(\alpha)$, we have

$$Y_{k+1}^{(\alpha)} = Y_k^{(\alpha)} + \left(\frac{\alpha}{g(\alpha)}\right)^2 \frac{g(\alpha)F(Y_k^{(\alpha)}g(\alpha) + x^*)}{\alpha} + \frac{\alpha}{g(\alpha)} w_k. \tag{4.18}$$

In view of the previous equation and the empirical observations in the previous section, we see that we need to choose a scaling function $g(\alpha)$ such that the following condition is satisfied.

**Condition 4.3.1.** The scaling function $g(\cdot)$ should be chosen such that

(1) $\lim_{\alpha\to 0} \frac{\alpha}{g(\alpha)} = 0$ and $\lim_{\alpha\to 0} g(\alpha) = 0$

(2) The function $\tilde{F} : \mathbb{R}^d \mapsto \mathbb{R}^d$ defined by $\tilde{F}(y) = \lim_{\alpha\to 0} \frac{g(\alpha)F(yg(\alpha)+x^*)}{\alpha}$ is a nontrivial function in the sense that $\tilde{F}(\cdot)$ is not identically equal to zero or infinity.

We next verify the choice of scaling functions in Section Section 4.2 using our proposed Condition 4.3.1. For SGD with a smooth and strong convex objective, since

$$\sigma\|x - x^*\|_2 \leq \|\nabla f(x) - \nabla f(x^*)\|_2 = \|\nabla f(x)\|_2 \leq L\|x - x^*\|_2, \quad \forall\, x \in \mathbb{R}^d,$$

we have

$$\sigma\frac{g(\alpha)^2}{\alpha}\|y\|_2 \leq \left\|\frac{g(\alpha)\nabla f(g(\alpha)y + x^*)}{\alpha}\right\|_2 \leq L\frac{g(\alpha)^2}{\alpha}\|y\|_2.$$

In view of the previous inequality and Condition 4.3.1, it is clear that the only possible choice of $g(\alpha)$ is $g(\alpha) = \sqrt{\alpha}$.

For linear SA algorithms studied in Subsection 4.2.2, since

$$\frac{g(\alpha)[A(g(\alpha)y + x^*) + b]}{\alpha} = \frac{g(\alpha)^2}{\alpha}Ay,$$

to satisfy Condition 4.3.1, we need to choose $g(\alpha) = \sqrt{\alpha}$.

As for contractive SA algorithms studied in Subsection 4.2.3, using the contraction property and we have

$$(1-\gamma)\|x-x^*\|_\mu \leq \|\mathcal{T}(x)-x\|_\mu = \|\mathcal{T}(x)-\mathcal{T}(x^*)-(x-x^*)\|_\mu \leq (1+\gamma)\|x-x^*\|_\mu.$$

It follows that

$$\frac{g(\alpha)^2}{\alpha}(1-\gamma)\|y\|_\mu \leq \left\|\frac{g(\alpha)[\mathcal{T}(g(\alpha)y+x^*)-(g(\alpha)y+x^*)]}{\alpha}\right\|_\mu \leq \frac{g(\alpha)^2}{\alpha}(1+\gamma)\|y\|_\mu.$$

Since all norms are "equivalent" in finite dimensional space, the previous inequality implies that we must choose $g(\alpha) = \sqrt{\alpha}$.

To further verify the correctness of the scaling function suggested by Condition 4.3.1, consider the SGD algorithm

$$X_{k+1}^{(\alpha)} = X_k^{(\alpha)} + \alpha(-\nabla f(X_k^{(\alpha)}) + w_k)$$

with the following two choices of objective functions: (1) $f(x) = e^{x^2}$, and (2) $f(x) = \frac{x^4}{4} + \frac{\sin^2(x)}{2}$. Note that in these two cases the function $f(\cdot)$ is not smooth and strongly convex.

**Case 1.** In the first case where $f(x) = e^{x^2}$, since

$$\left\|\frac{g(\alpha)F(yg(\alpha))}{\alpha}\right\|_2 = \frac{g(\alpha)^2}{\alpha}2|y|e^{(yg(\alpha))^2},$$

when choosing $g(\alpha) = \sqrt{\alpha}$, we have $\tilde{F}(y) = \lim_{\alpha\to 0}\frac{g(\alpha)^2}{\alpha}2ye^{(yg(\alpha))^2} = 2y$. This suggests that the distribution of the limiting random variable $Y$ has a density function proportional to $e^{\beta' x^2}$, where $\beta'$ is a numerical constant.

One interesting insight of this example is the following. Observe that we have $\frac{de^{x^2}}{dx} =$

$\sum_{k=1}^{\infty}(2k)x^{2k-1}$ by Taylor series. The function $\tilde{F}(\cdot)$ in this example is exactly the dominant term that appears in the Taylor series.

We next verify this choice of $g(\alpha)$ and the distribution of $Y^{(\alpha)}$ for small enough $\alpha$ using numerical simulation in the following.

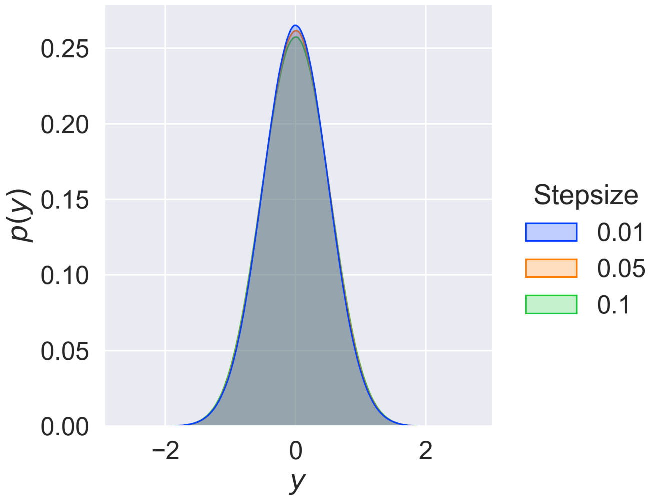

Figure 4.4: Estimated Density Functions When Choosing $g(\alpha) = \alpha^{1/2}$

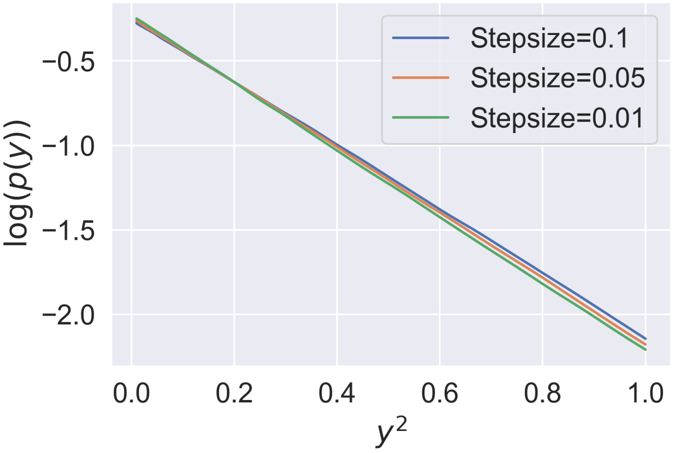

Figure 4.5: $\log(p_Y(y))$ as a Function of $y^2$

We see from Figure 11.4 that with the scaling function $g(\alpha) = \sqrt{\alpha}$, the empirical density function of the random variable $Y^{(\alpha)}$ seems to converge. Figure 4.5 further justifies this result by showing that the density function $p_Y(y)$ of the distribution of $Y$ in this case is proportional to $e^{\beta' x^2}$, where $\beta'$ is a numerical constant.

**Case 2.** Consider case where $f(x) = \frac{x^4}{4} + \frac{\sin^2(x)}{2}$. Observe that

$$\left\|\frac{g(\alpha)F(yg(\alpha))}{\alpha}\right\|_2 = \frac{g(\alpha)}{\alpha}|y^3 g(\alpha)^3 + sin(yg(\alpha))cos(yg(\alpha))|.$$

Since $\lim_{x\to 0}\frac{sin(x)}{x} = 1$, the only possible choice of the scaling function $g(\alpha)$ to satisfy Condition 4.3.1 (2) is $g(\alpha) = \sqrt{\alpha}$. In this case, we have $\tilde{F}(y) = \lim_{\alpha\to 0}\frac{1}{\sqrt{\alpha}}y^3\alpha^{3/2} + sin(y\sqrt{\alpha})cos(y\sqrt{\alpha}) = y$ by L'Hôpital's rule. Since $x^4$ is dominated by $sin^2(x)$ as $x$ approaches $x^*$ (which is $0$), the scaling function and the function $\tilde{F}(\cdot)$ are determined only by the dominant term.

Similarly, we verify this choice of scaling function via numerical experiments. In Figure 4.6 and Figure 4.7, we plot the empirical density function of the random variable $Y^{(\alpha)}$ for different stepsize $\alpha$, and see if the density function converges as $\alpha$ goes to zero. The results suggest that $g(\alpha) = \alpha^{1/2}$ seems to be the correct scaling. To further verify this result, we plot the logarithmic function of the empirical density of $Y^{(\alpha)}$ as a function of $y^2$ and observe straight lines. Therefore, the distribution of $Y^{(\alpha)}$ is proportional to $e^{\beta'' x^2}$, where $\beta''$ is a numerical constants.

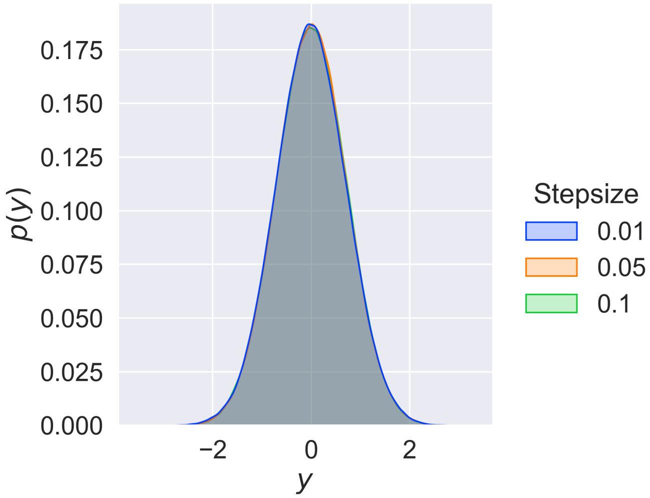

Figure 4.6: Estimated Density Functions When Choosing $g(\alpha) = \alpha^{1/2}$

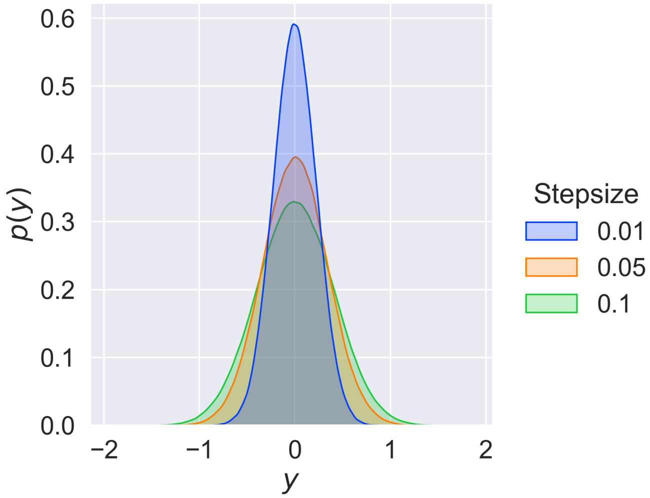

Figure 4.7: Estimated Density Functions When Choosing $g(\alpha) = \alpha^{1/4}$

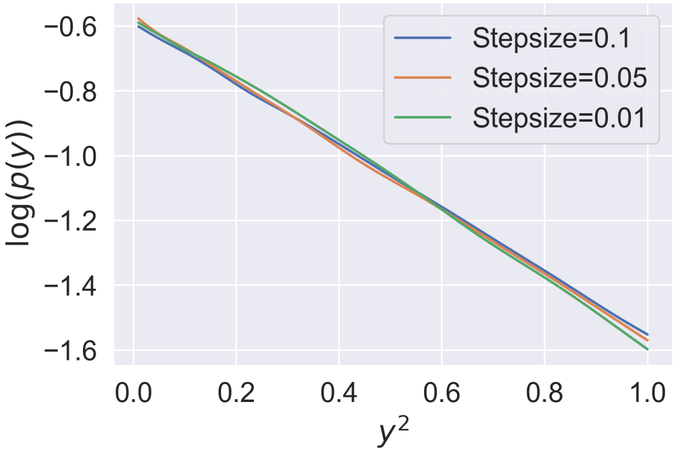

Figure 4.8: $\log(p_Y(y))$ as a Function of $y^2$

### 4.3.3 Connection to Euler-Maruyama Discretization Scheme for Approximating SDE

The choice of the scaling function suggested by Condition 4.3.1 has an insightful connection to the Euler-Maruyama discretization scheme for approximate the solution of an SDE, as elaborated below. Let $(B_t)_{t\geq 0}$ be a Brownian motion. Consider the following SDE:

$$dX_t = F(X_t)dt + dB_t \tag{4.19}$$

with initial condition $X_0$. The Euler-Maruyama discretization $\{\hat{X}_k\}$ to the solution $(X_t)$ of SDE (cf. Equation 4.19) is defined as follows. Let $\Delta t$ be the discretization accuracy. Set $\hat{X}_0 = X_0$, and recursively define $\hat{X}_k$ according to

$$\hat{X}_{k+1} = \hat{X}_k + \Delta t F(\hat{X}_k) + (B_{(k+1)\Delta t} - B_{k\Delta t}).$$

Since $(B_t)_{t\geq 0}$ is a Brownian motion, we have $(B_{(k+1)\Delta t} - B_{k\Delta t}) \sim \mathcal{N}(0, \Delta t)$. Therefore, by letting $\{Z_k\}$ be an i.i.d. sequence of standard normal random variables, we can rewrite the previous equation as

$$\hat{X}_{k+1} = \hat{X}_k + \Delta t F(\hat{X}_k) + \sqrt{\Delta t} Z_k. \tag{4.20}$$

The approximation property of the Euler-Maruyama discretization to its corresponding SDE has been studied in the literature, see [95]. Specifically, it was shown that under some mild conditions on $F(\cdot)$, the Euler-Maruyama scheme is known to have the first-order accuracy of the SDE. As a consequence, intuitively, when $(X_t)_{t\geq 0}$ has a stationary distribution $\mu$, the limiting distribution $\mu_{\Delta t}$ of $\{\hat{X}_k\}$ as a function of the discretization accuracy $\Delta t$ should converge weakly to $\mu$ as $\Delta t$ tends to zero. If we view the discretization accuracy $\Delta t$ as the stepsize in Equation 4.20. In order for $\mu_{\Delta t}$ to converge to some nontrivial distribution $\mu$ as $\Delta t$ tends to zero, it is important to notice that the scaling factor of the noise $Z_k$ in terms of $\Delta t$ must be order-wise equal to the square root of the scaling factor of $F(\hat{X}_k)$. This observation coincides with Equation 4.18 in the previous section, which eventually leads to our Condition 4.3.1.

## 4.4 Proof of All Theoretical Results

In this section, we present the proofs of Theorem 4.2.1, Theorem 4.2.2 and Theorem 4.2.3. We begin with Theorem 4.2.1.

**High Level Idea.** Before going into details, we first highlight the main ideas for the proof. Theorem 4.2.1 (1) follows from existing results in the literature, in particular, [55, Proposition 2.1]. As for Theorem 4.2.1 (2), consider any positive sequence $\{\alpha_k\}$ such that $\lim_{k\to\infty} \alpha_k = 0$. Since the family of random variables $\{Y^{(\alpha_k)}\}$ is tight (which is implied by Theorem 4.2.1 (1)), there is a weakly convergent subsequence $\{Y^{\alpha_{k_\ell}}\}$. We further show that the weak limit $Y$ of the subsequence $\{Y^{(\alpha_{k_\ell})}\}$ solves Equation 4.6. In this case, under the assumption that Eq. (Equation 4.6) has a unique solution, the random variable $Y$ is a Gaussian random variable with mean zero, and covariance matrix $\Sigma_Y$ being the unique solution of the Lyapunov equation $H_f\Sigma_Y + \Sigma_Y H_f^\top = \Sigma$. Since for every sequence $\{Y^{(\alpha_k)}\}$, there is a weakly convergent subsequence $\{Y^{(\alpha_{k_\ell})}\}$ with a common weak limit, the sequence of random variables $\{Y^{(\alpha_k)}\}$ also converges weakly to the same random variable $Y$.

### 4.4.1 Proof of Theorem 4.2.1 (1)

To prove the result, we will apply [55, Proposition 2.1]. For completeness, we first state [55, Proposition 2.1] (using our notation) in the following.

**Proposition 4.4.1.** *Consider $\{X_k^{(\alpha)}\}$ generated by Equation 4.4. Suppose that*

*(a) There exists $L' > 0$ such that $\|\nabla f(x)\|_2 \leq L'(1 + \|x\|_2)$ for any $x \in \mathbb{R}^d$.*

*(b) There exist $\ell_1, \ell_2 > 0$ such that $\langle x, \nabla f(x)\rangle \geq \ell_1\|x\|_2^2 - \ell_2$ for all $x \in \mathbb{R}^d$.*

*(c) The noise sequence $\{w_k\}$ is an i.i.d. sequence satisfying $\mathbb{E}[w_k] = 0$ and $\mathbb{E}^{1/2}[\|w_k\|_2^2] \leq L''(1 + \|x_k\|_2)$ for all $k \geq 0$, where $L'' > 0$ is a constant.*

*Then, when the constant stepsize $\alpha < \frac{\ell_1-\sqrt{\max(\ell_1^2-(3L'^2+L''),0)}}{3L'^2+L''^2}$, the following statements hold.*

*(1) The iterates $\{X_k^{(\alpha)}\}$ admit a unique stationary distribution $\pi_\alpha$, which depends on the choice of $\alpha$. In addition, let $X^{(\alpha)} \sim \pi_\alpha$, then we have $\mathbb{E}[\|X^{(\alpha)}\|_2^2] < \infty$.*

(2) *For a test function $\phi : \mathbb{R}^d \mapsto \mathbb{R}$ satisfying $|\phi(x)| \leq L_\phi(1 + \|x\|_2)$ for all $x \in \mathbb{R}^d$ and some $L_\phi > 0$, and for any initialization $X_0^{(\alpha)} \in \mathbb{R}^d$ of the SGD algorithm given in Equation 4.4, there exists $\rho \in (0,1)$ and $\kappa$ (both depending on $\alpha$) such that we have $|\mathbb{E}[\phi(X_k^{(\alpha)})] - \pi_\alpha(\phi)| \leq \kappa\rho^k(1 + \|X_0^{(\alpha)}\|_2^2)$, where $\pi_\alpha(\phi) = \mathbb{E}[\phi(X^{(\alpha)})]$.*

Note that Proposition 4.4.1 (2) implies that $\{X_k^{(\alpha)}\}$ converges weakly to $X^{(\alpha)}$. To apply Proposition 4.4.1, we next verify the assumptions.

(a) Since the objective function $f(\cdot)$ is assumed to be $L$ – smooth, we have for any $x \in \mathbb{R}^d$ that $\|\nabla f(x) - \nabla f(0)\|_2 \leq L\|x\|_2$, which implies

$$\|\nabla f(x)\|_2 \leq \|\nabla f(0)\|_2 + L\|x\|_2 \leq \underbrace{\max(\|\nabla f(0)\|_2, L)}_{L'}(\|x\|_2 + 1).$$

(b) Since the objective function is assumed to be $\sigma$ – strongly convex, we have for any $x \in \mathbb{R}^d$:

$$f(0) - f(x) \geq \langle \nabla f(x), -x \rangle + \frac{\sigma}{2}\|x\|_2^2,$$

which implies that

$$\langle \nabla f(x), x \rangle \geq \frac{\sigma}{2}\|x\|_2^2 + f(x) - f(0) \geq \underbrace{\frac{\sigma}{2}}_{\ell_1}\|x\|_2^2 + \underbrace{f(x^*) - f(0) - 1}_{\ell_2}.$$

(c) This is immediately implied by Assumption 4.2.1, with $L'' = \text{Trace}(\Sigma)^{1/2}$.

Now apply Proposition 4.4.1, when the stepsize $\alpha$ satisfies $\alpha < \frac{\sigma}{2(3L'^2 + \text{Trace}(\Sigma))} := \bar{\alpha}$, the SGD iterates $\{X_k^{(\alpha)}\}$ converge weakly to some random variable $X^{(\alpha)}$, which is distributed according to the unique stationary distribution $\pi_\alpha$. In addition, we have $\mathbb{E}[\|X^{(\alpha)}\|_2^2] < \infty$. Since $Y_k^{(\alpha)}$ is the centered scaled variant of $X_k^{(\alpha)}$, the sequence $\{Y_k^{(\alpha)}\}$ converges weakly to some random variable $Y^{(\alpha)}$ and $\mathbb{E}[\|Y^{(\alpha)}\|_2^2] < \infty$.

### 4.4.2 Proof of Theorem 4.2.1 (2)

Following the road map described in the beginning of this section, we present and prove a sequence of lemmas in the following. Together they imply the desired result.

**Lemma 4.4.1.** *The family of random variables $\{Y^{(\alpha)}\}_{0<\alpha\leq\bar{\alpha}}$ is tight.*

*Proof of Lemma 4.4.1.* We first show that there exists an absolute constant $C > 0$ such that $\mathbb{E}[\|Y^{(\alpha)}\|^2] \leq C$ for any $\alpha \in (0, \alpha_0]$. Using the update equation (cf. Equation 4.1), we have

$$\begin{aligned} Y_{k+1}^{(\alpha)} &= Y_k^{(\alpha)} + \frac{\alpha}{g(\alpha)}\left(-\nabla f(Y_k^{(\alpha)} g(\alpha) + x^*) + w_k\right) \\ &= Y_k^{(\alpha)} - \sqrt{\alpha}\nabla f(\sqrt{\alpha}Y_k^{(\alpha)} + x^*) + \sqrt{\alpha}w_k \end{aligned}$$

The existence and uniqueness of a stationary distribution $Y^{(\alpha)}$ is proved in Part (1) of this theorem. We next show that the family of random variables $\{Y^{(\alpha)}\}_{0\leq\alpha\leq\bar{\alpha}}$ is tight. Using the equation

$$Y^{(\alpha)} \overset{D}{=} Y^{(\alpha)} - \sqrt{\alpha}\nabla f(\sqrt{\alpha}Y^{(\alpha)} + x^*) + \sqrt{\alpha}w,$$

and we have

$$\begin{aligned} \mathbb{E}[\|Y^{(\alpha)}\|_2^2] = \mathbb{E}[\|Y^{(\alpha)}\|_2^2] &+ \alpha\mathbb{E}\left[\left\|\nabla f(\sqrt{\alpha}Y^{(\alpha)} + x^*)\right\|_2^2\right] + \alpha\text{Trace}(\Sigma) \\ &- 2\sqrt{\alpha}\mathbb{E}\left[{Y^{(\alpha)}}^\top \nabla f(\sqrt{\alpha}Y^{(\alpha)} + x^*)\right]. \end{aligned}$$

By smoothness, we have

$$\left\|\nabla f(\sqrt{\alpha}Y^{(\alpha)} + x^*)\right\|_2^2 \leq L^2\alpha\|Y^{(\alpha)}\|_2^2.$$

By strong convexity, we have

$$\begin{aligned} {Y^{(\alpha)}}^\top \nabla f(\sqrt{\alpha}Y^{(\alpha)} + x^*) &= \frac{1}{\sqrt{\alpha}}(\sqrt{\alpha}Y^{(\alpha)} + x^* - x^*)^\top \left(\nabla f(\sqrt{\alpha}Y^{(\alpha)} + x^*) - \nabla f(x^*)\right) \\ &\geq \sigma\sqrt{\alpha}\|Y^{(\alpha)}\|_2^2. \end{aligned}$$

Therefore, we obtain

$$0 \leq L^2\alpha^2\mathbb{E}[\|Y^{(\alpha)}\|_2^2] + \alpha\text{Trace}(\Sigma) - 2\sigma\alpha\mathbb{E}[\|Y^{(\alpha)}\|_2^2].$$

When $\alpha \in (0, \bar{\alpha}) \subseteq (0, \frac{\sigma}{L^2})$, we have from the previous inequality that

$$\mathbb{E}[\|Y^{(\alpha)}\|_2^2] \leq \frac{\text{Trace}(\Sigma)}{2\sigma - L^2\alpha} \leq \frac{\text{Trace}(\Sigma)}{\sigma}. \tag{4.21}$$

Hence, for any $\alpha \in (0, \bar{\alpha})$, let $M = \sqrt{\text{Trace}(\Sigma)/\sigma\alpha}$, then we have

$$\mathbb{P}(\|Y^{(\alpha)}\| > M) \leq \frac{\mathbb{E}[\|Y^{(\alpha)}\|^2]}{M^2} \leq \frac{\text{Trace}(\Sigma)}{\sigma M^2} = \alpha.$$

It follows that the family of random variables $\{Y^{(\alpha)}\}_{0<\alpha\leq\alpha_0}$ is tight. □

**Lemma 4.4.2.** *Let $\{\alpha_k\}$ be a positive sequence of real numbers such that $\alpha_k \in (0, \bar{\alpha})$ for all $k$ and $\lim_{k\to\infty} \alpha_k = 0$. Suppose that $\{Y^{(\alpha_k)}\}$ converges weakly to some random variable $Y$. Then $Y$ verifies the following equation*

$$\mathbb{E}\left[\frac{t^\top \Sigma t}{2} e^{it^\top Y}\right] = -\mathbb{E}\left[\exp(it^\top Y) it^\top H_f Y\right]. \tag{4.22}$$

*Proof of Lemma 4.4.2.* For any $k \geq 0$, we have

$$Y^{(\alpha_k)} \overset{D}{=} Y^{(\alpha_k)} - \sqrt{\alpha_k}\nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*) + \sqrt{\alpha_k}w,$$

which implies for any $t \in \mathbb{R}^d$:

$$\mathbb{E}\left[e^{it^\top Y^{(\alpha_k)}}\right] = \mathbb{E}\left[\exp\left(it^\top Y^{(\alpha_k)}\right)\exp\left(-\sqrt{\alpha_k}it^\top \nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*)\right)\right]\mathbb{E}\left[e^{\sqrt{\alpha_k}it^\top w}\right] \tag{4.23}$$

Using Taylor's theorem and we have

$$\begin{aligned}
&\exp\left(-\sqrt{\alpha_k}it^\top \nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*)\right) \\
&= 1 - \sqrt{\alpha_k}it^\top \nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*) + \mathcal{O}\left(\alpha_k\|t\|^2\|\nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*\|^2\right).
\end{aligned}$$

Using Theorem 3.3.20 from [96] and we have

$$\mathbb{E}\left[e^{\sqrt{\alpha_k}it^\top Y^{(\alpha_k)}}\right] = 1 - \frac{\alpha_k t^\top \Sigma t}{2} + o(\alpha_k\|t\|^2).$$

Substituting the previous two inequalities into Equation 4.23 and we have

$$\begin{aligned}
&\mathbb{E}\left[e^{it^\top Y^{(\alpha_k)}}\right] \\
&= \mathbb{E}\left[\exp(it^\top Y^{(\alpha_k)})\exp(-\sqrt{\alpha_k}it^\top \nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*))\right]\mathbb{E}[e^{\sqrt{\alpha_k}it^\top w}] \\
&= \left(1 - \frac{\alpha_k t^\top \Sigma t}{2}\right) \times \mathbb{E}[\exp(it^\top Y^{(\alpha_k)})(1 - \sqrt{\alpha_k}it^\top \nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*) \\
&\quad + \mathcal{O}\left(\alpha_k\|t\|^2\|\nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*\|^2\right)] \\
&\quad + \mathbb{E}\left[\exp(it^\top Y^{(\alpha_k)})\exp(-\sqrt{\alpha_k}it^\top \nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*))\right]o(\alpha_k\|t\|^2) \\
&= \mathbb{E}\left[e^{it^\top Y^{(\alpha_k)}}\right] - \mathbb{E}\left[\frac{\alpha_k t^\top \Sigma t}{2}e^{it^\top Y^{(\alpha_k)}}\right] - \mathbb{E}\left[\exp(it^\top Y^{(\alpha_k)})\sqrt{\alpha_k}it^\top \nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*)\right] \\
&\quad + \mathbb{E}\left[\frac{\alpha_k t^\top \Sigma t}{2}\exp(it^\top Y^{(\alpha_k)})\sqrt{\alpha_k}it^\top \nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*)\right] \\
&\quad + \mathbb{E}\left[e^{it^\top Y^{(\alpha_k)}}\mathcal{O}\left(\alpha_k\|t\|^2\|\nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*\|^2\right)\right] \\
&\quad - \mathbb{E}\left[\frac{\alpha_k t^\top \Sigma t}{2}e^{it^\top Y^{(\alpha_k)}}\mathcal{O}\left(\alpha_k\|t\|^2\|\nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*\|^2\right)\right] \\
&\quad + \mathbb{E}\left[\exp(it^\top Y^{(\alpha_k)})\exp(-\sqrt{\alpha_k}it^\top \nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*))\right]o(\alpha_k\|t\|^2).
\end{aligned}$$

Simplify the above equality and we obtain

$$\begin{aligned}
&\underbrace{\mathbb{E}\left[\frac{t^\top \Sigma t}{2} e^{it^\top Y^{(\alpha_k)}}\right]}_{T_1} \\
= &-\underbrace{\mathbb{E}\left[\exp(it^\top Y^{(\alpha_k)}) \frac{it^\top \nabla f(\sqrt{\alpha_k} Y^{(\alpha_k)} + x^*)}{\sqrt{\alpha_k}}\right]}_{T_2} \\
&+\underbrace{\mathbb{E}\left[\frac{t^\top \Sigma t}{2} \exp(it^\top Y^{(\alpha_k)}) \sqrt{\alpha_k} it^\top \nabla f(\sqrt{\alpha_k} Y^{(\alpha_k)} + x^*)\right]}_{T_3} \\
&+\underbrace{\mathbb{E}\left[e^{it^\top Y^{(\alpha_k)}} \mathcal{O}\left(\|t\|^2 \|\nabla f(\sqrt{\alpha_k} Y^{(\alpha_k)} + x^*\|^2\right)\right]}_{T_4} \\
&-\underbrace{\mathbb{E}\left[\frac{t^\top \Sigma t}{2} e^{it^\top Y^{(\alpha_k)}} \mathcal{O}\left(\alpha_k \|t\|^2 \|\nabla f(\sqrt{\alpha_k} Y^{(\alpha_k)} + x^*\|^2\right)\right]}_{T_5} \\
&+\underbrace{\mathbb{E}\left[\exp(it^\top Y^{(\alpha_k)}) \exp(-\sqrt{\alpha_k} it^\top \nabla f(\sqrt{\alpha_k} Y^{(\alpha_k)} + x^*))\right] \frac{o(\alpha_k \|t\|^2)}{\alpha_k}}_{T_6}.
\end{aligned}$$

We next let $k$ go to infinity on both sides of the previous inequality and evaluate the limit of the terms $\{T_i\}_{1 \leq i \leq 6}$.

Since $\{Y^{(\alpha_k)}\}$ converges weakly to some random variable $Y$, we have by continuity theorem (Theorem 3.3.17 in [96]) that

$$\lim_{k \to \infty} \mathbb{E}\left[\frac{t^\top \Sigma t}{2} e^{it^\top Y^{(\alpha_k)}}\right] = \frac{t^\top \Sigma t}{2} \mathbb{E}\left[e^{it^\top Y}\right].$$

For the term $T_6$, we have by bounded convergence theorem that $\lim_{\alpha_k \to 0} T_6 = 0$. To evaluate the terms $T_2$ to $T_5$, the following definition and a result from [97] is needed.

**Definition 4.4.1.** A sequence of random variables $\{X_n\}$ is called asymptotically uniformly integrable if $\lim_{M \to \infty} \limsup_{n \to \infty} \mathbb{E}[|X_n| \mathbb{I}\{|X_n| > M\}] = 0$.

**Theorem 4.4.1** (Theorem 2.20 in [97]). *Let $f : \mathbb{R}^d \mapsto \mathbb{R}$ be measurable and continuous at* *every point in a set $C$. Let $X_n \Rightarrow X$, where $X$ takes its values in $C$. Then $\mathbb{E}[f(X_n)] \to \mathbb{E}[f(X)]$ if and only if the sequence of random variables $f(X_n)$ is asymptotically uniformly integrable.*

Now consider the term $T_2$. Observe that

$$\begin{aligned}
&\mathbb{E}\Bigg[ \left|\exp(it^\top Y^{(\alpha_k)})\frac{it^\top \nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*)}{\sqrt{\alpha_k}}\right| \times \\
&\quad \mathbb{I}\left\{\left|\exp(it^\top Y^{(\alpha_k)})\frac{it^\top \nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*)}{\sqrt{\alpha_k}}\right| > M\right\}\Bigg] \\
\le{}& \frac{1}{M}\mathbb{E}\Bigg[ \left|\exp(it^\top Y^{(\alpha_k)})\frac{it^\top \nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*)}{\sqrt{\alpha_k}}\right|^2 \times \\
&\quad \mathbb{I}\left\{\left|\exp(it^\top Y^{(\alpha_k)})\frac{it^\top \nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*)}{\sqrt{\alpha_k}}\right| > M\right\}\Bigg] \\
\le{}& \frac{1}{\alpha_k M}\mathbb{E}\Bigg[ \left|t^\top \nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*)\right|^2 \times \\
&\quad \mathbb{I}\left\{\left|\exp(it^\top Y^{(\alpha_k)})\frac{it^\top \nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*)}{\sqrt{\alpha_k}}\right| > M\right\}\Bigg] \\
\le{}& \frac{\|t\|^2}{\alpha_k M}\mathbb{E}\left[ \|\nabla f(\sqrt{\alpha_k}Y^{(\alpha_k)} + x^*)\|^2\right] && \text{(Cauchy Schwarz inequality)} \\
\le{}& \frac{L\|t\|^2}{M}\mathbb{E}\left[ \|Y^{(\alpha_k)}\|^2\right] && \text{(Definition of smoothness)} \\
\le{}& \frac{L\text{Trace}(\Sigma)\|t\|^2}{\sigma M}, && \text{(Eq. Equation 4.21)}
\end{aligned}$$

which goes to zero as $M \to \infty$. Therefore, we have by Theorem 4.4.1 that

$$\lim_{k\to\infty} T_2 = \mathbb{E}\left[\exp(it^\top Y)it^\top H_f Y\right].$$

Using the same line of analysis, we have $\lim_{k\to\infty} T_3 = \lim_{k\to\infty} T_4 = \lim_{k\to\infty} T_5 = 0$. It follows that

$$\mathbb{E}\left[\frac{t^\top \Sigma t}{2}e^{it^\top Y}\right] = -\mathbb{E}\left[\exp(it^\top Y)it^\top H_f Y\right].$$

Rearranging terms and we obtain the desired equation. □

**Lemma 4.4.3.** *Suppose that Equation 4.6 admits a unique solution. Then the random variable $Y$ given in Lemma 4.4.2 obeys the Gaussian distribution with mean zero, and covariance matrix $\Sigma_Y$ being the unique solution of the Lyapunov equation $H_f\Sigma_Y + \Sigma_Y H_f^\top = \Sigma$.*

*Proof of Lemma 4.4.3.* It is enough to verify that the multinormal distribution with mean zero and covariance matrix being the unique solution to the Lyapunov equation $H_f^\top\Sigma_Y + \Sigma_Y H_f = \Sigma$ solves Equation 4.6. The proof is given in the following:

$$\begin{aligned}
&\mathbb{E}\left[(2it^\top H_f Y + t^\top \Sigma t)e^{it^\top Y}\right]\\
&= C\int_{\mathbb{R}^d}(2it^\top H_f y + t^\top\Sigma t)e^{it^\top y}e^{-\frac{1}{2}y^\top\Sigma_Y^{-1}y}dy && (C = \tfrac{1}{\sqrt{(2\pi)^d det(\Sigma_Y)}})\\
&= Ce^{-\frac{1}{2}t^\top\Sigma_Y t}\int_{\mathbb{R}^d}(2it^\top H_f y + t^\top\Sigma t)e^{-\frac{1}{2}(y-i\Sigma_Y t)^\top\Sigma_Y^{-1}(y-i\Sigma_Y t)}dy\\
&= Ce^{-\frac{1}{2}t^\top\Sigma_Y t}\int_{\mathbb{R}^d}(2it^\top H_f(z+i\Sigma_Y t) + t^\top\Sigma t)e^{-\frac{1}{2}z^\top\Sigma_Y^{-1}z}dz && \text{(change of variable)}\\
&= Ce^{-\frac{1}{2}t^\top\Sigma_Y t}\int_{\mathbb{R}^d}(-2t^\top H_f\Sigma_Y t + t^\top\Sigma t)e^{-\frac{1}{2}z^\top\Sigma_Y^{-1}z}dz\\
&= Ce^{-\frac{1}{2}t^\top\Sigma_Y t}\int_{\mathbb{R}^d}(-t^\top(H_f\Sigma_Y + \Sigma_Y H_f^\top)t + t^\top\Sigma t)e^{-\frac{1}{2}z^\top\Sigma_Y^{-1}z}dz\\
&= Ce^{-\frac{1}{2}t^\top\Sigma_Y t}\int_{\mathbb{R}^d}(-t^\top\Sigma t + t^\top\Sigma t)e^{-\frac{1}{2}z^\top\Sigma_\alpha z}dz && \text{(The Lyapunov equation)}\\
&= 0.
\end{aligned}$$

□

Now that we have proved Theorem 4.2.1, we next provide the proofs of Theorem 4.2.2 and Theorem 4.2.3 below. First of all, linear SA can be reformulated as contractive SA and hence we only need to prove Theorem 4.2.3. To see this, note that the target equation $Ax + b = 0$ in linear SA is equivalent to $(\eta A + I)x + \eta b = x$ for any positive constant $\eta$. Define $\mathcal{T}(x) = (\eta A + I)x + \eta b$. When $A$ is Hurwitz, one can easily show that $\mathcal{T}(\cdot)$ is a contractive operator with respect to some weighted $\ell_2$-norm when $\eta$ is small enough. Since the proof of Theorem 4.2.3 is entirely similar to that of Theorem 4.2.1, we omit the details and only highlight the major steps.

The first step is to show that for a fixed stepsize $\alpha$, the centered scaled iterate $\{Y_k^\alpha\}$ converges weakly to a random variable $Y^{(\alpha)}$, which satisfies $\mathbb{E}[\|Y^{(\alpha)}\|_2^2] < \infty$ (i.e., Part (1) of Theorem 4.2.3). This can be proved by either following the same steps of proving Proposition 2.1 of [55], or directly applying Theorem 1.1 in [98].

To prove Part (2) of Theorem 4.2.3, we again establish a sequence of lemmas analogous to Lemmas 4.4.1, 4.4.2, and 4.4.3. The major difference is that, instead of repeatedly using the strong convexity and smoothness property in SGD, we utilize the contraction property of the operator $\mathcal{T}(\cdot)$ for contractive SA. Specifically, we have the following three lemmas, which together immediately give Theorem 4.2.3.

**Lemma 4.4.4.** *The family of random variables $\{Y^{(\alpha)}\}_{0<\alpha\leq\bar{\alpha}''}$ is tight.*

*Proof of Lemma 4.4.4.* Following from the same steps of proving Lemma 4.4.1, we have

$$\begin{aligned}\mathbb{E}[\|Y^{(\alpha)}\|_\mu^2] = \mathbb{E}[\|Y^{(\alpha)}\|_\mu^2] + \alpha\mathbb{E}\left[\left\|\mathcal{T}(\sqrt{\alpha}Y^{(\alpha)} + x^*) - (\sqrt{\alpha}Y^{(\alpha)} + x^*)\right\|_\mu^2\right] + \alpha\mathbb{E}[\|w\|_\mu^2]\\ + 2\sqrt{\alpha}\mathbb{E}\left[{Y^{(\alpha)}}^\top D(\mathcal{T}(\sqrt{\alpha}Y^{(\alpha)} + x^*) - (\sqrt{\alpha}Y^{(\alpha)} + x^*))\right],\end{aligned}$$

where $D = \text{diag}(\mu)$. To proceed, observe that we have for any $x \in \mathbb{R}^d$:

$$\begin{aligned}\|\mathcal{T}(x + x^*) - (x + x^*)\|_\mu &= \|\mathcal{T}(x + x^*) - \mathcal{T}(x^*) + x^* - (x + x^*)\|_\mu\\ &= \|\mathcal{T}(x + x^*) - \mathcal{T}(x^*)\|_\mu + \|x^* - (x + x^*)\|_\mu\\ &\leq (\gamma + 1)\|x\|_\mu\end{aligned}$$

and

$$\begin{aligned}x^\top D(\mathcal{T}(x + x^*) - (x + x^*)) &= x^\top D(\mathcal{T}(x + x^*) - x^*) - x^\top Dx\\ &\leq (\gamma - 1)\|x\|_\mu^2.\end{aligned}$$

It follows that

$$0 \leq \alpha^2(\gamma+1)^2\mathbb{E}\left[\|Y^{(\alpha)}\|_\mu^2\right] + \alpha\mathbb{E}[\|w\|_\mu^2] - 2\alpha(1-\gamma)\mathbb{E}\left[\|Y^{(\alpha)}\|_\mu^2\right],$$

which implies $\mathbb{E}[\|Y^{(\alpha)}\|_\mu^2] \leq \frac{\mathbb{E}[\|w\|_\mu^2]}{1-\gamma}$ for all $\alpha$ small enough. The rest of the proof follows by applying the Markov inequality, which is the same as in the proof of Lemma 4.4.1. □

**Lemma 4.4.5.** *Let $\{\alpha_k\}$ be a positive sequence of real numbers such that $\alpha_k \in (0, \bar{\alpha}'')$ for all $k$ and $\lim_{k\to\infty}\alpha_k = 0$. Suppose that $\{Y^{(\alpha_k)}\}$ converges weakly to some random variable $Y$. Then $Y$ verifies the following equation*

$$\mathbb{E}\left[\left(t^\top\Sigma t - 2it^\top(J-I)Y\right)e^{it^\top Y}\right] = 0, \quad \forall\, t \in \mathbb{R}^d. \tag{4.24}$$

*Proof of Lemma 4.4.5.* Following from the same steps of proving Lemma 4.4.2, we have

$$\mathbb{E}\left[\frac{t^\top\Sigma t}{2}e^{it^\top Y^{(\alpha_k)}}\right] = \mathbb{E}\left[\exp(it^\top Y^{(\alpha_k)})\frac{it^\top(\mathcal{T}(\sqrt{\alpha_k}Y^{(\alpha_k)}+x^*) - (\sqrt{\alpha_k}Y^{(\alpha_k)}+x^*))}{\sqrt{\alpha_k}}\right] + \sum_{j=3}^{6}N_j',$$

where $\{N_j\}_{3\leq j\leq 6}$ correspond to $\{T_j\}_{3\leq j\leq 6}$ in the proof of Lemma 4.4.2. Using the tightness property established in Lemma 4.4.4 and Theorem 4.4.1, letting $k$ go to infinity and we have from the previous inequality that

$$\mathbb{E}\left[\left(t^\top\Sigma t - 2it^\top(J-I)Y\right)e^{it^\top Y}\right] = 0, \quad \forall\, t \in \mathbb{R}^d.$$

□

**Lemma 4.4.6.** *Suppose that Equation 4.24 admits a unique solution. Then the random variable $Y$ given in Lemma 4.4.5 obeys the Gaussian distribution with mean zero, and covariance matrix $\Sigma_Y$ being the unique solution of the Lyapunov equation $(J-I)\Sigma_Y +$*

$\Sigma_Y(J-I)^\top + \Sigma = 0.$

*Proof of Lemma 4.4.6.* The proof is identical to that of Lemma 4.4.3, where we verify that the desired Gaussian distribution solves Equation 4.24 by using the Lyapunov equation. □

## 4.5 Conclusion and Future Work

In this chapter, we characterize the asymptotic stationary distribution of properly centered scaled iterate of SA algorithms. In particular, we show that for (1) SGD with smooth and strongly convex objective, (2) linear SA, and (3) contractive SA, the scaling function is $g(\alpha) = \sqrt{\alpha}$ and the corresponding stationary distributions are Gaussian distributions with mean zero and covariance matrices being solutions of appropriate Lyapunov equations. For SA beyond these cases, we empirical show that the stationary distribution need not be Gaussian, and provide a heuristic method to determine the suitable scaling function. Theoretically studying more general SA algorithms is an immediate future direction of this work.

One benefit from characterizing the stationary distribution of the centered scaled iterates is that we can use this result as a guideline to design stochastic approximation algorithms with improved performance. For example, a possible future direction is to investigate how to modify SA algorithms so that the limiting random variable $Y$ has a smaller covariance matrix. This is related to various variance-reduction techniques in SA. Another possible future direction is to characterize the convergence rate of $Y^{(\alpha)}$ to $Y$ as the constant stepsize $\alpha$ goes to zero using Stein's method. This in conjunction with full characterization of the distribution of $Y$ will enable us to study the distribution of $Y^{(\alpha)}$ in the non-asymptotic regime.

# Part II

# RL with a Tabular Representation

# CHAPTER 5
# PRELIMINARIES

From now on, we will dive into RL. The RL problem is usually modeled as an MDP. However, unlike MDP, the environmental model (e.g. transition probabilities and the reward function) is unknown in RL. As a result, typical algorithms for solving MDPs such as value iteration and policy iteration are not implementable in RL because carrying out those algorithms requires using the environmental model, which is unknown. To overcome this difficulty, RL agent implements data-driven stochastic iterative algorithms, i.e., SA algorithms.

At a high level, RL algorithms can be divided into two categories: value-based algorithms and policy-based algorithms. In value-based algorithms, the agent aims at learning the optimal value function (or state-action value function), which is then used to compute an optimal policy via the Bellman optimality equation. Typical examples of value-based algorithms are $Q$-learning, and its on-policy variant SARSA.

Unlike value-based algorithms, policy-based algorithms directly work with policies. In each iteration, the agent first perform policy evaluation to estimate the value function of the current policy iterate, which is then used to update the policy via either approximate policy iteration or policy gradient.

In Part II of the thesis, we focus on value-based RL algorithms with a tabular representation. Specifically, we consider various on-policy TD-learning algorithms (e.g. $n$-step TD and TD($\lambda$)), various off-policy TD-learning ($Q^\pi(\lambda)$, Retrace($\lambda$), and $Q$-trace, etc., and $Q$-learning, and establish their finite-sample guarantees. The major theoretical workhorse used here is the results on Markovian SA under contractive operators presented in Chapter 2.

### 5.1 Problem Formulation

In this thesis, we consider modeling the RL problem as an infinite horizon discounted MDP defined by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is a finite state-space, $\mathcal{A}$ is a finite action-space, $\mathcal{P} = \{P_a \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|} \mid a \in \mathcal{A}\}$ is a set of action-dependent transition probability matrices, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto [-1, 1]$ is a reward function, and $\gamma \in (0, 1)$ is a discount factor. The transition probabilities and the reward function together are called the environmental model of the MDP. Importantly, in RL, the environmental model is *unknown* to the agent.

At each time step $k \geq 0$, the agent is at a certain state of the environment, denoted by $S_k$, and selects an action $A_k$ according to some chosen policy $\pi$, where $\pi$ is a (possibly stochastic) mapping from the state-space to the action-space. Then the agent moves to a new state $S_{k+1}$ based on the underlying transition probabilities, i.e., $S_{k+1} \sim P_{A_k}(S_k, \cdot)$, and receives a one-stage reward $\mathcal{R}(S_k, A_k)$. This process is then repeated, and the goal is to find an optimal policy of selecting actions to maximize the long-term reward.

The performance of a policy $\pi$ is captured by its value function $V^\pi : \mathcal{S} \mapsto \mathbb{R}$, which is defined by

$$V^\pi(s) = \mathbb{E}_\pi \left[ \sum_{k=0}^{\infty} \gamma^k \mathcal{R}(S_k, A_k) \;\middle|\; S_0 = s \right], \ \forall\, s \in \mathcal{S},$$

where we use $\mathbb{E}_\pi[\ \cdot\ ]$ to indicate that the actions are selected according to the policy $\pi$. Since we work with MDPs with finite state-action spaces, $V^\pi$ can equivalently be viewed as a vector in $\mathbb{R}^{|\mathcal{S}|}$. With the value function defined above, the goal of RL is to find an optimal policy $\pi^*$ such that its associated value function, denoted by $V^*$, is maximized uniformly across the states, i.e.,

$$V^*(s) \geq V^\pi(s), \ \forall\, s \in \mathcal{S}, \ \forall\, \pi.$$

For discounted MDPs, an optimal policy always exists [9].

In the rest of this chapter, we will present various value-based RL algorithms and establish their finite-sample guarantees. The idea is to reformulate the RL algorithm in the form of a Markovian SA algorithm under a contractive operator (i.e., Algorithm 1) and then apply Theorem 2.5.1.

## 5.2 Illustration via $Q$-Learning

To illustrate the recipe of applying our SA results to RL algorithms, we use the popular $Q$-learning algorithm as an example. The $Q$-learning algorithm is a recursive approach for finding the optimal policy corresponding to an MDP (see Chapter 8 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

$$Q_{k+1}(s,a) = Q_k(s,a) + \alpha_k \left( \mathcal{R}(S_k, A_k) + \gamma \max_{a' \in \mathcal{A}} Q_k(S_{k+1}, a') - Q_k(S_k, A_k) \right) \tag{5.1}$$

when $(s,a) = (S_k, A_k)$, and $Q_{k+1}(s,a) = Q_k(s,a)$ otherwise.

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, i.e., $\mathcal{R}(S_k, A_k) + \gamma \max_{a' \in \mathcal{A}} Q_k(S_{k+1}, a')$, at time step $k$. For simplicity of notation, we denote $\Gamma_1(Q,s,a,s')=\mathcal{R}(s,a)+\gamma\max_{a'\in\mathcal{A}}Q(s',a')-Q(s,a)$.

To apply our SA results, the first step is to reformulate asynchronous $Q$-learning in the form of Algorithm 1 by introducing an operator $F(\cdot,\cdot)$ and a Markov chain $\{Y_k\}$ that captures asynchronous updates along a trajectory. 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 by $[F(Q,s_0,a_0,s_1)](s,a)=\mathbb{1}_{\{(s_0,a_0)=(s,a)\}}\Gamma_1(Q,s_0,a_0,s_1)+Q(s,a)$ for all $(s,a)$. Then the $Q$-learning algorithm given in Equation 5.1 can be rewritten as:

$$Q_{k+1}=Q_k+\alpha_k\left(F(Q_k,S_k,A_k,S_{k+1})-Q_k\right), \tag{5.2}$$

which is in the form of Algorithm 1 with $x_k$ replaced by $Q_k$, $w_k=0$, and $Y_k=(S_k,A_k,S_{k+1})$. The key takeaway is that in Equation 5.2, the various noise terms (both due to performing asynchronous update and due to samples replacing an expectation in the Bellman equation) are encoded through introducing the operator $F(\cdot)$ and the associated evolution of the Markovian noise $\{Y_k\}$.

With 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 $\bar{F}(Q)=N\mathcal{H}(Q)+(I-N)Q$. Here $\mathcal{H}(\cdot)$ is the Bellman optimality operator for the $Q$-function [11]. The matrix $N$ is a diagonal matrix with $\{p(s,a)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$ 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 the $Q$-learning algorithm in that, at steady-state, $Q_k(s,a)$ is updated with probability $p(s,a)$, and remains unchanged otherwise. Moreover, since it is well-known that $\mathcal{H}(\cdot)$ is a contraction mapping with respect to $\|\cdot\|_\infty$, we also show that $\bar{F}(\cdot)$ is a contraction mapping with respect to $\|\cdot\|_\infty$, with the optimal $Q$-function being its unique fixed-point.

The SA reformulation together with 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$(\lambda)$ can all be modeled by Markovian SA algorithms involving a contraction mapping (possibly with respect to different norm), and Markovian noise. Therefore, our SA results provide a *unified* recipe for the finite-sample analysis of value-based RL algorithms.

# CHAPTER 6

# ON-POLICY PREDICTION: THE EFFICIENCY OF BOOTSTRAPPING

## 6.1 Introduction

Although the ultimate goal of RL is to find an optimal policy, there is usually a small goal, which is to estimate the value function of a given policy. This is called the prediction problem, or the policy evaluation problem. Solving the prediction problem is important for several reasons. First of all, suppose we are given a policy. Before implementing the policy in practice, we need to have an estimate on how good (or how safe) the policy is. More importantly, solving the prediction problem is usually an intermediate step to eventually find an optimal policy. For example, the popular actor-critic algorithm iteratively performs policy evaluation and policy improvement to solve the RL problem.

Formally, the prediction problem refers to the problem of estimating the value function $V^\pi$ (or state-action value function $Q^\pi$) of a given policy $\pi$, which we call the *target policy*. The most popular approach to solve the policy evaluation problem is TD-learning. To motivate the TD-learning algorithm, we next introduce the Bellman equation for $V^\pi$. Let $\mathcal{T}^\pi : \mathbb{R}^{|\mathcal{S}|} \mapsto \mathbb{R}^{|\mathcal{S}|}$ be the Bellman operator (associated with policy $\pi$) defined by

$$[\mathcal{T}^\pi(V)](s) = \mathbb{E}_\pi \left[\mathcal{R}(S_k, A_k) + \gamma V(S_{k+1}) \mid S_k = s\right], \; \forall \, s \in \mathcal{S}.$$

Then it was shown in the literature that $V^\pi$ uniquely solves the following fixed-point equation:

$$V^\pi = \mathcal{T}^\pi(V^\pi). \tag{6.1}$$

Therefore, to find $V^\pi$, it is enough to solve Equation 6.1. Since $\mathcal{T}^\pi(\cdot)$ is known to be a contraction mapping (with respect to the $\ell_\infty$-norm), with contraction factor $\gamma$, Equation 6.1 can be efficiently solved with the fixed-point iteration $V_{k+1} = \mathcal{T}^\pi(V_k)$. However, since computing $\mathcal{T}^\pi(V_k)$ requires using the underlying transition probability matrices of the MDP, we are not able to carry out such fixed-point iteration algorithm in the RL setting.

The TD-learning algorithm is designed to solve Equation 6.1 using the SA method, In addition to vanilla TD-learning, there are other variants of TD-learning algorithms such as $n$-step TD and TD($\lambda$). The $n$-step TD-learning algorithm is designed to solve the $n$-step Bellman equation $V^\pi = (\mathcal{T}^\pi)^n(V^\pi)$, and the TD($\lambda$) algorithm is designed to solve the $\lambda$-discounted Bellman equation $V^\pi = (1-\lambda)\sum_{n=1}^\infty \lambda^{n-1}(\mathcal{T}^\pi)^n(V^\pi)$. Both of the above variants of the Bellman equation are equivalent to the original Bellman equation (cf. Equation 6.1) in the sense that they all have $V^\pi$ as their unique solution. However, the induced SA algorithms (i.e., $n$-step TD and TD($\lambda$)) are different.

In $n$-step TD and TD($\lambda$), there is an important open problem, which is called the efficiency of bootstrapping [18]. Formally, it refers to the problem about how to choose the tunable parameters $n$ and $\lambda$ so that $n$-step TD and TD($\lambda$) achieve their optimal performance. In the rest of this chapter, we will establish finite-sample guarantees of both $n$-step TD and TD($\lambda$), and provide theoretical insights into the problem about the efficiency of bootstrapping.

## 6.2 Finite-Sample Analysis of $n$-Step TD

In this section, we present the $n$-step TD-learning algorithm for solving the prediction problem, and establish its finite-sample guarantees.

### 6.2.1 The $n$-Step TD-Learning Algorithm

We begin by presenting the $n$-step TD-learning algorithm in the following.

Observe that in Algorithm 3, the policy used to collect samples (called behavior policy, or sampling policy) is the target policy $\pi$. This is called on-policy sampling. When the

**Algorithm 3** $n$-Step TD-Learning

```
Input: Integer $K$, initialization $V_0 \in \mathbb{R}^{|\mathcal{S}|}$, and a trajectory of samples $\{(S_k, A_k)\}_{0\leq k\leq K+n-1}$ collected under the target policy $\pi$
for $k = 0, 1, \cdots, K-1$ do
    $V_{k+1}(S_k) = V_k(S_k) + \alpha_k(\sum_{i=k}^{k+n-1} \gamma^{i-k}\mathcal{R}(S_i, A_i) + \gamma^n V_k(S_{k+n}) - V_k(S_k))$
end for
Output: $V_K$
```

behavior policy is different than the target policy, the corresponding algorithm is called off-policy learning. We will study off-policy variants of TD-learning in Chapter 7.

In view of Algorithm 3, $n$-step TD-learning performs asynchronous update in the sense that only a single entry of the vector-valued iterate $V_k$ is updated in each time step. Moreover, the update can be viewed as a sample estimate of the difference between the LHS and the RHS of the $n$-step Bellman equation.

An important idea in $n$-step TD is to use the parameter $n$ to adjust the bootstrapping effect. When $n = 0$, Algorithm 3 is the standard 1-step TD update, which corresponds to extreme bootstrapping. When $n = \infty$, Algorithm 3 is the Monte Carlo method for estimating $V^\pi$, which corresponds to no bootstrapping. A long-standing question in RL is about the efficiency of bootstrapping, i.e., the choice of $n$ that leads to the optimal performance of Algorithm 3 [1].

In the following subsections, we will establish finite-sample convergence bounds of the $n$-step TD-learning algorithm. By evaluating the resulting sample complexity bound as a function of $n$, we provide theoretical insights into the bias-variance trade-off in terms of $n$, as well as an estimate of the optimal value of $n$. To proceed, we make the following assumption about the target policy $\pi$.

**Assumption 6.2.1.** The Markov chain $\mathcal{M}_\mathcal{S} = \{S_k\}$ induced by the target policy $\pi$ is irreducible and aperiodic.

Since we are using on-policy sampling in $n$-step TD, the target policy must enable the agent to sufficiently explore the state-space. Assumption 6.2.1 ensures this property, and also implies that $\{S_k\}$ has a unique stationary distribution (denoted by $\kappa_S \in \Delta^{|\mathcal{S}|}$), and the geometric mixing property [48].

### 6.2.2 Properties of the $n$-Step TD-Learning Algorithm

Our plan is to reformulate $n$-step TD as a Markovian SA algorithm of the form Algorithm 1, and then apply Theorem 2.5.1 to establish the finite-sample bounds.

We begin with the reformulation. Let a sequence $\{Y_k\}$ be defined by

$$Y_k = (S_k, A_k, ..., S_{k+n-1}, A_{k+n-1}, S_{k+n}),\ \forall\, k \geq 0.$$

It is clear that $\{Y_k\}$ is a Markov chain, whose state-space is denoted by $\mathcal{Y}$ and is finite. Define an operator $F : \mathbb{R}^{|\mathcal{S}|} \times \mathcal{Y} \mapsto \mathbb{R}^{|\mathcal{S}|}$ by

$$\begin{aligned}[F(V, y)](s) &= [F(V, s_0, a_0, ..., s_n)](s)\\ &= \mathbb{1}_{\{s_0=s\}} \left(\sum_{i=0}^{n-1} \gamma^i \mathcal{R}(s_i, a_i) + \gamma^n V(s_n) - V(s_0)\right) + V(s),\ \forall\, s \in \mathcal{S}.\end{aligned}$$

Then the update equation of $n$-step TD (i.e., line $3$ of Algorithm 3) can be equivalently written by

$$V_{k+1} = V_k + \alpha_k(F(V_k, Y_k) - V_k), \tag{6.2}$$

which is in the form of Algorithm 1 with $w_k \equiv 0$. We next establish the properties of the $n$-step TD algorithm in the following proposition, which enables us to apply Theorem 2.5.1. Let $\mathcal{K}_S \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ be a diagonal matrix with diagonal entries $\{\kappa_S(s)\}_{s\in\mathcal{S}}$, and let $\mathcal{K}_{S,\min} = \min_{s\in\mathcal{S}} \kappa_S(s)$.

**Proposition 6.2.1.** *Under Assumption 6.2.1, the $n$-step TD-learning algorithm has the following properties.*

*(1) The operator $F(\cdot,\cdot)$ satisfies:*

*(a) $\|F(V_1,y)-F(V_2,y)\|_2 \leq 2\|V_1-V_2\|_2$ for all $V_1,V_2\in\mathbb{R}^{|\mathcal{S}|}$ and $y\in\mathcal{Y}$.*

*(b) $\|F(\mathbf{0},y)\|_2 \leq \frac{1}{1-\gamma}$ for all $y\in\mathcal{Y}$.*

*(2) The Markov chain $\{Y_k\}$ has a unique stationary distribution, denoted by $\mu_Y$. Moreover, there exist $C>0$ and $\sigma\in(0,1)$ such that*

$$max_{y\in\mathcal{Y}}\|P_\pi^{k+n}(y,\cdot)-\mu_Y(\cdot)\|_{TV}\leq C\sigma^k,\ \forall\ k\geq 0.$$

*(3) Define an operator $\bar{F}:\mathbb{R}^{|\mathcal{S}|}\mapsto\mathbb{R}^{|\mathcal{S}|}$ by $\bar{F}(V)=\mathbb{E}_{Y\sim\mu_Y}[F(V,Y)]$ for all $V\in\mathbb{R}^{|\mathcal{S}|}$. Then*

*(a) $\bar{F}(\cdot)$ is explicitly given by*

$$\bar{F}(V)=\left[I-\mathcal{K}_S\sum_{i=0}^{n-1}(\gamma P_\pi)^i(I-\gamma P_\pi)\right]V+\mathcal{K}_S\sum_{i=0}^{n-1}(\gamma P_\pi)^iR_\pi,$$

*where $R_\pi\in\mathbb{R}^{|\mathcal{S}|}$ is defined by $\mathcal{R}_\pi(s)=\sum_{a\in\mathcal{A}}\pi(a|s)\mathcal{R}(s,a)$ for all $s\in\mathcal{S}$.*

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

*(c) $\bar{F}(\cdot)$ has a unique fixed-point $V^\pi$.*

From Proposition 6.2.1, we see that the asynchronous Bellman operator $\bar{F}(\cdot)$ associated with the on-policy $n$-step TD-learning algorithm is a $\beta$-contraction with respect to $\|\cdot\|_p$ for any $p\in[1,\infty]$. In particular, this implies that $\bar{F}(\cdot)$ is a contraction with respect to the standard Euclidean norm $\|\cdot\|_2$. This is the property we are going to exploit in establishing the finite-sample bounds of $n$-step TD in the next subsection.

To intuitively understand the $\|\cdot\|_2$-contraction property, recall a “less known” property from [92, 11] that the $n$-step Bellman operator $\mathcal{T}_\pi^n(\cdot)$ is a contraction operator with respect to the weighted $\ell_2$-norm $\|\cdot\|_{\kappa_S}$, with weights being the stationary distribution $\kappa_S$. Similar to $Q$-learning, the asynchronous Bellman operator $\bar{F}(\cdot)$ is a convex combination of the identity operator and the $n$-step Bellman operator $\mathcal{T}_\pi^n(\cdot)$, using the stationary distribution $\kappa_S$ as weights. Therefore, due to this "normalization", the asynchronous Bellman operator is a contraction mapping with respect to the unweighted $\ell_2$-norm.

### 6.2.3 Finite-Sample Bounds of $n$-Step TD

In this subsection, we use the $\|\cdot\|_2$-contraction property from Proposition 6.2.1 to derive finite-sample convergence bounds of Algorithm 3. Define

$$t_\delta = \min\left\{k \geq 0 : \max_{s\in\mathcal{S}} \|P_\pi^k(s,\cdot) - \kappa_S(\cdot)\|_{\text{TV}} \leq \delta\right\}$$

as the mixing time of the Markov chain $\{S_k\}$ (under policy $\pi$) with precision $\delta$. For simplicity, we here only present the case for using constant stepsize.

**Theorem 6.2.1.** *Consider $\{V_k\}$ of Algorithm 3. Suppose that Assumption 6.2.1 is satisfied, and $\alpha_k \equiv \alpha$ with $\alpha$ chosen such that $\alpha(t_\alpha + n) \leq \hat{c}_0(1-\beta)$ (where $\hat{c}_0$ is a numerical constant). Then we have for all $k \geq t_\alpha + n$:*

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

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

To analyze the impact of the parameter $n$, we begin by rewriting the convergence bounds in Theorem 6.2.1 focusing only on $n$-dependent terms. Using the explicit expression of the contraction factor $\beta$, in the $k$-th iteration, the bias term is of the size $(1 - \Theta(1-\gamma^n))^k$. Since the mixing time $t_\alpha$ of the original Markov chain $\{S_k\}$ does not depend on $n$, the variance term is of the size $\mathcal{O}(n/(1-\gamma^n))$. Now we can clearly see that as $n$ increases, the bias goes down while the variance goes up, thereby demonstrating a bias-variance trade-off in the $n$-step TD-learning algorithm.

To formally characterize how the parameters of the $n$-step TD algorithm impact its convergence rate and to compute an estimate of the optimal choice of $n$, we next derive the sample complexity of $n$-step TD based on Theorem 6.2.1.

**Corollary 6.2.1.** *In order to make $\mathbb{E}][\|V_k - V^\pi\|_2] \leq \epsilon$, the number of samples required for the $n$-step TD-learning algorithm is of the size*

$$\underbrace{\mathcal{O}\left(\frac{\log^2(1/\epsilon)}{\epsilon^2}\right)}_{\textit{Accuracy}}\underbrace{\tilde{\mathcal{O}}\left(\frac{1}{(1-\gamma)^2}\right)}_{\textit{Effective horizon}}\underbrace{\tilde{\mathcal{O}}\left(\frac{n}{(1-\gamma^n)^2}\right)}_{\textit{Parameter } n}\underbrace{\tilde{\mathcal{O}}(\mathcal{K}_{S,\min}^{-2})}_{\textit{Quality of exploration}}\quad\tilde{\mathcal{O}}(|\mathcal{S}|^{1/2})$$

Note that we used $\|V^\pi\|_2 \leq |\mathcal{S}|^{1/2}/(1-\gamma)$ in deriving the sample complexity.

In light of the dependence on the parameter $n$, which is $\tilde{\mathcal{O}}(n(1-\gamma^n)^{-2})$, the optimal choice of $n$ can be estimated by minimizing the function $n(1-\gamma^n)^{-2}$ over all positive integers. By doing that, we obtain the following estimate:

$$n_{\text{optimal}} \sim \min\left(1, \lfloor 1/\log(1/\gamma) \rceil\right),$$

where $\lfloor x \rceil$ stands for the integer closest to $x$. This result implies that when the discount factor $\gamma$ is small (specifically $\gamma \leq 1/e$), there is not much improvement in using multi-step TD-learning over using single step TD-learning, and when the discount factor is large, using $n$-step TD-learning with $n \sim \lfloor 1/\log(1/\gamma) \rceil$ has provable improvement.

### 6.2.4 Related Literature

The concept of using multi-step returns instead of only one-step return was introduced in [99]. See [1, Chapter 7] for more details about $n$-step TD. The asymptotic convergence of $n$-step TD can be established using the general stochastic approximation algorithm under contraction assumption [11]. Regarding the choice of $n$, it was observed in empirical experiments that $n$-step TD (with a suitable choice of $n$) usually outperforms both 1-step TD and Monte Carlo method [100, 1]. However, theoretical understanding to this phenomenon is not well established in the literature. We derive finite-sample convergence bounds of the $n$-step TD-learning algorithm as an explicit function of $n$. This requires us to compute the exact expression of the contraction factor $\beta$ of the asynchronous Bellman operator (cf. Proposition 6.2.1 (3)), and the mixing time (cf. Proposition 6.2.1 (2)).

## 6.3 Proof of All Theoretical Results in Section 6.2

### 6.3.1 Proof of Proposition 6.2.1

(1) (a) For any $V_1, V_2 \in \mathbb{R}^{|\mathcal{S}|}$ and $y \in \mathcal{Y}$, we have

$$\begin{aligned}
&\|F(V_1, y) - F(V_2, y)\|_2 \\
&= \Bigg(\sum_{s\in\mathcal{S}} \big[\mathbb{1}_{\{s_0=s\}} \left(\gamma^n(V_1(s_n) - V_2(s_n)) - (V_1(s_0) - V_2(s_0))\right) \\
&\quad + V_1(s) - V_2(s)\big]^2\Bigg)^{1/2} \\
&\leq \left(\sum_{s\in\mathcal{S}}[\mathbb{1}_{\{s_0=s\}}(\gamma^n + 1)\|V_1 - V_2\|_2]^2\right)^{1/2} + \|V_1 - V_2\|_2 \qquad \text{(triangle inequality)} \\
&\leq 3\|V_1 - V_2\|_2.
\end{aligned}$$

(1) (b) For any $y \in \mathcal{Y}$, we have

$$\begin{aligned}
\|F(\mathbf{0}, y)\|_2^2 &= \sum_{s\in\mathcal{S}} \left(\mathbb{1}_{\{s_0=s\}} \sum_{i=0}^{n-1} \gamma^i \mathcal{R}(s_i, a_i)\right)^2 \\
&\leq \sum_{s\in\mathcal{S}} \mathbb{1}_{\{s_0=s\}} \left(\sum_{i=0}^{n-1} \gamma^i\right)^2 \\
&\leq \frac{1}{(1-\gamma)^2}.
\end{aligned}$$

It follows that $\|F(\mathbf{0}, y)\|_2 \leq \frac{1}{1-\gamma}$.

(2) Since the Markov chain $\{S_k\}$ induced by the target policy $\pi$ is irreducible and aperiodic, there exists $C > 0$ and $\sigma \in (0,1)$ such that $\max_{s\in\mathcal{S}} \|P_\pi^k(s,\cdot) - \|_S(\cdot)\|_{\text{TV}} \leq C\sigma^k$ for all $k \geq 0$ [48]. Now consider the Markov chain $\{Y_k\}$. We have for all $k \geq 0$:

$$\begin{aligned}
&\max_{y\in\mathcal{Y}} \left\|P_\pi^{k+n+1}(y,\cdot) - \mu_Y(\cdot)\right\|_{\text{TV}} \\
&= \frac{1}{2} \max_{s_0,a_0,\dots,s_n,a_n} \sum_{s_0',a_0',\dots,s_n',a_n'} \left| \sum_s P_{a_n}(s_n,s) P_\pi^k(s,s_0') - \kappa_S(s_0') \right| \times \\
&\quad \pi(a_0'|s_0') \prod_{i=0}^{n-1} P_{a_i'}(s_i',s_{i+1}') \pi(a_{i+1}'|s_{i+1}') \\
&\qquad\qquad (P_a \text{ is the transition probability matrix of the MDP under action } a) \\
&\leq \frac{1}{2} \max_{s_n,a_n} \sum_{s_0'} \left| \sum_s P_{a_n}(s_n,s) P_\pi^k(s,s_0') - \kappa_S(s_0') \right| \\
&= \frac{1}{2} \max_{s_n,a_n} \sum_s P_{a_n}(s_n,s) \sum_{s_0'} \left| P_\pi^k(s,s_0') - \kappa_S(s_0') \right| \\
&\leq \frac{1}{2} \max_s \sum_{s_0'} \left| P_\pi^k(s,s_0') - \kappa_S(s_0') \right| \\
&= \max_{s\in\mathcal{S}} \left\|P_\pi^k(s,\cdot) - \kappa_S(\cdot)\right\|_{\text{TV}} \\
&\leq C\sigma^k.
\end{aligned}$$

(3) (a) Since the $n$-step Bellman operator is explicitly given by

$$(\mathcal{T}^\pi)^n(V) = \left[\sum_{i=0}^{n-1} (\gamma P_\pi)^i (I - \gamma P_\pi)\right] V + \sum_{i=0}^{n-1} (\gamma P_\pi)^i R_\pi$$

for any $V \in \mathbb{R}^{|\mathcal{S}|}$, we have

$$\bar{F}(V) = \left[I - \mathcal{K}_S \sum_{i=0}^{n-1} (\gamma P_\pi)^i (I - \gamma P_\pi)\right] V + \mathcal{K}_S \sum_{i=0}^{n-1} (\gamma P_\pi)^i R_\pi.$$

(3) (b) For any $V_1, V_2 \in \mathbb{R}^{|\mathcal{S}|}$ and $p \geq 1$, we have

$$\begin{aligned}\|\bar{F}(V_1) - \bar{F}(V_2)\|_p &= \left\| \left[ I - \mathcal{K}_S \sum_{i=0}^{n-1} (\gamma P_\pi)^i (I - \gamma P_\pi) \right] (V_1 - V_2) \right\|_p \\ &\leq \left\| I - \mathcal{K}_S \sum_{i=0}^{n-1} (\gamma P_\pi)^i (I - \gamma P_\pi) \right\|_p \|V_1 - V_2\|_p.\end{aligned}$$

For simplicity of notation, we denote $G = I - \mathcal{K}_S \sum_{i=0}^{n-1} (\gamma P_\pi)^i (I - \gamma P_\pi)$. Since

$$\begin{aligned}G &= I - \mathcal{K}_S \sum_{i=0}^{n-1} (\gamma P_\pi)^i (I - \gamma P_\pi) \\ &= I - \mathcal{K}_S \sum_{i=0}^{n-1} (\gamma P_\pi)^i + \mathcal{K}_S \sum_{i=0}^{n-1} (\gamma P_\pi)^{i+1} \\ &= I - \mathcal{K}_S - \mathcal{K}_S \sum_{i=1}^{n-1} (\gamma P_\pi)^i + \mathcal{K}_S \sum_{i=1}^{n} (\gamma P_\pi)^i \\ &= I - \mathcal{K}_S + \mathcal{K}_S (\gamma P_\pi)^n,\end{aligned}$$

we see that the matrix $G$ has non-negative entries. Therefore, we have

$$\|G\|_\infty = \|G\mathbf{1}\|_\infty = \left\| \mathbf{1} - \kappa_S \sum_{i=0}^{n-1} \gamma^i (1 - \gamma) \right\|_\infty = 1 - \mathcal{K}_{S,\min}(1 - \gamma^n).$$

Moreover, using the fact that $\kappa_S$ is the stationary distribution of $P_\pi$ (i.e., $\kappa_S^\top P_\pi = \kappa_S^\top$), we have

$$\|G\|_1 = \|\mathbf{1}^\top G\|_\infty = \left\| \mathbf{1}^\top - \kappa_S^\top \sum_{i=0}^{n-1} \gamma^i (1 - \gamma) \right\|_\infty = 1 - \mathcal{K}_{S,\min}(1 - \gamma^n).$$

To proceed, we need the following lemma.

**Lemma 6.3.1.** *Let $G \in \mathbb{R}^{d \times d}$ be a matrix with non-negative entries. Then we have for all*

$p \in [1, \infty]$*:*

$$\|G\|_p \leq \|G\|_1^{1/p} \|G\|_\infty^{1-1/p}.$$

*Proof of Lemma 6.3.1.* The result clearly holds when $p = 1$ or $p = \infty$. Now consider $p \in (1, \infty)$. Using the definition of induced matrix norm, we have for any $x \neq 0$:

$$\begin{aligned}
\|Gx\|_p^p &= \sum_{i=1}^d \left( \sum_{j=1}^d G_{ij} x_j \right)^p \\
&= \sum_{i=1}^d [G\mathbf{1}]_i^p \left( \sum_{j=1}^d \frac{G_{ij}}{[G\mathbf{1}]_i} x_j \right)^p \\
&\leq \sum_{i=1}^d [G\mathbf{1}]_i^{p-1} \sum_{j=1}^d G_{ij} x_j^p && \text{(Jensen's inequality)} \\
&\leq \|G\|_\infty^{p-1} \sum_{j=1}^d x_j^p \sum_{i=1}^d G_{ij} \\
&= \|G\|_\infty^{p-1} \sum_{j=1}^d x_j^p [\mathbf{1}^\top G]_j \\
&\leq \|G\|_\infty^{p-1} \|G\|_1 \|x\|_p^p.
\end{aligned}$$

It follows that $\|G\|_p \leq \|G\|_1^{1/p} \|G\|_\infty^{1-1/p}$. □

Using Lemma 6.3.1 and we have

$$\|G\|_p \leq \|G\|_1^{1/p} \|G\|_\infty^{1-1/p} \leq 1 - \mathcal{K}_{S,\min}(1 - \gamma^n) = \beta.$$

Therefore, we have $\|\bar{F}(V_1) - \bar{F}(V_2)\|_2 \leq \beta \|V_1 - V_2\|_2$. Hence the operator $\bar{F}(\cdot)$ is a contraction mapping with respect to $\|\cdot\|_2$, with contraction factor $\beta$.

(3) (c) The result follows by observing that $\bar{F}(V^\pi) = V^\pi$ and $\bar{F}(\cdot)$ being a contraction mapping (hence has a unique fixed point).

### 6.3.2 Proof of Theorem 6.2.1

We will apply Theorem 2.5.1 and Corollary 2.5.1 (1) to the $n$-step TD algorithm. We begin by identifying the constants:

$$
\begin{aligned}
&A = A_1 + A_2 + 1 = 4,\ B = B_1 + B_2 = \frac{1}{1-\gamma},\ \varphi_1 \leq 1,\ \varphi_2 \geq 1-\beta,\ \varphi_3 \leq 228 \\
&c_1 \leq (\|V_0 - V^\pi\|_2 + \|V_0\|_2 + 4)^2,\ c_2 = \frac{1}{(1-\gamma)^2}(4(1-\gamma)\|V^\pi\|_2 + 1)^2.
\end{aligned}
$$

Now apply Theorem 2.5.1 (2) (a). When $\alpha_k = \alpha$ for all $k \geq 0$, where $\alpha$ is chosen such that

$$
\alpha(t_\alpha + n) \leq \frac{\varphi_2}{\varphi_3 A^2} = \frac{1-\beta}{3648},
$$

we have for all $k \geq t_\alpha + n$:

$$
\begin{aligned}
\mathbb{E}[\|V_k - V^\pi\|_2^2] &\leq \varphi_1 c_1 (1-\varphi_2\alpha)^{k-(\alpha(t_\alpha+n))} + \frac{\varphi_3 c_2}{\varphi_2}\alpha t_\alpha(\mathcal{M}_Y) \\
&\leq (\|V_0 - V^\pi\|_2 + \|V_0\|_2 + 4)^2 (1-(1-\beta)\alpha)^{k-(\alpha(t_\alpha+n))} \\
&\quad + \frac{228}{1-\beta}\frac{1}{(1-\gamma)^2}(4(1-\gamma)\|V^\pi\|_2 + 1)^2 \alpha(t_\alpha + n) \\
&= \hat{c}_1 (1-(1-\beta)\alpha)^{k-(\alpha(t_\alpha+n))} + \hat{c}_2 \frac{\alpha(t_\alpha+n)}{(1-\beta)(1-\gamma)^2},
\end{aligned}
$$

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

## 6.4 Finite-Sample Analysis of TD($\lambda$)

In this section, we consider the on-policy TD($\lambda$) algorithm, which effectively uses a convex combination of all the multi-step temporal differences at each update. We begin by describing the TD($\lambda$) algorithm for estimating the value function $V^\pi$ of a policy $\pi$.

The sequence $\{z_k\}$ is called the eligibility trace [11, 1], which according to line 3 of Algorithm 4 can be expressed as $z_k(s) = \sum_{i=0}^{k}(\gamma\lambda)^{k-i}\mathbb{1}_{\{S_i=s\}}$ for all $s \in \mathcal{S}$.

**Algorithm 4** The TD($\lambda$) Algorithm

```
**Input:** Integer $K$, initialization $V_0 \in \mathbb{R}^{|\mathcal{S}|}$, $z_{-1} = \mathbf{0}$, and a trajectory of samples $\{(S_k, A_k)\}_{0\leq k\leq K-1}$ collected under the target policy $\pi$
**for** $k = 0, 1, \cdots, K-1$ **do**
    $z_k(s) = (\gamma\lambda)z_{k-1}(s) + \mathbb{1}_{\{S_k=s\}}$ for all $s \in \mathcal{S}$
    $V_{k+1}(S_k) = V_k(S_k) + \alpha_k z_k(s)(\mathcal{R}(S_k, A_k) + \gamma V_k(S_{k+1}) - V_k(S_k))$
**end for**
**Output:** $V_K$
```

A key idea in the TD($\lambda$) algorithm is to use the parameter $\lambda$ to adjust the bootstrapping effect. When $\lambda = 0$, Algorithm 4 becomes the standard TD($0$) update (or 1-step TD), which is pure bootstrapping. Another extreme case is when $\lambda = 1$. This corresponds to using pure Monte Carlo method. Theoretical understanding of the efficiency of bootstrapping is a core problem in RL [18].

In the following subsection, we establish finite-sample convergence bounds of the TD($\lambda$) algorithm. By evaluating the resulting bound as a function of $\lambda$, we provide theoretical insights into the bias-variance trade-off in choosing $\lambda$. Similar to $n$-step TD, we make the following assumption.

**Assumption 6.4.1.** The Markov chain $\{S_k\}$ induced by the target policy $\pi$ is irreducible and aperiodic.

As a result of Assumption 6.4.1, the Markov chain $\{S_k\}$ has a unique stationary distribution, denoted by $\kappa_S \in \Delta^{|\mathcal{S}|}$, and the geometric mixing property [48].

### 6.4.1 Properties of the TD($\lambda$) Algorithm

Unlike the $n$-step TD-learning algorithm, the TD($\lambda$) algorithm cannot be viewed as a direct variant of the SA algorithm presented in Chapter 2. This is because of the geometric averaging induced by the eligibility trace in TD($\lambda$), which creates dependencies over the *entire* past trajectory. We overcome this difficulty by using an additional truncation argument, and separately handle the residual error due to truncation. For ease of exposition, we consider only using constant stepsize in the TD($\lambda$) algorithm, i.e., $\alpha_k = \alpha$ for all $k \geq 0$.

For any $k \geq 0$, let $Y_k = (S_0, ..., S_k, A_k, S_{k+1})$ (which takes value in $\mathcal{Y}_k := \mathcal{S}^{k+2} \times \mathcal{A}$), and define a time-varying operator $F_k : \mathbb{R}^{|\mathcal{S}|} \times \mathcal{Y}_k \mapsto \mathbb{R}^{|\mathcal{S}|}$ by

$$\begin{aligned}[F_k(V, y)](s) &= [F_k(V, s_0, ..., s_k, a_k, s_{k+1})](s) \\ &= (\mathcal{R}(s_k, a_k) + \gamma V_k(s_{k+1}) - V_k(s_k)) \sum_{i=0}^{k} (\gamma\lambda)^{k-i} \mathbb{1}_{\{s_i = s\}} + V(s)\end{aligned}$$

for all $s \in \mathcal{S}$. Note that the sequence $\{Y_k\}$ is *not* a Markov chain since it has a time-varying state-space. Using the notation of $\{Y_k\}$ and $F_k(\cdot, \cdot)$, we can rewrite the update equation of the TD($\lambda$) algorithm by

$$V_{k+1} = V_k + \alpha \left( F_k(V_k, Y_k) - V_k \right). \tag{6.3}$$

Although Equation 6.3 is similar to the update equation of the contractive Markovian SA algorithm presented in Chapter 2, since the sequence $\{Y_k\}$ is not a Markov chain and the operator $F_k(\cdot, \cdot)$ is time-varying, Theorem 2.5.1 is not directly applicable.

To overcome this difficulty, let us take a careful look at the operator $F_k(\cdot, \cdot)$. Although $F_k(V_k, Y_k)$ depends on the whole trajectory of states visited before (through the eligibility trace $z_k(s) = \sum_{i=0}^{k} (\gamma\lambda)^{k-i} \mathbb{1}_{\{S_i = s\}}$), due to the geometric factor $(\gamma\lambda)^{k-i}$, the states visited during the early stage of the iteration are not important. Inspired by this observation, we define the truncated sequence $\{Y_k^\tau\}$ of $\{Y_k\}$ by $Y_k^\tau = (S_{k-\tau}, ..., S_k, A_k, S_{k+1})$ for all $k \geq \tau$, where $\tau$ is a *fixed* non-negative integer. Note that the random process $\{Y_k^\tau\}$ is now a Markov chain, whose state-space is denoted by $\mathcal{Y}_\tau$ and is finite. Similarly, we define the truncated operator $F_k^\tau : \mathbb{R}^{|\mathcal{S}|} \times \mathcal{Y}_\tau \mapsto \mathbb{R}^{|\mathcal{S}|}$ of $F_k(\cdot, \cdot)$ by

$$\begin{aligned}[F_k^\tau(V, y^\tau)](s) &= [F_k^\tau(V, s_{k-\tau}, \cdots, s_k, a_k, s_{k+1})](s) \\ &= (\mathcal{R}(s_k, a_k) + \gamma V_k(s_{k+1}) - V_k(s_k)) \sum_{i=k-\tau}^{k} (\gamma\lambda)^{k-i} \mathbb{1}_{\{s_i = s\}} + V(s)\end{aligned}$$

for all $s \in \mathcal{S}$. Using the above notation, we can further rewrite the update equation of Algorithm 4 by

$$V_{k+1} = V_k + \alpha \left( F_k^\tau(V_k, Y_k^\tau) - V_k \right) + \underbrace{\alpha \left( F_k(V_k, Y_k) - F_k^\tau(V_k, Y_k^\tau) \right)}_{\text{The Error Term}}. \tag{6.4}$$

Now, we argue that when the truncation level $\tau$ is large enough, the last term on the RHS of the previous equation is negligible compared to the other two terms. In fact, we have the following result.

**Lemma 6.4.1.** *For all $k \geq 0$ and $\tau \in [0, k]$, denote $y = (s_0, ..., s_k, a_k, s_{k+1})$ and $y_\tau = (s_{k-\tau}, ..., s_k, a_k, s_{k+1})$. Then the following inequality holds for all $V \in \mathbb{R}^{|\mathcal{S}|}$:*

$$\|F_k^\tau(V, y_\tau) - F_k(V, y)\|_2 \leq \frac{(\gamma\lambda)^{\tau+1}}{1 - \gamma\lambda}(1 + 2\|V\|_2).$$

Lemma 6.4.1 indicates that the error term in Equation 6.4 is indeed geometrically small. Suppose we ignore that error term. Then the update equation becomes $V_{k+1} \approx V_k + \alpha_k(F_k^\tau(V_k, Y_k^\tau) - V_k)$. Since the random process $\mathcal{M}_Y = \{Y_k^\tau\}$ is a Markov chain, once we establish the required properties for the truncated operator $F_k^\tau(\cdot, \cdot)$, our SA results in Chapter 2 become applicable.

From now on, we will choose $\tau = \min\{k \geq 0 : (\gamma\lambda)^{k+1} \leq \alpha\} \leq \frac{\log(1/\alpha)}{\log(1/(\gamma\lambda))}$, where $\alpha$ is the constant stepsize we use. This implies that the error term in Equation 6.4 is of the size $\mathcal{O}(\alpha^2)$. Under this choice of $\tau$, we next investigate the properties of the operator $F_k^\tau(\cdot, \cdot)$ and the random process $\{Y_k^\tau\}$ in the following proposition. Recall that $\mathcal{K}_S \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ is a diagonal matrix with diagonal entries $\{\kappa_S(s)\}_{s \in \mathcal{S}}$, and $\mathcal{K}_{S,\min} = \min_{s \in \mathcal{S}} \kappa_S(s)$.

**Proposition 6.4.1.** *Suppose that Assumption 6.2.1 is satisfied. Then we have the following results.*

*(1) For any $k \geq \tau$, the operator $F_k^\tau(\cdot, \cdot)$ satisfies*

*(a) $\|F_k^\tau(V_1, y) - F_k^\tau(V_2, y)\|_2 \leq \frac{3}{1-\gamma\lambda}\|V_1 - V_2\|_2$ for any $V_1, V_2 \in \mathbb{R}^{|\mathcal{S}|}$ and $y \in \mathcal{Y}_\tau$,*

*(b)* $\|F_k^\tau(\mathbf{0},y)\|_2 \leq \frac{1}{1-\gamma\lambda}$ *for any* $y \in \mathcal{Y}_\tau$.

*(2) The Markov chain* $\{Y_k^\tau\}_{k\geq\tau}$ *has a unique stationary distribution, denoted by* $\mu_Y$*. Moreover, there exist* $C > 0$ *and* $\sigma \in (0,1)$ *such that*

$$\max_{y\in\mathcal{Y}_\tau} \|P_\pi^{k+\tau+1}(y,\cdot) - \mu_Y(\cdot)\|_{TV} \leq C\sigma^k, \ \forall\ k \geq 0.$$

*(3) For any* $k \geq \tau$*, define the expected operator* $\bar{F}_k^\tau : \mathbb{R}^{|\mathcal{S}|} \mapsto \mathbb{R}^{|\mathcal{S}|}$ *by* $\bar{F}_k^\tau(V) = \mathbb{E}_{Y\sim\mu_Y}[F_k^\tau(V,Y)]$*. Then*

*(a)* $\bar{F}_k^\tau(\cdot)$ *is explicitly given by*

$$\bar{F}_k^\tau(V) = \left(I - \mathcal{K}_S\sum_{i=0}^{\tau}(\gamma\lambda P_\pi)^i(I-\gamma P_\pi)\right)V + \mathcal{K}_S\sum_{i=0}^{\tau}(\gamma\lambda P_\pi)^i R_\pi.$$

*(b)* $\bar{F}_k^\tau(\cdot)$ *is a contraction mapping with respect to* $\|\cdot\|_p$ *for any* $p \in [1,\infty]$*, with a common contraction factor*

$$\beta = 1 - \mathcal{K}_{S,\min}\frac{(1-\gamma)(1-(\gamma\lambda)^{\tau+1})}{1-\gamma\lambda}.$$

*(c)* $\bar{F}_k^\tau(\cdot)$ *has a unique fixed-point* $V^\pi$*.*

Similar to $n$-step TD, the truncated asynchronous Bellman operator $\bar{F}_k^\tau(\cdot)$ associated with the TD($\lambda$) algorithm is a contraction with respect to the $\ell_p$-norm $\|\cdot\|_p$ for any $1 \leq p \leq \infty$, with a common contraction factor $\beta$. This enables us to use Theorem 2.5.1 along with Corollary 2.5.1 (1).

### 6.4.2 Finite-Sample Bounds of TD($\lambda$)

We now present the finite-sample convergence bounds of the TD($\lambda$) algorithm for using constant stepsize, where we exploit only the $\|\cdot\|_2$-contraction property from Proposition 6.4.1.

**Theorem 6.4.1.** *Consider $\{V_k\}$ of Algorithm 4. Suppose that Assumption 6.4.1 is satisfied and $\alpha_k \equiv \alpha$ with $\alpha$ chosen such that $\alpha(t_\alpha + 2\tau + 1) \leq \tilde{c}_0(1-\beta)(1-\gamma\lambda)^2$ (where $\tilde{c}_0$ is a numerical constant). Then the following inequality holds for all $k \geq t_\alpha + 2\tau + 1$:*

$$\mathbb{E}[\|V_k - V^\pi\|_2^2] \leq \tilde{c}_1 \left(1 - (1-\beta)\alpha\right)^{k-(t_\alpha+2\tau+1)} + \tilde{c}_2 \frac{\alpha\,(t_\alpha + \tau + 1)}{(1-\gamma\lambda)^2(1-\beta)},$$

*where $\tilde{c}_1 = (\|V_0 - V^\pi\|_2 + \|V_0\|_2 + 1)^2$ and $\tilde{c}_2 = 114(4\|V^\pi\|_2 + 1)^2$.*

*Remark.* Under Assumption 6.4.1, the mixing time $t_\alpha$ is at most an affine function of $\log(1/\alpha)$, and does not depend on the parameter $\lambda$.

The convergence rate of TD$(\lambda)$ is similar to that of $n$-step TD. We here focus on the impact of the parameter $\lambda$. We begin by rewriting both the bias term and the variance term in the resulting convergence bound of Theorem 6.4.1 focusing only on $\lambda$-dependent terms. Then, the bias term is of the size $(1 - \Theta(1/(1-\gamma\lambda)))^k$ while the variance term is between $\Theta(1/(1-\gamma\lambda)\log(1/(\gamma\lambda)))$ and $\Theta(1/(1-\gamma\lambda))$. Now observe that the bias term is in favor of large $\lambda$ (i.e., less bootstrapping, more Monte Carlo) while the variance term is in favor of small $\lambda$ (i.e., more bootstrapping, less Monte Carlo). This observation agrees with empirical results in the literature [1, 101]. Therefore, we demonstrate a bias-variance trade-off in choosing $\lambda$, thereby providing theoretical insights into the open problem of the efficiency of bootstrapping in RL [18].

### 6.4.3 Related Literature on TD$(\lambda)$

The idea of using $\lambda$-return and eligibility traces was introduced and developed in [99, 29]. See [1, Chapter 12] for more details. The convergence of TD$(\lambda)$ was established in [102].

Regarding the parameter $\lambda$, empirical observations indicate that a properly chosen intermediate value of $\lambda$ usually outperforms both TD$(0)$ and TD$(1)$ [100]. Theoretical justification of this observation is, to some extend, provided in [101], where they study a variant of the TD$(\lambda)$ algorithm called phased TD. The TD$(\lambda)$ algorithm is often used along with function approximation in practice. The asymptotic convergence of TD($\lambda$) with linear function approximation was established in [92]. More recently, [40, 12] established the finite-sample bounds of TD($\lambda$) with linear function approximation by modeling the algorithm as a linear stochastic approximation with Markovian noise. The result of [40] indicates that TD($\lambda$) in general outperforms TD($0$). However, [40] does not provide explicit trade-offs between the convergence bias and variance in choosing $\lambda$. Similarly, [12] does not have an explicit bound, and thus do not study bias-variance trade-off, which is what we did in this paper. To achieve that, we need to carefully characterize the contraction factor $\beta$ of the truncated Bellman operator $\bar{F}_k^\tau(\cdot)$, as well as the mixing time of the truncated Markov chain $\{Y_k^\tau\}$.

## 6.5 Proof of All Theoretical Results in Section 6.4

### 6.5.1 Proof of Lemma 6.4.1

The following lemma is useful when proving Lemma 6.4.1 and Proposition 6.4.1.

**Lemma 6.5.1.** *Let $\mathcal{I}$ be a finite set. For any $k \geq 0$, define two sequences $\{i_t\}_{0\leq t\leq k}$ and $\{a_t\}_{0\leq t\leq k}$ be such that $i_t \in \mathcal{I}$ and $a_t \geq 0$ for all $t = 0, 1, ..., k$. Let $x \in \mathbb{R}^{|\mathcal{I}|}$ be defined by $x_i = \sum_{t=0}^k a_t \mathbb{1}_{\{i_t=i\}}$ for all $i \in \mathcal{I}$. Then we have*

$$\|x\|_2 \leq \sum_{t=0}^k a_t.$$

*Proof of Lemma 6.5.1.* Using the definition of $\|\cdot\|_2$, we have

$$\begin{aligned}
\|x\|_2^2 &= \sum_{i\in\mathcal{I}} \left( \sum_{t=0}^k a_t \mathbb{1}_{\{i_t=i\}} \right)^2 \\
&= \sum_{i\in\mathcal{I}} \sum_{t=0}^k \sum_{\ell=0}^k a_t a_\ell \mathbb{1}_{\{i_t=i, i_\ell=i\}} \\
&= \sum_{t=0}^k \sum_{\ell=0}^k a_t a_\ell \sum_{i\in\mathcal{I}} \mathbb{1}_{\{i_t=i, i_\ell=i\}}
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{t=0}^{k}\sum_{\ell=0}^{k} a_t a_\ell \\
&= \left(\sum_{t=0}^{k} a_t\right)^2 .
\end{aligned}
$$

The result follows by taking square root on both sides of the previous inequality. □

We now proceed to prove Lemma 6.4.1. For any $V \in \mathbb{R}^{|\mathcal{S}|}$ and $(s_0, ..., s_k, a_k, s_{k+1})$, we have by definition of the operators $F_k^\tau(\cdot,\cdot)$ and $F_k(\cdot,\cdot)$ that

$$
\begin{aligned}
&\|F_k^\tau(V, s_{k-\tau}, ..., s_k, a_k, s_{k+1}) - F_k(V, s_0, ..., s_k, a_k, s_{k+1})\|_2^2 \\
&= \sum_{s\in\mathcal{S}} \left[(\mathcal{R}(s_k, a_k) + \gamma V(s_{k+1}) - V(s_k)) \sum_{i=0}^{k-\tau-1} (\gamma\lambda)^{k-i} \mathbb{1}_{\{s_i = s\}}\right]^2 \\
&\leq (1 + 2\|V\|_2)^2 \sum_{s\in\mathcal{S}} \left[\sum_{i=0}^{k-\tau-1} (\gamma\lambda)^{k-i} \mathbb{1}_{\{s_i = s\}}\right]^2 \\
&= \frac{(\gamma\lambda)^{2(\tau+1)}}{(1-\gamma\lambda)^2}(1 + 2\|V\|_2)^2. && \text{(Lemma 6.5.1)}
\end{aligned}
$$

The result follows by taking the square root on both sides of the previous inequality.

### 6.5.2 Proof of Proposition 6.4.1

(1) (a) For any $V_1, V_2 \in \mathbb{R}^{|\mathcal{S}|}$ and $y \in \mathcal{Y}_\tau$, we have by triangle inequality that

$$
\begin{aligned}
&\|F_k^\tau(V_1, y) - F_k^\tau(V_2, y)\|_2 \\
&\leq \|V_1 - V_2\|_2 \\
&\quad + \left(\sum_{s\in\mathcal{S}} \left[(\gamma(V_1(s_{k+1}) - V_2(s_{k+1})) - (V_1(s_k) - V_2(s_k))) \sum_{i=k-\tau}^{k} (\gamma\lambda)^{k-i} \mathbb{1}_{\{s_i = s\}}\right]^2\right)^{1/2} \\
&\leq \|V_1 - V_2\|_2 + 2\|V_1 - V_2\|_2 \left(\sum_{s\in\mathcal{S}} \left[\sum_{i=k-\tau}^{k} (\gamma\lambda)^{k-i} \mathbb{1}_{\{s_i = s\}}\right]^2\right)^{1/2} \\
&\leq \|V_1 - V_2\|_2 + \frac{2}{1-\gamma\lambda}\|V_1 - V_2\|_2 && \text{(Lemma 6.5.1)}
\end{aligned}
$$

$$\leq \frac{3}{1-\gamma\lambda}\|V_1 - V_2\|_2.$$

(1) (b) Similarly, for any $y \in \mathcal{Y}_\tau$, we have

$$\begin{aligned}
\|F_k^\tau(\mathbf{0}, y)\|_2^2 &= \sum_{s\in\mathcal{S}}\left[\mathcal{R}(s_k, a_k)\sum_{i=k-\tau}^{k}(\gamma\lambda)^{k-i}\mathbb{1}_{\{s_i=s\}}\right]^2 \\
&\leq \sum_{s\in\mathcal{S}}\left[\sum_{i=k-\tau}^{k}(\gamma\lambda)^{k-i}\mathbb{1}_{\{s_i=s\}}\right]^2 && (\mathcal{R}(s,a)\in[0,1] \text{ for all } (s,a)) \\
&\leq \frac{1}{(1-\gamma\lambda)^2}. && \text{(Lemma 6.5.1)}
\end{aligned}$$

It follows that $\|F_k^\tau(\mathbf{0}, y)\|_2 \leq \frac{1}{1-\gamma\lambda}$.

(2) The proof is identical to that of Proposition 6.2.1 (2).

(3) (a) For any $V \in \mathbb{R}^{|\mathcal{S}|}$ and $s \in \mathcal{S}$, we have

$$\begin{aligned}
&\mathbb{E}_{Y\sim\mu_Y}\left[[F_k^\tau(V, Y)](s)\right] \\
&= \mathbb{E}_{Y\sim\mu_Y}\left[(\mathcal{R}(S_k, A_k) + \gamma V(S_{k+1}) - V(S_k))\sum_{i=k-\tau}^{k}(\gamma\lambda)^{k-i}\mathbb{1}_{\{S_i=s\}}\right] + V(s) \\
&= \mathbb{E}_{Y\sim\mu_Y}\left[\sum_{i=k-\tau}^{k}(\gamma\lambda)^{k-i}\mathbb{1}_{\{S_i=s\}}\mathbb{E}\left[(\mathcal{R}(S_k, A_k) + \gamma V(S_{k+1}) - V(S_k)) \mid S_k, S_{k-1}, ..., S_0\right]\right] \\
&\quad + V(s) \\
&= \mathbb{E}_{Y\sim\mu_Y}\left[\sum_{i=k-\tau}^{k}(\gamma\lambda)^{k-i}\mathbb{1}_{\{S_i=s\}}(R_\pi(S_k) + \gamma[P_\pi V](S_k) - V(S_k))\right] + V(s) \\
&= \sum_{i=k-\tau}^{k}(\gamma\lambda)^{k-i}\sum_{s_0\in\mathcal{S}}\kappa_S(s_0)P_\pi^i(s_0, s)\sum_{s'\in\mathcal{S}}P_\pi^{k-i}(s, s')(R_\pi(s') + \gamma[P_\pi V](s') - V(s')) \\
&\quad + V(s) \\
&= \kappa_S(s)\sum_{i=k-\tau}^{k}(\gamma\lambda)^{k-i}\sum_{s'\in\mathcal{S}}P_\pi^{k-i}(s, s')(R_\pi(s') + \gamma[P_\pi V](s') - V(s')) + V(s)
\end{aligned}$$

$$= \kappa_S(s) \sum_{i=k-\tau}^{k} (\gamma\lambda)^{k-i}[P_\pi^{k-i}(R_\pi + \gamma P_\pi V - V)](s) + V(s).$$

It follows that

$$\begin{aligned}
\bar{F}_k^\tau(V) &= \mathcal{K}_S \sum_{i=k-\tau}^{k} (\gamma\lambda P_\pi)^{k-i}(R_\pi + \gamma P_\pi V - V) + V \\
&= \mathcal{K}_S \sum_{i=0}^{\tau} (\gamma\lambda P_\pi)^{i}(R_\pi + \gamma P_\pi V - V) + V \\
&= \left[I - \mathcal{K}_S \sum_{i=0}^{\tau} (\gamma\lambda P_\pi)^{i}(I - \gamma P_\pi)\right] V + \mathcal{K}_S \sum_{i=0}^{\tau} (\gamma\lambda P_\pi)^{i} R_\pi.
\end{aligned}$$

(3) (b) For any $V_1, V_2 \in \mathbb{R}^{|\mathcal{S}|}$ and $p \in [1, \infty]$, we have

$$\begin{aligned}
\|\bar{F}_k^\tau(V_1) - \bar{F}_k^\tau(V_2)\|_p &= \left\| \left[I - \mathcal{K}_S \sum_{i=0}^{\tau} (\gamma\lambda P_\pi)^{i}(I - \gamma P_\pi)\right] (V_1 - V_2) \right\|_p \\
&\leq \left\| I - \mathcal{K}_S \sum_{i=0}^{\tau} (\gamma\lambda P_\pi)^{i}(I - \gamma P_\pi) \right\|_p \|V_1 - V_2\|_p.
\end{aligned}$$

Denote $G_{\lambda,\tau} = I - \mathcal{K}_S \sum_{i=0}^{\tau} (\gamma\lambda P_\pi)^i (I - \gamma P_\pi)$. It remains to provide an upper bound on $\|G_{\lambda,\tau}\|_p$. Since

$$\begin{aligned}
G_{\lambda,\tau} &= I - \mathcal{K}_S \sum_{i=0}^{\tau} (\gamma\lambda P_\pi)^{i} + \mathcal{K}_S \sum_{i=0}^{\tau} (\gamma\lambda P_\pi)^{i} \gamma P_\pi \\
&= I - \mathcal{K}_S - \mathcal{K}_S \sum_{i=1}^{\tau} (\gamma\lambda P_\pi)^{i} + \mathcal{K}_S \sum_{i=0}^{\tau} (\gamma\lambda P_\pi)^{i} \gamma P_\pi \\
&= I - \mathcal{K}_S - \mathcal{K}_S \sum_{i=0}^{\tau-1} (\gamma\lambda P_\pi)^{i+1} + \mathcal{K}_S \sum_{i=0}^{\tau} (\gamma\lambda P_\pi)^{i} \gamma P_\pi \\
&= I - \mathcal{K}_S + \mathcal{K}_S \sum_{i=0}^{\tau-1} (\gamma\lambda P_\pi)^{i} \gamma P_\pi (1 - \lambda) + \mathcal{K}_S (\gamma\lambda P_\pi)^{\tau} \gamma P_\pi,
\end{aligned}$$

the matrix $G_{\lambda,\tau}$ has non-negative entries. Therefore, we have

$$\|G_{\lambda,\tau}\|_\infty = \|G_{\lambda,\tau}\mathbf{1}\|_\infty$$

$$
\begin{aligned}
&= \left\| \mathbf{1} - \kappa_S \frac{(1-\gamma)(1-(\gamma\lambda)^{\tau+1})}{1-\gamma\lambda} \right\|_\infty \\
&= 1 - \mathcal{K}_{S,\min} \frac{(1-\gamma)(1-(\gamma\lambda)^{\tau+1})}{1-\gamma\lambda}
\end{aligned}
$$

and

$$
\begin{aligned}
\|G_{\lambda,\tau}\|_1 &= \|\mathbf{1}^\top G_{\lambda,\tau}\|_\infty \\
&= \left\| \mathbf{1}^\top - \kappa_S^\top \frac{(1-\gamma)(1-(\gamma\lambda)^{\tau+1})}{1-\gamma\lambda} \right\|_\infty \\
&= 1 - \mathcal{K}_{S,\min} \frac{(1-\gamma)(1-(\gamma\lambda)^{\tau+1})}{1-\gamma\lambda}.
\end{aligned}
$$

It then follows from Lemma 6.3.1 that

$$
\|G_{\lambda,\tau}\|_p \leq \|G_{\lambda,\tau}\|_1^{1/p} \|G_{\lambda,\tau}\|_\infty^{1-1/p} \leq 1 - \mathcal{K}_{S,\min} \frac{(1-\gamma)(1-(\gamma\lambda)^{\tau+1})}{1-\gamma\lambda}.
$$

Hence the operator $F_k^\tau(\cdot,\cdot)$ is a contraction with respect to $\|\cdot\|_p$ for any $p \in [1,\infty]$, with a common contraction factor $\beta = 1 - \mathcal{K}_{S,\min} \frac{(1-\gamma)(1-(\gamma\lambda)^{\tau+1})}{1-\gamma\lambda}$.

(3) (c) It is enough to show that $V^\pi$ is a fixed-point of $\bar{F}_k^\tau(\cdot)$, the uniqueness follows from $\bar{F}_k^\tau(\cdot)$ being a contraction. Using the Bellman equation $R_\pi + \gamma P_\pi V^\pi - V^\pi = 0$, we have

$$
\bar{F}_k^\tau(V^\pi) = \mathcal{K}_S \sum_{i=0}^{\tau} (\gamma\lambda P_\pi)^i (R_\pi + \gamma P_\pi V^\pi - V^\pi) + V^\pi = V^\pi.
$$

### 6.5.3 Proof of Theorem 6.4.1

We will exploit the $\|\cdot\|_2$-contraction property of the operator $\bar{F}_k^\tau(\cdot)$ provided in Proposition 6.4.1. Let $M(x) = \|x\|_2^2$ be our Lyapunov function. Using the update equation Equation 6.4

and we have for all $k \geq 0$:

$$\begin{aligned}\|V_{k+1} - V^\pi\|_2^2 = \|V_k - V^\pi\|_2^2 + &\underbrace{2\alpha(V_k - V^\pi)^\top \left(\bar{F}_k^\tau(V_k) - V_k\right)}_{T_1'} \\ &+ \underbrace{2\alpha(V_k - V^\pi)^\top \left(F_k^\tau(V_k, Y_k^\tau) - \bar{F}_k^\tau(V_k)\right)}_{T_2'} + \underbrace{\alpha^2\|F_k^\tau(V_k, Y_k^\tau) - V_k\|_2^2}_{T_3'} \\ &+ \underbrace{\alpha^2\|F_k(V_k, Y_k) - F_k^\tau(V_k, Y_k^\tau)\|_2^2}_{T_4'} \\ &+ \underbrace{2\alpha(V_k - V^\pi)^\top \left(F_k(V_k, Y_k) - F_k^\tau(V_k, Y_k^\tau)\right)}_{T_5'} \\ &+ \underbrace{2\alpha\left(F_k^\tau(V_k, Y_k^\tau) - V_k\right)^\top \left(F_k(V_k, Y_k) - F_k^\tau(V_k, Y_k^\tau)\right)}_{T_6'}. \end{aligned} \tag{6.5}$$

The terms $T_1'$, $T_2'$, and $T_3'$ correspond to the terms $T_1$, $T_3$, and $T_4$ in Equation 2.6, and hence can be controlled in the exact same way as provided in Lemmas 2.6.1, 2.6.4, and 2.6.5. The upper bounds of $T_1'$, $T_2'$, and $T_3'$ are summarized in the following lemma, whose proof is omitted.

**Lemma 6.5.2.** *The following inequalities hold:*

*(1)* $T_1' \leq -2\alpha(1-\beta)\|V_k - V^\pi\|_2^2$ *for any* $k \geq \tau$.

*(2)* $\mathbb{E}[T_2'] \leq \frac{662\alpha^2(t_\alpha+\tau)}{(1-\gamma\lambda)^2}\|V_k - V^\pi\|_2^2 + \frac{102\alpha^2(t_\alpha+\tau)}{(1-\gamma\lambda)^2}(4\|V^\pi\|_2 + 1)^2$ *for all* $k \geq 2\tau + t_\alpha$.

*(3)* $T_3' \leq \frac{32\alpha^2}{(1-\gamma\lambda)^2}\|V_k - V^\pi\|_2^2 + \frac{2\alpha^2}{(1-\gamma\lambda)^2}(4\|V^\pi\|_2 + 1)^2$ *for all* $k \geq \tau$.

As for the terms $T_4'$, $T_5'$, and $T_6'$, we can easily use Lemma 6.5.3 along with the Cauchy-Schwarz inequality to bound them, which gives the following result.

**Lemma 6.5.3.** *The following inequalities hold:*

*(1)* $T_4' \leq \frac{8\alpha^2}{(1-\gamma\lambda)^2}\|V_k - V^\pi\|_2^2 + \frac{2\alpha^2}{(1-\gamma\lambda)^2}(4\|V^\pi\|_2 + 1)^2$ *for all* $k \geq \tau$.

*(2)* $T_5' \leq \frac{16\alpha^2}{(1-\gamma\lambda)}\|V_k - V^\pi\|_2^2 + \frac{4\alpha^2}{(1-\gamma\lambda)}(4\|V^\pi\|_2 + 1)^2$ *for all* $k \geq \tau$.

*(3)* $T_6' \leq \frac{64\alpha^2}{(1-\gamma\lambda)^2}\|V_k - V^\pi\|_2^2 + \frac{4\alpha^2}{(1-\gamma\lambda)^2}(4\|V^\pi\|_2+1)^2$ *for all* $k \geq \tau$.

*Proof of Lemma 6.5.3.* (1) For all $k \geq \tau$, we have

$$\begin{aligned}
T_4' &= \alpha^2\|F_k(V_k,Y_k) - F_k^\tau(V_k,Y_k^\tau)\|_2^2 \\
&\leq \frac{\alpha^2(\gamma\lambda)^{2(\tau+1)}}{(1-\gamma\lambda)^2}(2\|V_k\|_2+1)^2 && \text{(Lemma 6.4.1)} \\
&\leq \frac{\alpha^4}{(1-\gamma\lambda)^2}(2\|V_k - V^\pi\|_2 + 2\|V^\pi\|_2 + 1)^2 \\
&\leq \frac{8\alpha^2}{(1-\gamma\lambda)^2}\|V_k - V^\pi\|_2^2 + \frac{2\alpha^2}{(1-\gamma\lambda)^2}(4\|V^\pi\|_2+1)^2.
\end{aligned}$$

(2) For all $k \geq \tau$, we have

$$\begin{aligned}
T_5' &= 2\alpha(V_k - V^\pi)^\top\left(F_k(V_k,Y_k) - F_k^\tau(V_k,Y_k^\tau)\right) \\
&\leq 2\alpha\|V_k - V^\pi\|_2\|F_k(V_k,Y_k) - F_k^\tau(V_k,Y_k^\tau)\|_2 \\
&\leq \frac{2\alpha(\gamma\lambda)^{\tau+1}}{(1-\gamma\lambda)}\|V_k - V^\pi\|_2(2\|V_k\|_2+1) && \text{(Proposition 6.4.1 (1))} \\
&\leq \frac{2\alpha(\gamma\lambda)^{\tau+1}}{(1-\gamma\lambda)}(2\|V_k - V^\pi\|_2 + 2\|V^\pi\|_2 + 1)^2 \\
&\leq \frac{16\alpha(\gamma\lambda)^{\tau+1}}{(1-\gamma\lambda)}\|V_k - V^\pi\|_2^2 + \frac{4\alpha(\gamma\lambda)^{\tau+1}}{(1-\gamma\lambda)}(4\|V^\pi\|_2+1)^2 \\
&\leq \frac{16\alpha^2}{(1-\gamma\lambda)}\|V_k - V^\pi\|_2^2 + \frac{4\alpha^2}{(1-\gamma\lambda)}(4\|V^\pi\|_2+1)^2,. && \text{(The choice of } \tau\text{)}
\end{aligned}$$

(3) For all $k \geq \tau$, we have

$$\begin{aligned}
T_6' &= 2\alpha\left(F_k^\tau(V_k,Y_k^\tau) - V_k\right)^\top\left(F_k(V_k,Y_k) - F_k^\tau(V_k,Y_k^\tau)\right) \\
&\leq 2\alpha\|F_k^\tau(V_k,Y_k^\tau) - V_k\|_2\|F_k(V_k,Y_k) - F_k^\tau(V_k,Y_k^\tau)\|_2 \\
&\leq \frac{2\alpha(\gamma\lambda)^{\tau+1}}{1-\gamma\lambda}\left(\frac{3}{1-\gamma\lambda}\|V_k\|_2 + \frac{1}{1-\gamma\lambda} + \|V_k\|_2\right)(2\|V_k\|_2+1) \\
&\leq \frac{2\alpha(\gamma\lambda)^{\tau+1}}{(1-\gamma\lambda)^2}(4\|V_k\|_2+1)(2\|V_k\|_2+1) \\
&\leq \frac{2\alpha(\gamma\lambda)^{\tau+1}}{(1-\gamma\lambda)^2}(4\|V_k - V^\pi\|_2 + 4\|V^\pi\|_2 + 1)^2
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{64\alpha(\gamma\lambda)^{\tau+1}}{(1-\gamma\lambda)^2}\|V_k - V^\pi\|_2^2 + \frac{4\alpha(\gamma\lambda)^{\tau+1}}{(1-\gamma\lambda)^2}(4\|V^\pi\|_2+1)^2 \\
&\leq \frac{64\alpha^2}{(1-\gamma\lambda)^2}\|V_k - V^\pi\|_2^2 + \frac{4\alpha^2}{(1-\gamma\lambda)^2}(4\|V^\pi\|_2+1)^2. \qquad \text{(The choice of } \tau\text{)}
\end{aligned}
$$

□

The rest of the proof is to use the upper bounds we derived for the terms $T_1'$ to $T_6'$ in Equation 6.5 to obtain the one-step contractive inequality. Repeatedly using such one-step inequality and we get the finite-sample bounds stated in Theorem 6.4.1. This part is identical to the proof of Theorem 2.5.1.

## 6.6 Conclusion and Future Work

In this chapter, we present finite-sample guarantees of two popular families of TD-learning algorithms, namely the $n$-step TD and the TD$(\lambda)$. Moreover, the finite-sample guarantees shed light on the open problem about the efficiency of bootstrapping, which is about how to pick the parameters $n$ and $\lambda$ so that $n$-step TD and TD$(\lambda)$ achieve their best performance.

However, the bias-variance trade-off we demonstrate (or the estimated optimal choice of $n$ in $n$-step TD) may not be accurate since we only have upper bounds. To complete the story, in addition to upper bounds, we also need lower bounds (hopefully of the same order). Deriving lower bounds of $n$-step TD and TD$(\lambda)$ is a future direction of this line of work.

# CHAPTER 7
# OFF-POLICY PREDICTION: THE BIAS-VARIANCE TRADE-OFF

## 7.1 Introduction

In TD-learning, a key ingredient 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 [103, 104], and hence we need to work with historical data that is generated by a possibly different policy (i.e., off-policy sampling).

Off-policy learning is inevitable in high-stakes applications such as healthcare [105], education [106], robotics [107] and clinical trials [108, 104]. The agent there may not have direct access to the environment in order to perform online sampling, and one has to work with limited historical data that is collected under a fixed behavior policy. Moreover, off-policy sampling enables off-line learning by decoupling data collection from learning, and is observed to extract the maximum possible utility out of limited available data [109].

Although off-policy sampling is more practical than on-policy sampling, it is more challenging to analyze and is known to have high variance [110], 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 $Q^\pi(\lambda)$ [13], TB$(\lambda)$) [14], Retrace$(\lambda)$ [15], and $Q$-trace [16], etc.

### 7.1.1 Main Contributions

In this chapter, 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 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 $\ell_p$-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 $\epsilon$ 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$(\lambda)$, Retrace$(\lambda)$, and $Q$-trace. For $Q^\pi(\lambda)$, TB$(\lambda)$, and Retrace$(\lambda)$, we establish the first-known results in the literature, while for $Q$-trace, we improve the best known results in [16] in terms of the dependency on the size of the state-action space. The weighted $\ell_p$-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.

### 7.1.2 Generalized Bellman Operator and Stochastic Approximation

In this subsection, 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 of estimating the state-action value function $Q^\pi$ of a given policy $\pi$. 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 defined by $[\mathcal{H}^\pi(Q)](s,a) = \mathcal{R}(s,a) + \gamma\mathbb{E}[\max_{a' \in \mathcal{A}} Q(S_{k+1}, a') \mid S_k = s, A_k = a]$ for all $(s,a)$. The generalized Bellman operator $\mathcal{B}(\cdot)$ we consider in this paper is defined by:

$$\mathcal{B}(Q) = \mathcal{T}(\mathcal{H}(Q) - Q) + Q, \tag{7.1}$$

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($\lambda$) 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, $\gamma$ is the discount factor, and $P_\pi$ is the transition probability matrix under policy $\pi$. 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($\lambda$) 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 $\ell_p$-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.

### 7.1.3 Related Literature

The TD-learning method was first proposed in [56] 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 [56]. Later it was extended to using multi-step bootstrapping (i.e., the $n$-step TD-learning algorithm [99, 111, 112]), and using eligibility trace (i.e., the TD$(\lambda)$ algorithm [56, 113]). The asymptotic convergence of TD-learning was established in [24, 29, 114]. As for finite-sample analysis, a unified Lyapunov approach is presented in [115]. 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 [92], and finite-sample bounds in [40, 12, 116, 41]. Very recently, the convergence and finite-sample guarantee of TD-learning with neural network approximation were studied in [117, 118].

*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 $Q^\pi(\lambda)$ [14], TB$(\lambda)$ [13], Retrace$(\lambda)$ [15], and $Q$-trace [16] (which is an extension of $V$-trace [25]), 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 [16, 119, 115]. In the function approximation setting, TD-learning with off-policy sampling and function approximation is a typical example of the deadly triad [1], and can be unstable [1, 8]. To achieve convergence, one needs to significantly modify the original TD-learning algorithm, resulting in two time-scale algorithms such as GTD [120], TDC [121], and emphatic TD [122], etc.

## 7.2 Finite-Sample Analysis

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.

### 7.2.1 A Generic Model for Multi-Step Off-Policy TD-Learning

Algorithm 5 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 5 to serve as generalized importance sampling ratios in order to account for the discrepancy between the target policy $\pi$ and the behavior policy $\pi_b$. We denote $c_{\max} = \max_{s,a} c(s,a)$ and $\rho_{\max} = \max_{s,a} \rho(s,a)$. We next show how Algorithm 5 captures variants of off-policy TD-learning algorithms in the literature by using different generalized importance sampling ratios $c(\cdot,\cdot)$ and $\rho(\cdot,\cdot)$.

**Algorithm 5** A Generic Algorithm for Multi-Step Off-Policy TD-Learning

```
Input: $K$, $\{\alpha_k\}$, $Q_0$, $\pi$, $\pi_b$, 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 $\pi_b$.
for $k = 0, 1, \cdots, K-1$ do
    $\alpha_k(s,a) = \alpha_k \mathbb{I}\{(s,a) = (S_k, A_k)\}$ for all $(s,a)$
    $\Delta(S_i, A_i, S_{i+1}, A_{i+1}, Q_k) = \mathcal{R}(S_i, A_i) + \gamma\rho(S_{i+1}, A_{i+1}) Q_k(S_{i+1}, A_{i+1}) - Q_k(S_i, A_i)$
    for all $i \in \{k, k+1, ..., k+n-1\}$.
    $Q_{k+1}(s,a) = Q_k(s,a) + \alpha_k(s,a) \sum_{i=k}^{k+n-1} \gamma^{i-k} \prod_{j=k+1}^{i} c(S_j, A_j) \Delta(S_i, A_i, S_{i+1}, A_{i+1}, Q_k)$
    for all $(s,a)$
end for
Output: $Q_K$
```

**Vanilla IS.** When $c(s,a) = \rho(s,a) = \pi(a|s)/\pi_b(a|s)$ for all $(s,a)$, Algorithm 5 is the standard off-policy TD-learning with importance sampling [14]. We will refer to this algorithm as Vanilla IS. Although Vanilla IS was shown to converge to $Q^\pi$ [14], 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 5 is the $Q^\pi(\lambda)$ algorithm [13]. The $Q^\pi(\lambda)$ algorithm overcomes the high variance issue in Vanilla IS by introducing the parameter $\lambda$. However, the algorithm converges only when $\lambda$ is sufficiently small [15].

**The TB$(\lambda)$ 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 [14]. The TB$(\lambda)$ algorithm also overcomes the high variance issue in Vanilla IS and is guaranteed to converge to $Q^\pi$ without needing any strong assumptions. However, as discussed in [15], the TB$(\lambda)$ algorithm lacks sample efficiency as it does not effectively use the multi-step return.

**The Retrace$(\lambda)$ 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 $Q^\pi$. The convergence rate of Retrace$(\lambda)$ is empirically observed to be better than TB$(\lambda)$ in [15].

**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 5 is the $Q$-trace algorithm [16]. The $Q$-trace algorithm is an analog of the $V$-trace algorithm [25] 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 $Q^\pi$, but to a biased limit point, denoted by $Q^{\pi,\rho}$ [16].

From now on, we focus on studying Algorithm 5. We make the following assumption about the behavior policy $\pi_b$, which is fairly standard in off-policy TD-learning.

**Assumption 7.2.1.** The behavior policy $\pi_b$ satisfies $\pi_b(a|s) > 0$ for all $(s,a)$. In addition, the Markov chain $\{S_k\}$ induced by the behavior policy $\pi_b$ 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 $\max_{s\in\mathcal{S}} \|P_{\pi_b}^k(s, \cdot) - \kappa_S(\cdot)\|_{\text{TV}} \leq C\sigma^k$ for all $k \geq 0$, where $\|\cdot\|_{\text{TV}}$ is the total variation distance [48]. 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 $\pi_b$. 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.

### 7.2.2 Identifying the Generalized Bellman Operator

In this subsection, we identify the generalized Bellman equation which Algorithm 5 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 5 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 5 performs asynchronous update, we further define the asynchronous variant $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ of the generalized Bellman operator $\mathcal{B}_{c,\rho}(\cdot)$ by

$$\tilde{\mathcal{B}}_{c,\rho}(Q) := \mathcal{K}_{SA}\mathcal{B}_{c,\rho}(Q) + (I - \mathcal{K}_{SA})Q = \mathcal{K}_{SA}\mathcal{T}_c(\mathcal{H}_\rho(Q) - Q) + Q. \tag{7.2}$$

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).

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$(\lambda)$, and Retrace$(\lambda)$.

### 7.2.3 Establishing the Contraction Property

In this subsection, 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 $D_\rho$ respectively.

In view of the definition of $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ in Equation 7.2, 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 7.2.1.** *Suppose that $D_{\rho,\max} < 1/\gamma$. Then the operator $\mathcal{H}_\rho(\cdot)$ is a contraction* *mapping with respect to the $\ell_\infty$-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)* $\|Q^\pi - Q^{\pi,\rho}\|_\infty \leq \frac{\gamma \max_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} |\pi(a|s)-\pi_b(a|s)\rho(s,a)|}{(1-\gamma)(1-\gamma D_{\rho,\max})}$,

*(2)* $\|Q^{\pi,\rho}\|_\infty \leq \frac{1}{1-\gamma D_{\rho,\max}}$.

Observe from Proposition 7.2.1 (1) that when $\rho(s,a) = \pi(a|s)/\pi_b(a|s)$, which is the case for $Q^\pi(\lambda)$, TB($\lambda$), and Retrace($\lambda$), the unique fixed-point $Q^{\pi,\rho}$ is exactly the target value function $Q^\pi$. 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 $Q^\pi$. In that case, Proposition 7.2.1 provides an error bound on the difference between the potentially biased limit $Q^{\pi,\rho}$ and $Q^\pi$. Such error bound will be useful for us to study the $Q$-trace algorithm in Section 7.3. Proposition 7.2.1 (2) can be viewed as an analog to the inequality that $\|Q^\pi\|_\infty \leq 1/(1-\gamma)$ for any policy $\pi$. 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 7.2.1 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 7.2.1.** 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 $\ell_p$-norm ($p \in [1,\infty)$) of $x$ with weights $\{\mu_i\}$ is defined by $\|x\|_{\mu,p} = (\sum_i \mu_i |x_i|^p)^{1/p}$ for any $x \in \mathbb{R}^d$.

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 7.2.1.** *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 7.2.1 (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 7.2.1 (2) can be viewed as an extension of Theorem 7.2.1 (1) because $\lim_{p\to\infty}\|x\|_{\mu,p} = \|x\|_\infty$ for any $x \in \mathbb{R}^d$ and weight vector $\mu$, and $\lim_{p\to\infty}(1-\omega)^{1-1/p}(1-\theta\omega)^{1/p} = 1-\omega$.

Theorem 7.2.1 is the key result for our finite-sample analysis. The weighted $\ell_p$-norm (especially the weighted $\ell_2$-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 [92] is that the corresponding Bellman operator is a contraction mapping not only with respect to the $\ell_\infty$-norm, but also with respect to a weighted $\ell_2$-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.

### 7.2.4 Finite-Sample Convergence Guarantees

In light of Theorem 7.2.1, Algorithm 5 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 $\mu$ is unknown, to make the finite-sample bound independent of $\mu$, we use the lower bound on the components of $\mu$ provided in Theorem 7.2.1, and also tune the parameters $p$ and $\theta$ 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 7.2.1) plays an important role in such tuning process.

To present the results, we need to introduce more notation. For any $\delta > 0$, define $t_\delta$ as the mixing time of the Markov chain $\{S_k\}$ (induced by $\pi_b$) with precision $\delta$, i.e., $t_\delta = \min\{k \geq 0 : \max_{s\in\mathcal{S}} \|P_{\pi_b}^k(s,\cdot) - \kappa_S(\cdot)\|_{\text{TV}} \leq \delta\}$. Under Assumption 7.2.1, one can easily verify that $t_\delta \leq L(\log(1/\delta) + 1)$ for some constant $L > 0$, which depends only on $C$ and $\delta$. Let $\tau_{\delta,n} = t_\delta + n + 1$. The parameters $c_1, c_2$ and $c_3$ used in stating the following theorem are numerical constants, and will be explicitly given in Section 7.4 where we present its proof. For ease of exposition, we here only present the finite-sample bound for using constant stepsize.

**Theorem 7.2.2.** *Consider $\{Q_k\}$ of Algorithm 5. Suppose that (1) Assumptions 7.2.1 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 $\alpha$ 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 $k \geq \tau_{\alpha,n}$:*

$$\mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_\infty^2] \leq \zeta_1 \left(1 - \frac{\omega\alpha}{2}\right)^{k-\tau_{\alpha,n}} + \zeta_2 \frac{f(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2\log(2|\mathcal{S}||\mathcal{A}|/\omega)}{\omega}\alpha\tau_{\alpha,n}, \tag{7.3}$$

*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 7.2.2 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 Equation 7.3 is usually called the bias in SA literature [23], and it goes to zero at a geometric rate. The second term on the RHS of Equation 7.3 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 5 in the following.

**Corollary 7.2.1.** *For an accuracy $\epsilon > 0$, to obtain $\mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_\infty] \leq \epsilon$, the sample complexity is*

$$\underbrace{\mathcal{O}\left(\frac{\log^2(1/\epsilon)}{\epsilon^2}\right)}_{T_1}\underbrace{\tilde{\mathcal{O}}\left(\frac{1}{\omega^2}\right)}_{T_2}\underbrace{\tilde{\mathcal{O}}\left(\frac{nf(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2}{(1-\gamma D_{\rho,\max})^2}\right)}_{T_3}. \tag{7.4}$$

In Corollary 7.2.1, the $\tilde{\mathcal{O}}(\epsilon^{-2})$ dependence on the accuracy is the same as $n$-step TD-learning in the on-policy setting (cf. Chapter 6), and is in general not improvable. The term $T_2$ can be equivalently written as $\tilde{O}(1/(1-\text{Contraction factor})^2)$, 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 Equation 7.3. 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 $1/\log(1/\gamma)$ in Chapter 6. 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 5 is intuitively of the form $\tilde{\mathcal{O}}\left(\frac{\text{Variance}}{(1-\text{ Contraction factor})^2}\right)$.

## 7.3 Application to Various Off-Policy TD-Learning Algorithms

In this section, we apply Theorem 7.2.2 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 $\pi$ and the behavior policy $\pi_b$ 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 $\pi$ and $\pi_b$. We will overload the notation of $\zeta_1$ and $\zeta_2$ from Theorem 7.2.2. Note that $Q^{\pi,\rho} = Q^\pi$ in $Q^\pi(\lambda)$, TB($\lambda$), and Retrace($\lambda$), but $Q^{\pi,\rho} \neq Q^\pi$ in $Q$-trace.

### 7.3.1 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 7.3.1.** *Consider Algorithm 5 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 7.2.1 is satisfied. Then, to achieve $\epsilon$-accuracy, the sample complexity is*

$$O\left(\frac{\log^2(1/\epsilon)}{\epsilon^2}\right)\tilde{\mathcal{O}}\left(\frac{1}{\omega^2}\right)\tilde{\mathcal{O}}\left(\frac{n((\gamma r_{\max})^n+1)^2}{(1-\gamma)^2}\right).$$

In the special case where $\pi = \pi_b$ (i.e., on-policy $n$-step TD), the sample complexity bound reduces to $\tilde{\mathcal{O}}\left(\frac{n\log^2(1/\epsilon)}{\epsilon^2\mathcal{K}_{SA,\min}^2(1-\gamma^n)^2(1-\gamma)^2}\right)$, which is comparable to the results in Chapter 6. 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.

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

In this subsection, 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 7.3.2.** *Consider Algorithm 5 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 7.2.1 is satisfied, and $\lambda \leq r_{\min}$. Then, to achieve $\epsilon$-accuracy, the sample complexity is*

$$O\left(\frac{\log^2(1/\epsilon)}{\epsilon^2}\right)\tilde{\mathcal{O}}\left(\frac{1}{\omega^2}\right)\tilde{\mathcal{O}}\left(\frac{nf(\gamma\lambda)^2(\gamma r_{\max}+1)^2}{(1-\gamma)^2}\right).$$

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 $\lambda$, $Q^\pi(\lambda)$ algorithm successfully avoids an exponential large factor in the sample complexity. However, choosing a small $\lambda$ to control the variance has a side effect on the contraction factor. Intuitively, when $\lambda$ is small, $Q^\pi(\lambda)$ does not effectively use the $n$-step return. Hence the parameter $\omega$ in $Q^\pi(\lambda)$ is less (worse) than the one in Vanilla IS.

### 7.3.3 Finite-Sample Analysis of TB$(\lambda)$

In this subsection, we present the sample complexity of the TB$(\lambda)$ 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 7.3.3.** *Consider Algorithm 5 with TB$(\lambda)$ update. 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 7.2.1 is satisfied, and $\lambda \leq 1/\pi_{b,\max}$. Then, to achieve $\epsilon$-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($\lambda$) is that the constraint on $\lambda$ 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 $\omega$ in TB($\lambda$) is less (worse) than the one in Vanilla IS.

### 7.3.4 Finite-Sample Analysis of Retrace($\lambda$)

In this subsection, we present the sample complexity of the Retrace($\lambda$) 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 7.3.4.** *Consider Algorithm 5 with Retrace($\lambda$) update. 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 7.2.1 is satisfied, and $\lambda \leq 1$. Then, to achieve $\epsilon$-accuracy, the sample complexity is*

$$O\left(\frac{\log^2(1/\epsilon)}{\epsilon^2}\right)\tilde{O}\left(\frac{1}{\omega^2}\right)\tilde{O}\left(\frac{nf(\gamma\lambda)^2(\gamma r_{\max}+1)^2}{(1-\gamma)^2}\right).$$

The Retrace($\lambda$) 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 $\lambda$ 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($\lambda$) algorithm than the TB($\lambda$) algorithm. As a result, Retrace($\lambda$) does not truncate the $n$-step return as heavy as TB($\lambda$), and hence the parameter $\omega$ is larger (better) in Retrace($\lambda$) than in TB($\lambda$).

### 7.3.5 Finite-Sample Analysis of $Q$-Trace

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

**Theorem 7.3.5.** *Consider Algorithm 5 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 7.2.1 is satisfied, and $\bar{c} \leq \bar{\rho}$. Then, to achieve $\epsilon$-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($\lambda$) is that the importance sampling ratio $\rho(\cdot,\cdot)$ inside the temporal difference (line 4 of Algorithm 5) also involves a truncation. As shown in Subsection 7.2.3, due to introducing the truncation level $\bar{\rho}$, the algorithm converges to a biased limit $Q^{\pi,\rho}$ instead of $Q^\pi$. Such truncation bias can be controlled using Proposition 7.2.1. These observations agree with the results [16], where the finite-sample bounds of $Q$-trace were first established.

Compared to [16], we have an improved sample complexity. Specifically, the result in [16] 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 $\omega^{-1}$. 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 [16]. Similarly, since the $V$-trace algorithm [25] is an analog of the $Q$-trace algorithm, we can also improve the sample complexity for $V$-trace in [115].

In addition to analyzing existing algorithms, observe that our results, especially Theorem 7.2.2, provide sufficient conditions under which Algorithm 5 has provable finite-sample guarantees, and hence enable us to design new algorithms. As an example, in light of the Retrace($\lambda$) 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 $\lambda_c$, $\lambda_\rho$, $\bar{c}$, and $\bar{\rho}$ are tunable parameters. As long as $\lambda_c\bar{c}\leq\lambda_\rho\bar{\rho}<1/\gamma$, Theorem 7.2.2 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 $b$.

Table 7.1: Summary of the Sample Complexity Bounds

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

In view of Table 1, when $r_{\max}<1/\gamma$, which indicates that the target policy $\pi$ and the behavior policy $\pi_b$ 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($\lambda$), and Retrace($\lambda$), $Q^\pi(\lambda)$ has the best sample complexity bound. However, we need to point out that 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 [15]. 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.

## 7.4 Proof of All Theoretical Results

### 7.4.1 Proof of Theorem 7.2.1

We begin by explicitly computing the asynchronous generalized Bellman operator $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$. Let $\pi_c$ and $\pi_\rho$ 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 $\pi'$, let $P_{\pi'}$ be the transition probability matrix of the Markov chain $\{(S_k, A_k)\}$ under $\pi'$, 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 7.4.1.** *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 7.4.1, to prove Theorem 7.2.1, it is enough to study the matrix $A$. To proceed, we require the following definition.

**Definition 7.4.1.** Given $\beta \in [0,1]$, a matrix $M \in \mathbb{R}^{d\times d}$ is called a substochastic matrix with modulus $\beta$ 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 $\beta$ automatically implies that $\|M\|_\infty \leq 1 - \beta$.

We next show in the following two propositions that (1) the matrix $A$ given in Proposition 7.4.1 is a substochastic matrix with modulus $\omega$, 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 7.2.1.

**Proposition 7.4.2.** *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 7.4.1 is a substochastic matrix with modulus $\omega$.*

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 $\omega$ is an important parameter for our finite-sample analysis. In view of Theorem 7.2.1, we see that large modulus gives smaller (or better) contraction factor of $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$.

**Proposition 7.4.3.** *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 7.4.3 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.

[1] A non-negative matrix is irreducible if and only if its associated graph is strongly connected [123].

Note that Proposition 7.4.3 introduces the tunable parameter $\theta$. It is clear that large $\theta$ gives better contraction factor of $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ but worse lower bound on the entries of the weight vector $\mu$. In 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 $\omega$, 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 $\mu_d$ close to unity, we can get $\|M\|_{\mu,p}$ arbitrarily close to $1-\omega$. This is analogous to choosing $\theta$ close to one in Proposition 7.4.3.

### 7.4.2 Proof of Proposition 7.2.1

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

$$\begin{aligned}
&|[\mathcal{H}_\rho(Q_1)](s,a) - [\mathcal{H}_\rho(Q_2)](s,a)|\\
={}& \gamma\left|\sum_{s'\in\mathcal{A}} P_a(s,s')\sum_{a'\in\mathcal{A}}\pi_b(a'|s')\rho(s',a')(Q_1(s',a')-Q_2(s',a'))\right|\\
\leq{}& \gamma\sum_{s'\in\mathcal{A}} P_a(s,s')\sum_{a'\in\mathcal{A}}\pi_b(a'|s')\rho(s',a')|Q_1(s',a')-Q_2(s',a')|\\
\leq{}& \gamma\|Q_1-Q_2\|_\infty\sum_{s'\in\mathcal{A}} P_a(s,s')\sum_{a'\in\mathcal{A}}\pi_b(a'|s')\rho(s',a')\\
\leq{}& \gamma\sum_{s'\in\mathcal{A}} P_a(s,s')D_{\rho,\max}\|Q_1-Q_2\|_\infty\\
={}& \gamma D_{\rho,\max}\|Q_1-Q_2\|_\infty.
\end{aligned}$$

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

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

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

$$Q^{\pi,\rho} = \mathcal{H}_\rho(Q^{\pi,\rho}) = R + \gamma P_{\pi_\rho} D_\rho Q^{\pi,\rho}, \qquad (7.5)$$

where we recall the definition of $D_\rho$ and $\pi_\rho$ in Section 7.2. Equation 7.5 is equivalent to $Q^{\pi,\rho} = (I - \gamma P_{\pi_\rho} D_\rho)^{-1} R$. Therefore, we have

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

### 7.4.3 Proof of Proposition 7.4.1

Recall the definition of $\tilde{\mathcal{B}}_{c,\rho}(\cdot)$ in Equation 7.2:

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

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 $\pi_\rho$, 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

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

$$= \underbrace{\left[ I - \mathcal{K}_{SA} \sum_{i=0}^{n-1} (\gamma P_{\pi_c} D_c)^i (I - \gamma P_{\pi_\rho} D_\rho) \right]}_{A} Q + \underbrace{\mathcal{K}_{SA} \sum_{i=0}^{n-1} (\gamma P_{\pi_c} D_c)^i R}_{b} .$$

### 7.4.4 Proof of Proposition 7.4.2

Consider the matrix $A$ given in Proposition 7.4.1. To show that $A$ is a substochastic matrix with a positive modulus, we first show that $A$ is non-negative. Observe that

$$\begin{aligned} A &= I - \mathcal{K}_{SA} \sum_{i=0}^{n-1} (\gamma P_{\pi_c} D_c)^i + \mathcal{K}_{SA} \sum_{i=0}^{n-1} (\gamma P_{\pi_c} D_c)^i \gamma P_{\pi_\rho} D_\rho \\ &= (I - \mathcal{K}_{SA}) - \mathcal{K}_{SA} \sum_{i=1}^{n-1} (\gamma P_{\pi_c} D_c)^i + \mathcal{K}_{SA} \sum_{i=0}^{n-1} (\gamma P_{\pi_c} D_c)^i \gamma P_{\pi_\rho} D_\rho \\ &= (I - \mathcal{K}_{SA}) - \mathcal{K}_{SA} \sum_{i=0}^{n-2} (\gamma P_{\pi_c} D_c)^{i+1} + \mathcal{K}_{SA} \sum_{i=0}^{n-1} (\gamma P_{\pi_c} D_c)^i \gamma P_{\pi_\rho} D_\rho \\ &= (I - \mathcal{K}_{SA}) + \mathcal{K}_{SA} \sum_{i=0}^{n-2} (\gamma P_{\pi_c} D_c)^i \gamma (P_{\pi_\rho} D_\rho - P_{\pi_c} D_c) + \mathcal{K}_{SA} (\gamma P_{\pi_c} D_c)^{n-1} \gamma P_{\pi_\rho} D_\rho . \end{aligned} \tag{7.6}$$

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 $c(s', a') \leq \rho(s', a')$ for all $(s', a')$, we have

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

We next show that $A\mathbf{1} \leq (1 - \omega)\mathbf{1}$, 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

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

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

It follows that

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

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

### 7.4.5 Proof of Proposition 7.4.3

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

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

the matrix $M'$ is a substochastic matrix with modulus $0$, there exists a stochastic matrix $M''$ such that $M'' \geq M' > 0$. 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 $\mu$. Since $\mu^\top = \mu^\top M''$, we have

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

This proves the lower bound on the entries of $\mu$.

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

$$\begin{aligned}
\|Mx\|_{\mu,p}^p &= \sum_i \mu_i \left|\sum_j M_{ij}x_j\right|^p \\
&= \sum_i \mu_i \left(\sum_\ell M_{i\ell}\right)^p \left|\sum_j \frac{M_{ij}}{\sum_\ell M_{i\ell}} x_j\right|^p \\
&\leq \sum_i \mu_i \left(\sum_\ell M_{i\ell}\right)^{p-1} \sum_j M_{ij}|x_j|^p && \text{(Jensen's inequality)} \\
&\leq (1-\beta)^{p-1} \sum_i \mu_i \sum_j M_{ij}|x_j|^p \\
&\leq (1-\beta)^{p-1}(1-\theta\beta) \sum_i \mu_i \sum_j M'_{ij}|x_j|^p && \text{(definition of } M'\text{)} \\
&\leq (1-\beta)^{p-1}(1-\theta\beta) \sum_i \mu_i \sum_j M''_{ij}|x_j|^p && \text{(definition of } M''\text{)} \\
&= (1-\beta)^{p-1}(1-\theta\beta) \sum_j |x_j|^p \sum_i \mu_i M''_{ij} && \text{(change of summation order)} \\
&= (1-\beta)^{p-1}(1-\theta\beta) \sum_j \mu_j |x_j|^p && (\mu^\top M'' = \mu^\top) \\
&= (1-\beta)^{p-1}(1-\theta\beta)\|x\|_{\mu,p}^p.
\end{aligned}$$

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.

### 7.4.6 Proof of Theorem 7.2.2

We first state a more general result in the following, which implies Theorem 7.2.2.

**Theorem 7.4.1.** *Consider the iterates $\{Q_k\}$ generated by Algorithm 5. Suppose that As-*

*sumption 7.2.1 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* $\ell_p$*-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* $\alpha$ *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 $\ell_p$-norm), Theorem 7.4.1 implies the following finite-sample bound on $\mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_p]$.

**Corollary 7.4.1.** *Under same assumptions as Theorem 7.2.1, 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 7.2.2, observe that for any $p \geq 1$ we have

$$\begin{aligned}\mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_\infty^2] &\leq \mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_p^2]\\ &\leq \frac{\tilde{\zeta}_1}{\mu_{\min}^{2/p}}(1-\theta\omega\alpha)^{k-\tau_{\alpha,n}} + \frac{\tilde{\zeta}_2 pf(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2}{\mu_{\min}^{4/p}\omega}\alpha\tau_{\alpha,n}.\end{aligned}$$

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

$$\begin{aligned}&\frac{1}{\mu_{\min}^{2/p}} = \mu_{\min}^{-\frac{1}{2\log(1/\mu_{\min})}} = \mu_{\min}^{\frac{1}{2\log(\mu_{\min})}} = \sqrt{e} \leq 2, \quad \text{and}\\ &\frac{p}{\mu_{\min}^{4/p}} \leq 4e\log(1/\mu_{\min}) \leq 4e\log\left(\frac{2|\mathcal{S}||\mathcal{A}|}{\omega}\right). \qquad \text{(Using the lower bound on } \mu_{\min}\text{)}\end{aligned}$$

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 $k \geq \tau_{\alpha,n}$:

$$\mathbb{E}[\|Q_k - Q^{\pi,\rho}\|_\infty^2] \leq 2\tilde{\zeta}_1\left(1-\frac{\omega\alpha}{2}\right)^{k-\tau_{\alpha,n}}$$

$$
\begin{aligned}
&+ 4e\tilde{\zeta}_2 \frac{f(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2\log(2|\mathcal{S}||\mathcal{A}|/\omega)}{\omega}\alpha\tau_{\alpha,n} \\
&= \zeta_1\left(1-\frac{\omega\alpha}{2}\right)^{k-\tau_{\alpha,n}} \\
&\quad + \zeta_2 \frac{f(\gamma c_{\max})^2(\gamma\rho_{\max}+1)^2\log(2|\mathcal{S}||\mathcal{A}|/\omega)}{\omega}\alpha\tau_{\alpha,n},
\end{aligned}
$$

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 7.2.2.

### 7.4.7 Proof of Theorem 7.4.1

To prove Theorem 7.4.1, we apply Theorem 2.5.1, which studies general SA under contraction assumption. We begin by rewriting Algorithm 5 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 7.2.1 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

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

Then the update equation of Algorithm 5 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 $\{Y_k\}$.

**Proposition 7.4.4.** *The following statements hold.*

*(1) The operator $F(\cdot,\cdot)$ 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 $\max_{y\in\mathcal{Y}} \|P_{\pi_b}^{k+n+1}(y,\cdot) - \kappa_Y(\cdot)\|_{TV} \leq C\sigma^k$.*

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

We next present how to apply Theorem 2.5.1 to obtain the results. We begin by restating Theorem 2.5.1 in the case of weighted $\ell_p$-norm contraction with weights $\{\mu_i\}_{1\leq i\leq d}$. Using the notation in Chapter 2, we choose the smoothing norm $\|\cdot\|_s$ to be the same norm as the contraction norm: $\|\cdot\|_{\mu,p}$.

**Theorem 7.4.2.** *Consider the SA algorithm*

$$x_{k+1} = x_k + \alpha(F(x_k,Y_k) - x_k). \tag{7.7}$$

*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 $\kappa_Y$, 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 $\alpha$ 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 7.4.4 in conjunction with Theorem 7.2.1 imply that the requirements for applying Theorem 7.4.2 are satisfied. For any $\theta \in (0,1)$, when the constant stepsize $\alpha$ 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 $\tilde{\zeta}_2 = 228(3\|Q^{\pi,\rho}\|_{\mu,p} + 1)^2$.

### 7.4.8 Proof of Proposition 7.4.4

(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

$$\begin{aligned}
&\|F(Q_1, s_0, a_0, ..., s_n, a_n) - F(Q_2, s_0, a_0, ..., s_n, a_n)\|_{\mu,p}\\
\leq\;& \left[\sum_{s,a}\mu(s,a)\left(\mathbb{I}_{\{(s,a)=(s_0,a_0)\}}\sum_{i=0}^{n-1}(\gamma c_{\max})^i(\gamma\rho_{\max}+1)\|Q_1 - Q_2\|_\infty\right)^p\right]^{1/p}\\
&+ \|Q_1 - Q_2\|_{\mu,p} \qquad\qquad \text{(triangle inequality)}\\
=\;& f(\gamma c_{\max})(\gamma\rho_{\max}+1)\|Q_1 - Q_2\|_\infty + \|Q_1 - Q_2\|_{\mu,p}.\\
\leq\;& \frac{2}{\mu_{\min}^{1/p}}f(\gamma c_{\max})(\gamma\rho_{\max}+1)\|Q_1 - Q_2\|_{\mu,p}.
\end{aligned}$$

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 7.2.1, it is clear that $\{Y_k\}$ has a unique stationary distribution, which we have denoted by $\kappa_Y$, 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 $k \geq 0$:

$$\begin{aligned}
&\|P^{k+n+1}((s_0, a_0, ..., s_n, a_n), \cdot) - \kappa_Y(\cdot)\|_{\text{TV}} \\
=& \frac{1}{2} \sum_{s_0', a_0', ..., s_n', a_n'} \left| \sum_s P_{a_n}(s_n, s) P_{\pi_b}^k(s, s_0') - \kappa_S(s_0') \right| \left( \prod_{i=0}^{n-1} \pi(a_i'|s_i') P_{a_i'}(s_i', s_{i+1}') \right) \pi(a_n'|s_n') \\
=& \frac{1}{2} \sum_{s_0'} \left| \sum_s P_{a_n}(s_n, s) P_{\pi_b}^k(s, s_0') - \kappa_S(s_0') \right| \\
\leq& \frac{1}{2} \sum_{s_0'} \sum_s P_{a_n}(s_n, s) \left| P_{\pi_b}^k(s, s_0') - \kappa_S(s_0') \right| \\
=& \frac{1}{2} \sum_s P_{a_n}(s_n, s) \sum_{s_0'} \left| P_{\pi_b}^k(s, s_0') - \kappa_S(s_0') \right| \\
\leq& \frac{1}{2} \sum_s P_{a_n}(s_n, s) \max_{s'} \sum_{s_0'} \left| P_{\pi_b}^k(s', s_0') - \kappa_S(s_0') \right| \\
=& \max_{s \in \mathcal{S}} \|P_{\pi_b}^k(s, \cdot) - \kappa_S(\cdot)\|_{\text{TV}} \\
\leq& C\sigma^k.
\end{aligned}$$

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

### 7.4.9 Proof of Theorem 7.3.1

Since Vanilla IS is a special case of Algorithm 5, one can directly apply Theorem 7.2.2 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 7.4.4 (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

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

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

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

Using this improved Lipschitz constant and we obtain Theorem 7.3.1, where the rest of the proof is identical to that of Theorem 7.2.2.

### 7.4.10 Proof of Theorem 7.3.2 to Theorem 7.3.5

The results are obtained by directly applying Theorem 7.2.2.

## 7.5 Conclusion

In this chapter, 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 $\ell_p$-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$(\lambda)$, and Retrace$(\lambda)$, we provide the first-known results, and for $Q$-trace, we improve the result in [16]. The finite-sample bounds we establish also provide insights about the trade-offs between the bias in the limit and the variance in the stochastic iterates.

# CHAPTER 8

# OFF-POLICY CONTROL: $Q$-LEARNING

## 8.1 Introduction

The $Q$-learning algorithm is the most popular value-based RL algorithms in the literature. Specifically, a variant of $Q$-learning known as the Deep $Q$-Network was used at scale in solving practical problems, such as Atari games [7], robotics [124], and healthcare [125], etc.

Unlike TD-learning, which is for policy evaluation, and must be used in an actor-critic framework to find an optimal policy, $Q$-learning is for directly finding an optimal policy through finding the optimal $Q$-function. To motivate $Q$-learning, we first define the state-action value function (aka. the $Q$-function) in the following. Let $Q^\pi : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ be defined by

$$Q^\pi(s,a) = \mathbb{E}_\pi \left[ \sum_{k=0}^{\infty} \gamma^k \mathcal{R}(S_k, A_k) \,\middle|\, S_k = s, A_k = a \right], \; \forall\, (s,a) \in \mathcal{S} \times \mathcal{A}.$$

Similar to the $V$-function, $Q^\pi$ can be viewed as a vector in $\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$. Denote $Q^*$ as the $Q$-function associated with an optimal policy $\pi^*$. Note that all optimal policies share the same optimal $Q$-function. The motivation of the $Q$-learning algorithm is based on the following result [11, 1]:

$$\pi^* \text{ is an optimal policy } \iff \{a \mid \pi^*(a|s) > 0\} \subseteq \arg\max_{a \in \mathcal{A}} Q^*(s,a), \; \forall\, (s,a). \tag{8.1}$$

Note that $\arg\max_{a \in \mathcal{A}} Q^*(s,a)$ should be understood as a set since the maximum action may not be unique. Equation 8.1 states that the optimal policy is supported on the set of actions that maximize the optimal $Q$-function. Therefore, knowing the optimal $Q$-function alone is enough to compute an optimal policy.

To find the optimal $Q$-function, we next introduce the Bellman optimality equation. The optimal $Q$-function $Q^*$, uniquely solves the following system of equations:

$$Q^*(s,a) = \mathcal{R}(S_k, A_k) + \gamma \mathbb{E}\left[\max_{a' \in \mathcal{A}} Q^*(S_{k+1}, a') \;\middle|\; S_k = s, A_k = a\right], \ \forall\, (s,a) \in \mathcal{S} \times \mathcal{A}. \tag{8.2}$$

For simplicity of notation, let $\mathcal{H} : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \mapsto \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ be the Bellman optimality operator defined by

$$[\mathcal{H}(Q)](s,a) = \mathcal{R}(s,a) + \gamma \mathbb{E}\left[\max_{a' \in \mathcal{A}} Q(S_{k+1}, a') \;\middle|\; S_k = s, A_k = a\right]$$

for all $Q \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and $(s,a) \in \mathcal{S} \times \mathcal{A}$. Then Equation 8.2 can be written compactly as

$$Q^* = \mathcal{H}(Q^*), \tag{8.3}$$

which is the fixed-point equation of the Bellman optimality operator $\mathcal{H}(\cdot)$. Since the operator $\mathcal{H}(\cdot)$ is a contraction mapping with respect to the $\ell_\infty$-norm $\|\cdot\|_\infty$, with contraction factor being the discount factor $\gamma$ [9], Equation 8.3 can be efficiently solved using the fixed-point iteration:

$$Q_{k+1} = \mathcal{H}(Q_k), \ \forall\, k \geq 0. \tag{8.4}$$

which is also known as the value iteration algorithm, and converges to $Q^*$ geometrically fast [47]. However, to carry out such fixed-point iteration, we need to compute the expectation within the definition of the Bellman optimality operator $\mathcal{H}(\cdot)$, which is not possible since the environmental model is unknown in RL.

To overcome this challenge, [17] proposes the $Q$-learning algorithm, which can be viewed as a stochastic variant of Equation 8.4. Other variants of $Q$-function estimation algorithms includes SARSA [126], fitted $Q$-iteration [127], and zap $Q$-learning [128], etc.

### 8.1.1 Related Literature

The $Q$-learning algorithm [17] is perhaps one of the most well-known algorithms in RL literature. The asymptotic convergence of $Q$-learning was established in [24, 29, 31], and asymptotic convergence rate in [129, 128]. Beyond asymptotic behavior, finite-sample analysis of $Q$-learning was also thoroughly studied in the literature. The results are summarized in Table 8.1. Note that there is a different perspective about $Q$-learning in terms of regret bound [130], which is fundamentally different to the setting of this work.

From Table 8.1, we see that for synchronous $Q$-learning, the state-of-the-art mean square bound goes to [132, 119], and is $\tilde{\mathcal{O}}(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^5\epsilon^2})$. For synchronous $Q$-learning, [131] establishes the state-of-the-art concentration bound with only a $1/(1-\gamma)^4$ factor in the sample complexity.

In this thesis, we consider the asynchronous $Q$-learning algorithm, which is fundamentally different from synchronous $Q$-learning in terms of the update rule and the sample collecting process, and is more challenging to analyze. The state-of-the-art mean square bound of asynchronous $Q$-learning goes to [20] and concentration bound goes to [134]. It is clear that our result advances the results in [20] by a factor of at least $|\mathcal{S}||\mathcal{A}|$. To compare our result with the results in [134], we need to first translate concentration bound to mean square bound using the formula $\mathbb{E}[X] = \int_0^\infty \mathbb{P}(X > x)dx$ (which holds for any non-negative random variable $X$), and then perform the comparison. By using this translation technique for [134], the concentration bound in [134] does not imply our results. Even if we *conjecture* a stronger concentration bound based on the results in [134] and integrate that bound, the resulting mean square bound is no better than ours. See [16, Appendix B.5] for a detailed discussion.

Table 8.1: Summary of the Results about $Q$-Learning

| Algorithm | Reference | Sample Complexity | Guarantees |
|---|---|---|---|
| Synchronous $Q$-learning | [21] | $2^{\frac{1}{1-\gamma}}\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^4\epsilon^2}$ | concentration (tail) bound |
| | [131] | $\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^4\epsilon^2}$ | |
| | [19] | $\frac{\lvert\mathcal{S}\rvert^2\lvert\mathcal{A}\rvert^2}{(1-\gamma)^5\epsilon^2}$ | mean square bound |
| | [132] | $\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^5\epsilon^2}$ | |
| | [119] | $\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^5\epsilon^2}$ | |
| Asynchronous $Q$-learning | [21] | $\frac{t_{\text{cover}}^{\frac{1}{1-\gamma}}}{(1-\gamma)^4\epsilon^2}$ | concentration (tail) bound |
| | [133] | $\frac{t_{\text{mix}}}{\mathcal{K}_{SA,\min}^2(1-\gamma)^5\epsilon^2}$ | |
| | [134] | $\frac{t_{\text{cover}}}{(1-\gamma)^5\epsilon^2}$ | |
| | [20] | $\frac{t_{\text{cover}}^3\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^5\epsilon^2}$ | mean square bound |
| | *This work* | $\frac{1}{\mathcal{K}_{SA,\min}^3(1-\gamma)^5\epsilon^2}$ | |

In Table 8.1, all the polylogarithmic factors are ignored. The parameter $t_{\text{mix}} = t_{1/4}$ stands for the mixing time of the underlying Markov chain $\{(S_k, A_k)\}$ generated by the behavior policy $\pi_b$. The parameter $t_{\text{cover}}$ roughly represents the amount of time needed to visit all state-action pairs at least once. The parameter $\mathcal{K}_{SA,\min}$ is the minimal entry of the stationary distribution on $\mathcal{S} \times \mathcal{A}$. Note that we have $t_{\text{cover}} \geq |\mathcal{S}||\mathcal{A}|$ and $\mathcal{K}_{SA,\min} \leq 1/(|\mathcal{S}||\mathcal{A}|)$. Note that high probability (tail) bounds and mean square bounds are *not* directly comparable. See [16] for more details.

## 8.2 Finite-Sample Analysis

In this section, we formally present the $Q$-learning algorithm, reformulate it as a Markovian SA algorithm under a contractive operator, and apply Theorem 2.5.1 from Chapter 2 to establish the finite-sample bound and the sample complexity of $Q$-learning.

### 8.2.1 The $Q$-Learning Algorithm

We first present the $Q$-learning algorithm in the following.

Several remarks are in order. First of all, for ease of exposition, we use a fixed behavior

**Algorithm 6** $Q$-Learning

```
Input: Integer $K$, initialization $Q_0 \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, and behavior policy $\pi_b$
for $k = 0, 1, \cdots, K-1$ do
    Sample $A_k \sim \pi_b(\cdot|S_k)$, observe $S_{k+1} \sim P_{A_k}(S_k, \cdot)$
    $Q_{k+1}(S_k, A_k) = Q_k(S_k, A_k) + \alpha_k(\mathcal{R}(S_k, A_k) + \gamma \max_{a' \in \mathcal{A}} Q_k(S_{k+1}, a') - Q_k(S_k, A_k))$
end for
Output: $Q_K$
```

policy $\pi_b$ to present the $Q$-learning algorithm. In practice, the behavior policy can be time-varying. For example, it can be the $\epsilon$-greedy policy, or the $\epsilon$-softmax policy with respect to the current $Q$-function estimate. The asymptotic convergence of $Q$-learning is guaranteed as long as the behavior policy ensures sufficient exploration [24].

Similar to TD-learning, $Q$-learning performs the so-called asynchronous update, and the amount of update is equal to the difference between the LHS and the RHS of the Bellman optimality equation (cf. Equation 8.2), after replacing the expectation by sample estimate.

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

**Assumption 8.2.1.** The behavior policy $\pi_b$ satisfies $\pi_b(a|s) > 0$ for all $(s, a)$, and the Markov chain $\mathcal{M}_S = \{S_k\}$ induced by $\pi_b$ 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 [24]. The irreducibility and aperiodicity assumption is also standard in related work [92, 135]. Since we work with finite-state MDPs, Assumption 8.2.1 implies that $\mathcal{M}_S$ has a unique stationary distribution, denoted by $\kappa_S \in \Delta^{|\mathcal{S}|}$, and $\mathcal{M}_S$ mixes at a geometric rate [48].

### 8.2.2 Reformulation through Markovian SA

In this section, we formally remodeling the $Q$-learning algorithm as a Markovian SA algorithm in the form of Equation 4.12. Let $Y_k = (S_k, A_k, S_{k+1})$ for all $k \geq 0$. Note that the random process $\mathcal{M}_Y = \{Y_k\}$ is also a Markov chain, whose state-space is given by

$$\mathcal{Y} = \{(s, a, s') \mid s \in \mathcal{S}, \pi_b(a|s) > 0, P_a(s, s') > 0\},$$

and is finite. Define an operator $F : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \times \mathcal{Y} \mapsto \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ by

$$\begin{aligned}[F(Q, y)](s, a) &= [F(Q, s_0, a_0, s_1)](s, a)\\ &= \mathbb{1}_{\{(s_0,a_0)=(s,a)\}}(\mathcal{R}(s_0, a_0) + \gamma \max_{a'\in\mathcal{A}} Q(s_1, a') - Q(s_0, a_0)) + Q(s, a)\end{aligned}$$

for all $(s, a)$. Then the update equation of the $Q$-learning algorithm (i.e., Algorithm 6 line $4$) can be written by

$$Q_{k+1} = Q_k + \alpha_k (F(Q_k, Y_k) - Q_k),$$

which is in the same form of Equation 4.12 with $w_k$ being identically equal to zero. Next, we establish the properties of the operator $F(\cdot, \cdot)$ and the Markov chain $\{Y_k\}$ in the following proposition, which guarantees that Assumptions 2.2.2 – 2.2.3 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 $\{\kappa_S(s)\pi_b(a|s)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$ sitting on its diagonal. Let $\mathcal{K}_{SA,\min} = \min_{(s,a)} \kappa_S(s)\pi_b(a|s)$, which is positive under Assumption 8.2.1.

**Proposition 8.2.1.** *Suppose that Assumption 8.2.1 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 all $Q_1, Q_2 \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and $y \in \mathcal{Y}$,*

*(b) $\|F(\mathbf{0}, y)\|_\infty \leq 1$ for all $y \in \mathcal{Y}$.*

*(2) The Markov chain $\mathcal{M}_Y = \{Y_k\}$ has a unique stationary distribution $\mu_Y$, and there*

*exist $C > 0$ and $\sigma \in (0,1)$ such that*

$$\max_{y\in\mathcal{Y}} \|P_{\pi_b}^{k+1}(y,\cdot) - \mu_Y(\cdot)\|_{TV} \leq C\sigma^k,\ \forall\ k \geq 0.$$

*(3) Define an operator $\bar{F} : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \mapsto \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ by $\bar{F}(Q) = \mathbb{E}_{Y\sim\mu_Y}[F(Q,Y)]$. Then*

*(a) $\bar{F}(\cdot)$ is explicitly given by $\bar{F}(Q) = \mathcal{K}_{SA}\mathcal{H}(Q) + (I - \mathcal{K}_{SA})Q$, where $\mathcal{H}(\cdot)$ is the Bellman optimality operator.*

*(b) $\bar{F}(\cdot)$ is a contraction mapping with respect to $\|\cdot\|_\infty$, with contraction factor $\beta := 1 - \mathcal{K}_{SA,\min}(1-\gamma)$.*

*(c) $\bar{F}(\cdot)$ has a unique fixed-point $Q^*$.*

As we see, the $(s,a)$-th entry of the asynchronous Bellman operator $\bar{F}(Q)$ 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 captures the nature of performing asynchronous update as illustrated in Chapter 5.

### 8.2.3 Finite-Sample Guarantees

Proposition 8.2.1 enables us to apply Theorem Theorem 2.5.1 to the $Q$-learning algorithm. For ease of exposition, we only present the result of using constant stepsize. Define

$$t_\delta = \min\left\{k \geq 0 : \max_{s\in\mathcal{S}} \|P_{\pi_b}^k(s,\cdot) - \kappa_S(\cdot)\|_{\text{TV}} \leq \delta\right\}$$

as the mixing time of the Markov chain $\{S_k\}$ with precision $\delta$.

**Theorem 8.2.1.** *Consider $\{Q_k\}$ of Algorithm 6. Suppose that Assumption 8.2.1 is satisfied, and $\alpha_k = \alpha$ for all $k \geq 0$, where $\alpha$ is chosen such that $\alpha(t_\alpha + 1) \leq c_{Q,0}\frac{(1-\beta)^2}{\log(|\mathcal{S}||\mathcal{A}|)}$ (where*

*$c_{Q,0}$ is a numerical constant). Then we have for all $k \geq t_\alpha$:*

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

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

*Remark.* Using Proposition 8.2.1 (2), we see that $t_\alpha$ produces an additional $\log(1/\alpha)$ factor in the bound.

We view the first term on the RHS of the convergence bound as the the bias, and the second term as the variance. Since we are using constant stepsize, the bias term goes to zero geometrically fast while the variance is of the size $\mathcal{O}(\alpha\log(1/\alpha))$.

Based on Theorem 8.2.1, we next derive the sample complexity of $Q$-learning.

**Corollary 8.2.1.** *In order to make $\mathbb{E}[\|Q_k - Q^*\|_\infty] \leq \epsilon$, where $\epsilon > 0$ is a given accuracy, the total number of samples required is of the size*

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

*Remark.* We upper bound $\|Q^*\|_\infty$ by $1/(1-\gamma)$ in deriving the sample complexity result.

From Corollary 8.2.1, we see that the dependence on the accuracy $\epsilon$ 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 [20]. The parameter $\mathcal{K}_{SA,\min} = \min_{s,a}\kappa_S(s)\pi_b(a|s)$ captures the quality of exploration of the behavior policy $\pi_b$. Since $\mathcal{K}_{SA,\min} \geq 1/|\mathcal{S}||\mathcal{A}|$, we see that the best possible dependence on the size of the state-action space is $\tilde{\mathcal{O}}(|\mathcal{S}|^3|\mathcal{A}|^3)$.

## 8.3 Proof of All Theoretical Results

### 8.3.1 Proof of Proposition 8.2.1

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

$$\begin{aligned}\|F(Q_1, y) - F(Q_2, y)\|_\infty \leq\; & \max_{(s,a)} \left| \gamma \mathbb{1}_{\{(s_0,a_0)=(s,a)\}} (\max_{a_1 \in \mathcal{A}} Q_1(s_1, a_1) - \max_{a_2 \in \mathcal{A}} Q_2(s_1, a_2)) \right| \\ & + \max_{s,a} \left| \mathbb{1}_{\{(s_0,a_0)\neq(s,a)\}} (Q_1(s_0, a_0) - Q_2(s_0, a_0)) \right| \\ \leq\; & 2\|Q_1 - Q_2\|_\infty.\end{aligned}$$

Similarly, for any $y \in \mathcal{Y}$, we have

$$\|F(\mathbf{0}, y)\|_\infty = \max_{(s,a)} \left| \mathbb{1}_{\{(s_0,a_0)=(s,a)\}} \mathcal{R}(s_0, a_0) \right| \leq 1.$$

(2) It is clear from Assumption 8.2.1 that $\{Y_k\}$ has a unique stationary distribution, which we have denoted by $\mu_Y$. Moreover, we have $\mu(s, a, s') = \kappa_S(s)\pi_b(a|s)P_a(s, s')$ for any $(s, a, s') \in \mathcal{Y}$. Consider the second claim. Using the definition of total variation distance, we have for all $k \geq 0$:

$$\begin{aligned}& \max_{y \in \mathcal{Y}} \|P_{\pi_b}^{k+1}(y, \cdot) - \mu_Y(\cdot)\|_{\text{TV}} \\ =\; & \frac{1}{2} \max_{(s_0,a_0,s_1) \in \mathcal{Y}} \sum_{s,a,s'} |P_{\pi_b}^{k+1}((s_0, a_0, s_1), (s, a, s')) - \kappa_S(s)\pi_b(a|s)P_a(s, s')| \\ =\; & \frac{1}{2} \max_{s_1 \in \mathcal{S}} \sum_{s,a,s'} |P_{\pi_b}^{k}(s_1, s)\pi_b(a|s)P_a(s, s') - \kappa_S(s)\pi_b(a|s)P_a(s, s')| \\ =\; & \frac{1}{2} \max_{s_1 \in \mathcal{S}} \sum_{s} |P_{\pi_b}^{k}(s_1, s) - \kappa_S(s)| \\ =\; & \max_{s \in \mathcal{S}} \|P_{\pi_b}^{k}(s, \cdot) - \kappa_S(\cdot)\|_{\text{TV}} \\ \leq\; & C\sigma^k,\end{aligned}$$

where $C > 0$ and $\sigma \in (0,1)$ are constants. Note that the last line of the previous inequality follows from Assumption 8.2.1.

(3) (a) Using the Markov property, we have for any $Q \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and $(s,a)$:

$$\begin{aligned}
&\mathbb{E}_{S_k\sim\kappa_S}\left[[F(Q,S_k,A_k,S_{k+1})](s,a)\right] \\
=\;&\mathbb{E}_{S_k\sim\kappa_S}\left[\mathbb{1}_{\{(S_k,A_k)=(s,a)\}}\left(\mathcal{R}(S_k,A_k)+\gamma\max_{a'\in\mathcal{A}}Q(S_{k+1},a')-Q(S_k,A_k)\right)+Q(s,a)\right] \\
=\;&\mathbb{E}_{S_k\sim\kappa_S}\left[\mathbb{1}_{\{(S_k,A_k)=(s,a)\}}\left(\mathcal{R}(S_k,A_k)+\gamma\max_{a'\in\mathcal{A}}Q(S_{k+1},a')\right)\right. \\
&\left.\quad+(1-\mathbb{1}_{\{(S_k,A_k)=(s,a)\}})Q(S_k,A_k)\right] \\
=\;&\kappa_S(s)\pi_b(a|s)[\mathcal{H}(Q)](s,a)+(1-\kappa_S(s)\pi_b(a|s))Q(s,a),
\end{aligned}$$

where $\mathcal{H}(\cdot)$ is the Bellman optimality operator. Now use the definition of the matrix $\mathcal{K}_{SA}$ and we have $\bar{F}(Q)=\mathcal{K}_{SA}\mathcal{H}(Q)+(I-\mathcal{K}_{SA})Q$.

(3) (b) Since it is well-known that the Bellman optimality operator $\mathcal{H}(\cdot)$ is a $\gamma$-contraction with respect to $\|\cdot\|_\infty$, we have for any $Q_1,Q_2\in\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$:

$$\begin{aligned}
&\|\bar{F}(Q_1)-\bar{F}(Q_2)\|_\infty \\
=\;&\|N(\mathcal{H}(Q_1)-\mathcal{H}(Q_2))+(I-N)(Q_1-Q_2)\|_\infty \\
=\;&\max_{(s,a)}|\kappa_S(s)\pi_b(a|s)([\mathcal{H}(Q_1)](s,a)-[\mathcal{H}(Q_2)](s,a)) \\
&\quad+(1-\kappa_S(s)\pi_b(a|s))(Q_1(s,a)-Q_2(s,a))| \\
\leq\;&\max_{(s,a)}\left[\kappa_S(s)\pi_b(a|s)|[\mathcal{H}(Q_1)](s,a)-[\mathcal{H}(Q_2)](s,a)|\right. \\
&\left.\quad+(1-\kappa_S(s)\pi_b(a|s))|Q_1(s,a)-Q_2(s,a)|\right] \\
\leq\;&\max_{(s,a)}\left[\kappa_S(s)\pi_b(a|s)\|\mathcal{H}(Q_1)-\mathcal{H}(Q_2)\|_\infty+(1-\kappa_S(s)\pi_b(a|s))\|Q_1-Q_2\|_\infty\right]
\end{aligned}$$

$$\begin{aligned}
&\leq \max_{(s,a)}\left[\kappa_S(s)\pi_b(a|s)\gamma\|Q_1-Q_2\|_\infty+(1-\kappa_S(s)\pi_b(a|s))\|Q_1-Q_2\|_\infty\right]\\
&=\|Q_1-Q_2\|_\infty\max_{(s,a)}(1-(1-\gamma)\kappa_S(s)\pi_b(a|s))\\
&=(1-\mathcal{K}_{SA,\min}(1-\gamma))\|Q_1-Q_2\|_\infty.
\end{aligned}$$

Therefore, the operator $\bar{F}(\cdot)$ is a contraction mapping with respect to $\|\cdot\|_\infty$, with contraction factor $\beta=1-\mathcal{K}_{SA,\min}(1-\gamma)$.

(3) (c) It is enough to show that $Q^*$ is a fixed-point of $\bar{F}(\cdot)$, the uniqueness part follows from $\bar{F}(\cdot)$ being a contraction [47]. Using the fact that $\mathcal{H}(Q^*)=Q^*$, we have

$$\bar{F}(Q^*)=\mathcal{K}_{SA}\mathcal{H}(Q^*)+(I-\mathcal{K}_{SA})Q^*=\mathcal{K}_{SA}Q^*+(I-\mathcal{K}_{SA})Q^*=Q^*.$$

### 8.3.2 Proof of Theorem 8.2.1

We prove Theorem 8.2.1 using Theorem 2.5.1 for general Markovian SA under a contraction operator. Since the contraction norm is $\|\cdot\|_\infty$, Corollary 2.5.1 (2) is applicable. To apply Theorem 2.5.1, we first identify the corresponding constants using Proposition 8.2.1 in the following:

$$\begin{aligned}
&A=A_1+A_2+1=3,\ B=B_1+B_2=1,\ \varphi_1\leq 3,\ \varphi_2\geq\frac{1-\beta}{2},\\
&\varphi_3\leq\frac{456e\log(|\mathcal{S}||\mathcal{A}|)}{1-\beta},\ c_1\leq(\|Q_0-Q^*\|_\infty+\|Q_0\|_\infty+1)^2,\ c_2=(3\|Q^*\|_\infty+1)^2.
\end{aligned}$$

Now we apply Theorem Theorem 2.5.1 (2) (a). When $\alpha_k\equiv\alpha$ with $\alpha$ chosen such that

$$\alpha t_\alpha\leq\frac{\varphi_2}{\varphi_3A^2}\frac{(1-\beta)^2}{8208e\log(|\mathcal{S}||\mathcal{A}|)}.$$

we have for all $k \geq t_\alpha$:

$$
\begin{aligned}
\mathbb{E}[\|Q_k - Q^*\|_\infty^2] &\leq \varphi_1 c_1 \left(1 - \frac{1-\beta}{2}\alpha\right)^{k-t_\alpha} + \frac{\varphi_3 c_2}{\varphi_2}\alpha t_\alpha(\mathcal{M}_Y) \\
&\leq 3(\|Q_0 - Q^*\|_\infty + \|Q_0\|_\infty + 1)^2 \left(1 - \frac{1-\beta}{2}\alpha\right)^{k-t_\alpha} \\
&\quad + \frac{912e\log(|\mathcal{S}||\mathcal{A}|)}{(1-\beta)^2}(3\|Q^*\|_\infty + 1)^2 \alpha t_\alpha \\
&= c_{Q,1}\left(1 - \frac{1-\beta}{2}\alpha\right)^{k-t_\alpha} + c_{Q,2}\frac{\log(|\mathcal{S}||\mathcal{A}|)}{(1-\beta)^2}\alpha t_\alpha,
\end{aligned}
$$

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

## 8.4 Conclusion and Future Work

In this chapter, we have established the finite-sample mean-square bounds of the $Q$-learning algorithm, which implies an $\tilde{\mathcal{O}}(\frac{1}{\epsilon^2(1-\gamma)^5})$ sample complexity. Our approach is to formulate $Q$-learning as a Markovian SA algorithm under an $\ell_\infty$-norm contraction operator, which is called the asynchronous Bellman operator. The finite-sample bounds then follow from our SA results in Chapter 2.

Empirically, numerical simulations presented in [132] suggest that the dependence on the effective horizon is $1/(1-\gamma)^4$, implying that there is a gap between theory and experiment. The $1/(1-\gamma)^4$ dependence was later established theoretically for synchronous $Q$-learning in [131]. Establishing the $1/(1-\gamma)^4$ dependence for the more practical asynchronous $Q$-learning is an immediate future direction of this line of work.

# Part III

# RL with Linear Function Approximation

In Part II, we provide a unified approach for finite-sample analysis of tabular RL algorithms. However, tabular RL algorithms become computationally intractable when the size of the state-action space is large. This motivates the use of function approximation. The idea here is to approximate the desired quantity (e.g. $Q$-function, $V$-function, etc.) from a pre-specified function class, thereby artificially reducing the complexity of the problem. For example, the popular Deep $Q$-Network is designed to approximate the optimal $Q$-function using deep neural net. Since most of the realistic RL problems have huge state-action spaces, function approximation is of vital importance for the successes of RL.

Theoretically, a major challenge for studying RL with function approximation is the deadly triad, which refers to bootstrapping, off-policy sampling, and function approximation. In particular, it was observed in the literature that when the deadly triad appears, RL algorithms can diverge. In this part of the thesis, we are going to consider RL with linear function approximation, and design algorithms with provable convergence and finite-sample guarantees in the presence of the deadly triad.

We first consider TD-learning in Chapter 9 and design a convergent algorithm under off-policy sampling and linear function approximation, where we exploit the advantage of using multi-step return. Such TD-learning algorithm was later used in Chapter 10 to study general policy-based methods, where we establish the $\mathcal{O}(\epsilon^{-2})$ sample complexity. In Chapter 11, we switch our focus to $Q$-learning with linear function approximation, and show that the algorithm provably converges and establish the finite-sample bounds when the discount factor of the problem is small. Later in Chapter 12, by modifying the original $Q$-learning algorithm using target network and truncation, we successfully remove the restriction on the discount factor for achieving convergence.

# CHAPTER 9
# OFF-POLICY TD-LEARNING WITH LINEAR FUNCTION APPROXIMATION

## 9.1 Introduction

Recall the TD-learning algorithms we studied in Chapter 6 and Chapter 7. An important feature there is that TD-learning performs asynchronous update, where only a single entry of the vector-valued iterate is updated in each time step. Therefore, we require at least $|\mathcal{S}|$ (or $|\mathcal{S}||\mathcal{A}|$) amount of samples to update each entry of $V_k$ (or $Q_k$ if we are estimating $Q^\pi$) once. In practical applications, the state-action space of the RL problems is usually extremely large. Therefore tabular TD-learning becomes computationally intractable.

To overcome the curse of dimensionality, in this chapter, we consider TD-learning (for evaluating $Q^\pi$ of some target policy $\pi$) using linear function approximation, and we employ off-policy sampling. However, when TD-learning is used along with off-policy sampling and linear function approximation, the deadly triad appears and the algorithm can be unstable. In addition, the product of the importance sampling ratios in off-policy learning may cause high variance in the stochastic iterates.

We propose a generic framework of TD-learning algorithms (including two specific algorithms: the $\lambda$-averaged $Q$-trace and the two-sided $Q$-trace), which provably converge in the presence of the deadly triad, and do not suffer from the high variance issue in off-policy learning (albeit at a cost of a bias in the limit point).

### 9.1.1 Related Literature

The TD-learning method is used to solve the policy evaluation sub-problem, and is usually used in policy-based methods to ultimately find an optimal policy. The asymptotic convergence of TD-learning was established in [24, 102, 136]. Finite-sample analysis of variants of TD-learning algorithms using on-policy sampling was performed in [16], and using off-policy sampling was performed in [137, 138].

In the function approximation setting, TD-learning with linear function approximation was studied in [92, 139, 12, 40] when using on-policy sampling. In the off-policy linear function approximation setting, due to the presence of the deadly triad, TD-learning algorithms can diverge [1]. Variants of TD-learning algorithms such as TDC [121], GTD [140], emphathic TD [122], and $n$-step TD (with a large enough $n$) [141] were used to resolve the divergence issue, and the finite-sample bounds were studied in [142, 143, 141]. Note that TDC, GTD, and emphathic TD are two time-scale algorithms, while $n$-step TD is single time-scale, it suffers from a high variance due to the cumulative product of the importance sampling ratios.

### 9.1.2 Problem Setting

In linear function approximation, we choose a set of basis vectors $\phi_i \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, $1 \leq i \leq d$. Let $\Phi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times d}$ be a matrix defined by $\Phi = [\phi_1, \cdots, \phi_d]$. Then, the goal is to find from the linear sub-space $\mathcal{Q} = \{\tilde{Q}_w = \Phi w \mid w \in \mathbb{R}^d\}$ the "best" approximation of the $Q$-function $Q^\pi$, where $w \in \mathbb{R}^d$ is the weight vector.

Let $\phi(s,a) = [\phi_1(s,a), \phi_2(s,a), \cdots, \phi_d(s,a)]^\top \in \mathbb{R}^d$ be the feature vector associated with the pair $(s,a)$. Throughout this chapter, we impose the following assumption on the basis vectors.

**Assumption 9.1.1.** The matrix $\Phi$ has full column-rank, and satisfies $\|\Phi\|_\infty \leq 1$.

Assumption 9.1.1 is indeed without loss of generality since we can disregard dependent basis vectors, and performing feature normalization does not change the approximation power of the function class.

## 9.2 Algorithm Design

We present in Algorithm 7 a generic TD-learning algorithm using off-policy sampling and linear function approximation.

**Algorithm 7** A Generic Multi-Step Off-Policy TD-Learning with Linear Function Approximation

```
Input: Integer $K$, bootstrapping parameter $n$, stepsize sequence $\{\alpha_k\}$, initialization $w_0$, target policy $\pi$, behavior policy $\pi_b$, generalized importance sampling ratios $c, \rho : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}_+$, and a single trajectory of samples $\{(S_k, A_k)\}_{0 \leq k \leq K+n-1}$ generated by the behavior policy $\pi_b$.
for $k = 0, 1, \cdots, K-1$ do
    $\Delta_i(w_k) = \mathcal{R}(S_i, A_i) + \gamma\rho(S_{i+1}, A_{i+1})\phi(S_{i+1}, A_{i+1})^\top w_k - \phi(S_i, A_i)^\top w_k$, $i \in \{k, k+1, \cdots, k+n-1\}$
    $w_{k+1} = w_k + \alpha_k \phi(S_k, A_k) \sum_{i=k}^{k+n-1} \gamma^{i-k} \prod_{j=k+1}^{i} c(S_j, A_j)\Delta_i(w_k)$
end for
Output: $w_K$
```

In Algorithm 7, the choice of the generalized importance sampling ratios $c(\cdot,\cdot)$ and $\rho(\cdot,\cdot)$ is of vital importance. We next present two specific choices, resulting in two novel algorithms called $\lambda$-averaged $Q$-trace and two-sided $Q$-trace.

**The $\lambda$-Averaged $Q$-Trace Algorithm.** Let $\lambda \in \mathbb{R}^{|\mathcal{S}|}$ be a vector-valued tunable parameter satisfying $\lambda \in [\mathbf{0}, \mathbf{1}]$. Then the generalized importance sampling ratios are chosen as $c(s,a) = \rho(s,a) = \lambda(s)\frac{\pi(a|s)}{\pi_b(a|s)} + 1 - \lambda(s)$ for all $(s,a)$. Observe that when $\lambda = \mathbf{1}$, we have $c(s,a) = \rho(s,a) = \frac{\pi(a|s)}{\pi_b(a|s)}$, and Algorithm 7 reduces to the convergent multi-step off-policy TD-learning algorithm presented in [141], which however suffers from an exponential large variance due to the cumulative product of the importance sampling ratios. On the other hand, when $\lambda = \mathbf{0}$, we have $c(s,a) = \rho(s,a) = 1$, and hence the product of the generalized importance sampling ratios is deterministically equal to one, resulting in no variance at all. However, in this case, we are essentially performing policy evaluation of the behavior policy $\pi_b$ instead of the target policy $\pi$, hence there will be a bias in the limit of Algorithm 7. More generally, when $\lambda \in (\mathbf{0}, \mathbf{1})$, there is a trade-off between the variance and the bias in the limit point. Such trade-off will be studied quantitatively in Section 9.4.

**The Two-Sided $Q$-Trace Algorithm.** To introduce the algorithm, we first define the two-sided truncation function. Given upper and lower truncation levels $a, b \in \mathbb{R}$, define $g_{a,b} : \mathbb{R} \mapsto \mathbb{R}$ by $g_{a,b}(x) = a$ when $x < a$, $g_{a,b}(x) = x$ when $a \leq x \leq b$, and $g_{a,b}(x) = b$ when $x > b$. Let $\ell, u \in \mathbb{R}^{|\mathcal{S}|}$ be two vector-valued tunable parameters satisfying $\mathbf{0} \leq \ell \leq \mathbf{1} \leq u$. Then, for the two-sided $Q$-trace algorithm, the generalized importance sampling ratios are chosen as $c(s,a) = \rho(s,a) = g_{\ell(s),u(s)}\left(\pi(a|s)/\pi_b(a|s)\right)$ for all $(s,a)$. The idea of truncating the importance sampling ratios from above was already employed in existing algorithms such as Retrace($\lambda$) [15], $V$-trace [25], and $Q$-trace [137], and is used to control the high variance in off-policy learning. However, none of them were shown to converge in the function approximation setting. Introducing the lower truncation level is crucial to ensure the convergence of the two-sided $Q$-trace algorithm in the presence of the deadly triad. This will be illustrated in detail in Section 9.4.

## 9.3 The Generalized PBE

We next theoretically analyze Algorithm 7. Specifically, in this section, we formulate Algorithm 7 as an SA algorithm for solving a generalized PBE and study its properties. We begin by stating our assumption.

**Assumption 9.3.1.** The behavior policy $\pi_b$ satisfies $\pi_b(a|s) > 0$ for all $(s,a)$, and induces an irreducible and aperiodic Markov chain $\{S_k\}$.

Assumption 9.3.1 implies that the Markov chain $\{S_k\}$ induced by $\pi_b$ has a unique stationary distribution $\kappa_S \in \Delta^{|\mathcal{S}|}$. Moreover, there exist $C \geq 1$ and $\sigma \in (0,1)$ such that $\max_{s \in \mathcal{S}} \|P_{\pi_b}^k(s,\cdot) - \kappa_S(\cdot)\|_{\text{TV}} \leq C\sigma^k$ for all $k \geq 0$, where $P_{\pi_b}$ is the transition probability matrix of the Markov chain $\{S_k\}$ under $\pi_b$ [48].

For simplicity, denote $c_{i,j} = \prod_{k=i}^{j} c(S_k, A_k)$. The target equation Algorithm 7 aims at solving is:

$$\mathbb{E}_{S_0\sim\kappa_S}\left[\phi(S_0,A_0)\sum_{i=0}^{n-1}\gamma^i c_{1,i}\Delta_i(w)\right]=0, \tag{9.1}$$

where $A_i\sim\pi_b(\cdot|S_i)$ and $S_{i+1}\sim P_{A_i}(S_i,\cdot)$. The following lemma formulates Equation 9.1 in the form of a generalized PBE. To present the lemma, we first introduce some notation. Let $\mathcal{K}_{SA}\in\mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times|\mathcal{S}||\mathcal{A}|}$ be a diagonal matrix with diagonal entries $\{\kappa_S(s)\pi_b(a|s)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$, and let $\mathcal{K}_{SA,\min}$ be the minimal diagonal entry. Let $\|\cdot\|_{\mathcal{K}_{SA}}$ be the weighted $\ell_2$-norm with weights $\{\kappa_S(s)\pi_b(a|s)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$, and denote $\text{Proj}_{\mathcal{Q}}$ as the projection operator onto the linear sub-space $\mathcal{Q}$ with respect to $\|\cdot\|_{\mathcal{K}_{SA}}$. 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 $[\mathcal{T}_c(Q)](s,a)=\sum_{i=0}^{n-1}\mathbb{E}_{\pi_b}[\gamma^i c_{1,i}Q(S_i,A_i)\mid S_0=s,A_0=a]$ 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 any $Q\in\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and state-action pair $(s,a)$.

**Lemma 9.3.1.** *Equation 9.1 is equivalent to*

$$\Phi w=\textit{Proj}_{\mathcal{Q}}\mathcal{B}_{c,\rho}(\Phi w), \tag{9.2}$$

*where $\mathcal{B}_{c,\rho}(\cdot)$ is the generalized Bellman operator defined by $\mathcal{B}_{c,\rho}(Q)=\mathcal{T}_c(\mathcal{H}_\rho(Q)-Q)+Q$.*

The generalized Bellman operator $\mathcal{B}_{c,\rho}(\cdot)$ was previously introduced in Chapter 7 to study off-policy TD-learning algorithms in the *tabular* setting (i.e., $\Phi=I_{SA}$), where the contraction property of $\mathcal{B}_{c,\rho}(\cdot)$ was shown. However, $\mathcal{B}_{c,\rho}(\cdot)$ alone being a contraction is not enough to guarantee the convergence of Algorithm 7 because of function approximation, which introduces an additional projection operator $\text{Proj}_{\mathcal{Q}}$. What we truly need is that (1) the composed operator $\text{Proj}_{\mathcal{Q}}\mathcal{B}_{c,\rho}(\cdot)$ is a contraction mapping, and (2) the solution $w_{c,\rho}^\pi$ of Equation 9.2 is such that $\Phi w_{c,\rho}^\pi$ is an approximation of the $Q$-function $Q^\pi$. We next provide sufficient conditions on the choices of the generalized importance sampling ratios $c(\cdot,\cdot)$ and $\rho(\cdot,\cdot)$, and the bootstrapping parameter $n$ so that the above two requirements are satisfied.

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 $(s,a)$. Let $D_{c,\max}$ and $D_{\rho,\max}$ ($D_{c,\min}$ and $D_{\rho,\min}$) be the maximam (minimum) diagonal entries of the matrices $D_c$ and $D_\rho$ respectively.

**Condition 9.3.1.** The generalized importance sampling ratios $c(\cdot,\cdot)$ and $\rho(\cdot,\cdot)$ satisfy (1) $c(s,a) \leq \rho(s,a), \forall\, (s,a)$, (2) $D_{\rho,\max} < 1/\gamma$, and (3) $\frac{\gamma(D_{\rho,\max}-D_{c,\min})}{(1-\gamma D_{c,\min})\sqrt{\mathcal{K}_{SA,\min}}} < 1$.

Condition 9.3.1 (1) and (2) were introduced in Chapter 7, and were used to show the contraction property of the operator $\mathcal{B}_{c,\rho}(\cdot)$. In particular, it was shown that the generalized Bellman operator $\mathcal{B}_{c,\rho}(\cdot)$ is a contraction mapping with respect to $\|\cdot\|_\infty$, with contraction factor $\tilde{\gamma}(n) = 1 - f_n(\gamma D_{c,\min})(1-\gamma D_{\rho,\max})$, where $f_n : \mathbb{R} \mapsto \mathbb{R}$ is defined by $f_n(x) = \sum_{i=0}^{n-1} x^i$ for any $x$. It is clear that $\tilde{\gamma}(n) \in (0,1)$, and is a decreasing function of $n$.

As illustrated earlier, $\mathcal{B}_{c,\rho}(\cdot)$ being a contraction mapping is not sufficient to guarantee the stability of Algorithm 7. We require the composed operator $\text{Proj}_{\mathcal{Q}}\mathcal{B}_{c,\rho}(\cdot)$ to be contraction mapping with appropriate choice of $n$. This is guaranteed by Condition 9.3.1 (3). To see this, first note that we have the following lemma, which is obtained by using the contraction property of $\mathcal{B}_{c,\rho}(\cdot)$ and the "equivalence" between norms in finite-dimensional spaces.

**Lemma 9.3.2.** *Under Condition 9.3.1, it holds for any $Q_1, Q_2 \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ that*

$$\|Proj_{\mathcal{Q}}\mathcal{B}_{c,\rho}(Q_1) - Proj_{\mathcal{Q}}\mathcal{B}_{c,\rho}(Q_2)\|_{\mathcal{K}_{SA}} \leq \frac{\tilde{\gamma}(n)}{\sqrt{\mathcal{K}_{SA,\min}}}\|Q_1 - Q_2\|_{\mathcal{K}_{SA}}.$$

In view of Lemma 9.3.2, the composed operator $\text{Proj}_{\mathcal{Q}}\mathcal{B}_{c,\rho}(\cdot)$ is a contraction mapping as long as $\lim_{n\to\infty}\tilde{\gamma}(n)/\sqrt{\mathcal{K}_{SA,\min}} < 1$, which after straightforward algebra is equivalent to Condition 9.3.1 (3).

To satisfy Condition 9.3.1 (3), intuitively we should make $D_{\rho,\max}$ and $D_{c,\min}$ arbitrarily close to each other. It is not clear if this is possible for existing off-policy TD-learning algorithms such as Retrace($\lambda$) [15], $Q^\pi(\lambda)$ [13], $V$-trace [25], and $Q$-trace. That is the reason why none of them were shown to converge in the function approximation setting. In contrast, consider the $\lambda$-averaged $Q$-trace algorithm. Both $D_c$ and $D_\rho$ are identity matrices (which implies $D_{\rho,\max} = D_{c,\min} = 1$), hence Condition 9.3.1 (3) is always satisfied. Similarly, in the two-sided $Q$-trace algorithm, for any choice of the upper truncation level $u \geq \mathbf{1}$, we can always choose the lower truncation level $\mathbf{0} \leq \ell \leq \mathbf{1}$ appropriately to satisfy Condition 9.3.1 (3). Specifically, for any $s \in \mathcal{S}$ and $u(s) \geq 1$, choosing $\ell(s) \leq 1$ such that $\sum_{a\in\mathcal{A}} \pi_b(a|s) g_{\ell(s),u(s)}(\pi(a|s)/\pi_b(a|s)) = 1$ satisfies Condition 9.3.1 (3). Therefore, compared to $V$-trace, Retrace($\lambda$), and $Q$-trace, where the importance sampling ratios were only truncated above, the primary reason for introducing the lower truncation level is to satisfy Condition 9.3.1 (3), thereby ensuring convergence of the resulting two-sided $Q$-trace algorithm.

In the next lemma, we show that under Condition 9.3.1, with properly chosen $n$, the composed operator $\text{Proj}_{\mathcal{Q}}\mathcal{B}_{c,\rho}(\cdot)$ is a contraction mapping, which ensures that Equation 9.2 has a unique solution, denoted by $w_{c,\rho}^\pi$. Moreover, we provide performance guarantees on the solution $w_{c,\rho}^\pi$ in terms of an upper bound on the difference between $Q^\pi$ and $\Phi w_{c,\rho}^\pi$. Let $Q_{c,\rho}^\pi$ be the solution of generalized Bellman equation $Q = \mathcal{B}_{c,\rho}(Q)$, which is guaranteed to exist and is unique since $\mathcal{B}_{c,\rho}(\cdot)$ itself is a contraction mapping under Condition 9.3.1 (1) and (2) [138].

**Lemma 9.3.3.** *Under Condition 9.3.1, suppose that the parameter $n$ is chosen such that $\gamma_c := \tilde{\gamma}(n)/\sqrt{\mathcal{K}_{SA,\min}} < 1$. Then the composed operator $\text{Proj}_{\mathcal{Q}}\mathcal{B}_{c,\rho}(\cdot)$ is a $\gamma_c$-contraction mapping with respect to $\|\cdot\|_{\mathcal{K}_{SA}}$. In this case, the unique solution $w_{c,\rho}^\pi$ of the generalized PBE (cf. Equation 9.2) satisfies*

$$\begin{aligned}\|Q^\pi - \Phi w_{c,\rho}^\pi\|_{\mathcal{K}_{SA}} \leq\ & \frac{1}{\sqrt{1-\gamma_c^2}}\|Q_{c,\rho}^\pi - \textit{Proj}_{\mathcal{Q}} Q_{c,\rho}^\pi\|_{\mathcal{K}_{SA}} \\ & + \frac{\gamma \max_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} |\pi(a|s) - \pi_b(a|s)\rho(s,a)|}{(1-\gamma)(1-\gamma D_{\rho,\max})}. \end{aligned} \tag{9.3}$$

The first term on the RHS of Equation 9.3 captures the error due to function approximation, which is in the same spirit to Theorem 1 (4) of the seminal paper [92], and vanishes in the tabular setting. The second term on the RHS of Equation 9.3 arises because of the use of generalized importance sampling ratios, which is introduced to overcome the high variance in off-policy learning. Note that the second term vanishes when $\rho(s,a) = \pi(a|s)/\pi_b(a|s)$ for all $(s,a)$, which corresponds to choosing $\lambda = \mathbf{1}$ in $\lambda$-averaged $Q$-trace and choosing $\ell(s) \leq \min_{s,a} \pi(a|s)/\pi_b(a|s)$ and $u(s) \geq \max_{s,a} \pi(a|s)/\pi_b(a|s)$ for all $s$ in two-sided $Q$-trace. However, in these cases, the cumulative product of importance sampling ratios leads to a high variance in Algorithm 7. The trade-off between the variance and the bias in $w_{c,\rho}^{\pi}$ (i.e., second term on the RHS of Equation 9.3) will be elaborated in detail in the next section.

## 9.4 Finite-Sample Analysis

With the contraction property of the generalized PBE established, the almost sure convergence of Algorithm 7 under mild conditions directly follows from standard SA results in the literature [11, 33]. In this section, we take a step further and perform finite-sample analysis of Algorithm 7. For ease of exposition, we here only present the finite-sample bounds of $\lambda$-averaged $Q$-trace and two-sided $Q$-trace, where $c(\cdot,\cdot)$ and $\rho(\cdot,\cdot)$ are explicitly specified.

For any $\delta > 0$, let $t_\delta = \min\{k \geq 0 : \max_{s\in\mathcal{S}} \|P_{\pi_b}^k(s,\cdot) - \kappa_S(\cdot)\|_{\text{TV}} \leq \delta\}$ be the mixing time of the Markov chain $\{S_k\}$ under $\pi_b$ with precision $\delta$. Note that Assumption 9.3.1 implies that $t_\delta = \mathcal{O}(\log(1/\delta))$. Let $\lambda_{\min}$ be the mininum eigenvalue of the positive definite matrix $\Phi^\top \mathcal{K}_{SA}\Phi$. Let $L = 1 + (\gamma\rho_{\max})^n$, where $\rho_{\max} = \max_{s,a} \rho(s,a)$.

We next present finite-sample guarantees of the $\lambda$-averaged $Q$-trace algorithm when using constant stepsize (i.e., $\alpha_k \equiv \alpha$). The results for using diminishing stepsizes are trivial extensions.

**Theorem 9.4.1.** *Consider $\{w_k\}$ of the $\lambda$-averaged $Q$-trace Algorithm. Suppose that (1) Assumptions 9.1.1 and 9.3.1 are satisfied, (2) $\lambda \in [\mathbf{0}, \mathbf{1}]$, (3) the parameter $n$ is chosen such* *that* $\gamma_c := \gamma^n/\sqrt{\mathcal{K}_{SA,\min}} < 1$*, and (4) the stepsize* $\alpha$ *is chosen such that* $\alpha(t_\alpha + n + 1) \leq \frac{(1-\gamma_c)\lambda_{\min}}{130L^2}$*. Then, we have for all* $k \geq t_\alpha + n + 1$ *that*

$$\mathbb{E}[\|w_k - w_{c,\rho}^\pi\|_2^2] \leq c_1(1 - (1-\gamma_c)\lambda_{\min}\alpha)^{k-(t_\alpha+n+1)} + c_2\frac{\alpha L^2(t_\alpha + n + 1)}{(1-\gamma_c)\lambda_{\min}}, \tag{9.4}$$

*where* $c_1 = (\|w_0\|_2 + \|w_0 - w_{c,\rho}^\pi\|_2 + 1)^2$ *and* $c_2 = 130(\|w_{c,\rho}^\pi\|_2 + 1)^2$*. Moreover, we have*

$$\begin{aligned}\|Q^\pi - \Phi w_{c,\rho}^\pi\|_{\mathcal{K}_{SA}} \leq{}& \frac{1}{\sqrt{1-\gamma_c^2}}\|Q_{c,\rho}^\pi - \textit{Proj}_{\mathcal{Q}} Q_{c,\rho}^\pi\|_{\mathcal{K}_{SA}} \\ &+ \frac{\gamma \max_{s\in\mathcal{S}}(1-\lambda(s))\|\pi(\cdot|s) - \pi_b(\cdot|s)\|_1}{(1-\gamma)^2}.\end{aligned} \tag{9.5}$$

Using the common terminology in SA literature, we call the first term on the RHS of Equation 9.4 *convergence bias*, and the second term *variance*. When constant stepsize is used, the convergence bias goes to zero at a geometric rate while the variance is a constant roughly proportional to $\alpha t_\alpha$. Since $\lim_{\alpha\to 0} \alpha t_\alpha = 0$ under Assumption 9.3.1, the variance can be made arbitrarily small by using small $\alpha$.

The parameter $L = 1 + (\gamma\rho_{\max})^n$ plays an important role in the finite-sample bound. In fact, $L$ appears quadratically in the variance term of Equation 9.4, and captures the impact of the cumulative product of the importance sampling ratios. To overcome the high variance in off-policy learning (i.e., to make sure that the parameter $L = 1 + (\gamma\rho_{\max})^n$ does not grow exponentially fast with respect to $n$), we choose $\lambda \in \mathbb{R}^{|\mathcal{S}|}$ such that $\rho_{\max} = \max_s \lambda(s)(\max_a \pi(a|s)/\pi_b(a|s) - 1) + 1 \leq 1/\gamma$. However, as long as $\lambda \neq \mathbf{1}$, the limit point of the $\lambda$-averaged $Q$-trace algorithm involves an additional bias term (i.e., the second term on the RHS of Equation 9.5) that does not vanish even in the tabular setting.

In light of the discussion above, it is clear that there is a trade-off between the variance (cf. second term on the RHS of Equation 9.4) and the bias in the limit point (cf. the second term on the RHS of Equation 9.3) in choosing the parameter $\lambda$. Specifically, large $\lambda$ leads to large $\rho_{\max}$ and hence large $L$ and large variance, but in this case the second term on the

RHS of Equation 9.3 is smaller, implying that we have a smaller bias in the limit point.

Next, we present the finite-sample bounds of the two-sided $Q$-trace algorithm.

**Theorem 9.4.2.** *Consider $\{w_k\}$ of the two-sided $Q$-trace Algorithm. Suppose that (1) Assumptions 9.1.1 and 9.3.1 are satisfied, (2) the upper and lower truncation levels $\ell, u \in \mathbb{R}^{|\mathcal{S}|}$ are chosen such that $\sum_{a\in\mathcal{A}} \pi_b(a|s) g_{\ell(s),u(s)}(\pi(a|s)/\pi_b(a|s)) = 1$ for all $s$, (3) the parameter $n$ is chosen such that $\gamma_c := \gamma^n/\sqrt{\mathcal{K}_{SA,\min}} < 1$, and (4) the stepsize $\alpha$ is chosen such that $\alpha(t_\alpha + n + 1) \leq \frac{(1-\gamma_c)\lambda_{\min}}{130L^2}$. Then, we have for all $k \geq t_\alpha + n + 1$ that*

$$\mathbb{E}[\|w_k - w_{c,\rho}^\pi\|_2^2] \leq c_1(1-(1-\gamma_c)\lambda_{\min}\alpha)^{k-(t_\alpha+n+1)} + c_2\frac{\alpha L^2(t_\alpha+n+1)}{(1-\gamma_c)\lambda_{\min}}, \tag{9.6}$$

*where $c_1 = (\|w_0\|_2 + \|w_0 - w_{c,\rho}^\pi\|_2 + 1)^2$ and $c_2 = 130(\|w_{c,\rho}^\pi\|_2 + 1)^2$. Moreover, we have*

$$\begin{aligned}\|Q^\pi - \Phi w_{c,\rho}^\pi\|_{\mathcal{K}_{SA}} \leq\;& \frac{1}{\sqrt{1-\gamma_c^2}}\|Q_{c,\rho}^\pi - \Phi w_{c,\rho}^\pi\|_{\mathcal{K}_{SA}} \\ &+ \frac{\gamma \max_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}}(u_{\pi,\pi_b}(s,a) - \ell_{\pi,\pi_b}(s,a))}{(1-\gamma)^2},\end{aligned} \tag{9.7}$$

*where $u_{\pi,\pi_b}(s,a) = \max(\pi(a|s) - \pi_b(a|s)u(s), 0)$ and $\ell_{\pi,\pi_b}(s,a) = \min(\pi(a|s) - \pi_b(a|s)\ell(s), 0)$ for all $(s,a)$.*

The finite-sample bound of the two-sided $Q$-trace algorithm is qualitatively similar to that of the $\lambda$-averaged $Q$-trace algorithm. To overcome the high variance issue in off-policy learning, we choose the upper truncation level such that $\gamma u(s) \leq 1$ for all $s$, which ensures that the parameter $L = 1 + (\gamma\rho_{\max})^n \leq 1 + (\gamma \max_s u(s))^n$ does not grow exponentially with respect to $n$. Then we choose the lower truncation level accordingly to satisfy requirement (2) stated in Theorem 9.4.2. However, as long as there exists $s \in \mathcal{S}$ such that $u(s) < \max_{s,a} \pi(a|s)/\pi_b(a|s)$ or $\ell(s) > \min_{s,a} \pi(a|s)/\pi_b(a|s)$, the second term on the RHS of Equation 9.7 is in general non-zero, hence adding an additional bias term to the limit point even in the tabular setting. As a result, the trade-off between the variance and the bias in the limit point is also present in the two-sided $Q$-trace algorithm.

In view of Theorem 9.4.1 and Theorem 9.4.2, one limitation of this work is that the choice of $n$ to make $\gamma_c < 1$ depends on the unknown parameter $\mathcal{K}_{SA,\min}$ of the problem. In practice, one can start with a specific choice of $n$ and then gradually tune $n$ to achieve the convergence of the $\lambda$-averaged $Q$-trace algorithm or the two-sided $Q$-trace algorithm.

## 9.5 Proof Sketch of Theorem 9.4.1 and Theorem 9.4.2

Instead of proving Theorem 9.4.1 and Theorem 9.4.2, we will state and prove finite-sample bounds for Algorithm 7 with $c(\cdot,\cdot)$ and $\rho(\cdot,\cdot)$ satisfying Condition 9.3.1, which subsumes Theorem 9.4.1 and Theorem 9.4.2 as its special cases. In this more general setup where we do not have $c(\cdot,\cdot) = \rho(\cdot,\cdot)$, we define the constant parameter $L$ as

$$L = \begin{cases} (1 + (\gamma\rho_{\max})^n), & c(\cdot,\cdot) = \rho(\cdot,\cdot), \\ (1 + \gamma\rho_{\max}) f_n(\gamma c_{\max}), & c(\cdot,\cdot) \neq \rho(\cdot,\cdot), \end{cases} \tag{9.8}$$

where $c_{\max} = \max_{s,a} c(s,a)$ and $\rho_{\max} = \max_{s,a} \rho(s,a)$.

**Theorem 9.5.1.** *Consider $\{w_k\}$ of Algorithm Algorithm 7. Suppose that (1) Assumptions 9.1.1 and 9.3.1 are satisfied, (2) the generalized importance sampling ratios satisfy Condition 9.3.1, (3) the parameter $n$ is chosen such that $\gamma_c := \tilde{\gamma}(n)/\sqrt{\mathcal{K}_{SA,\min}} < 1$, and (4) the constant stepsize $\alpha$ is chosen such that $\alpha(t_\alpha + n + 1) \leq \frac{(1-\gamma_c)\lambda_{\min}}{130L^2}$. Then, we have for all $k \geq t_\alpha + n + 1$:*

$$\mathbb{E}[\|w_k - w_{c,\rho}^\pi\|_2^2] \leq c_1(1 - (1-\gamma_c)\lambda_{\min}\alpha)^{k-(t_\alpha+n+1)} + c_2 L^2 \frac{\alpha(t_\alpha + n + 1)}{(1-\gamma_c)\lambda_{\min}},$$

*where $c_1 = (\|w_0\|_2 + \|w_0 - w_{c,\rho}^\pi\|_2 + 1)^2$ and $c_2 = 130(\|w_{c,\rho}^\pi\|_2 + 1)^2$.*

To prove Theorem 9.5.1, we first rewrite Algorithm 7 as an SA algorithm. Let $\{X_k\}$ be a finite-state Markov chain defined by $X_k = (S_k, A_k, ..., S_{k+n}, A_{k+n})$ for any $k \geq 0$. Denote the state-space of $\{X_k\}$ by $\mathcal{X}$. It is clear that under Assumption 9.3.1, the Markov chain $\{X_k\}$ also admits a unique stationary distribution, which we denote by $\nu \in \Delta^{|\mathcal{X}|}$. Let $F : \mathbb{R}^d \times \mathcal{X} \mapsto \mathbb{R}^d$ be an operator defined by $F(w, x) = \phi(s_0, a_0) \sum_{i=0}^{n-1} \gamma^i c_{1,i} \Delta_i(w)$ for any $w \in \mathbb{R}^d$ and $x = (s_0, a_0, ..., s_n, a_n) \in \mathcal{X}$. Let $\bar{F} : \mathbb{R}^d \mapsto \mathbb{R}^d$ be the "expected" operator of $F(\cdot, \cdot)$ defined by $\bar{F}(w) = \mathbb{E}_{X \sim \nu}[F(w, X)]$. Using the notation above, the update equation (line 4) of Algorithm 7 can be compactly written as

$$w_{k+1} = w_k + \alpha_k F(w_k, X_k), \tag{9.9}$$

which is an SA algorithm for solving the equation $\bar{F}(w) = 0$ with Markovian noise. Note that $\bar{F}(w) = 0$ is equivalent to the generalized PBE (cf. Lemma 9.3.1). We next establish the properties of the operators $F(\cdot, \cdot)$, $\bar{F}(\cdot)$, and the Markov chain $\{X_k\}$ in the following proposition, which enables us to use our SA results in Chapter 3 to derive finite-sample bounds of Algorithm 7.

**Proposition 9.5.1.** *The following statements hold.*

*(1)* $\|F(w_1, x) - F(w_2, x)\|_2 \leq L\|w_1 - w_2\|_2$ *for any* $w_1, w_2 \in \mathbb{R}^d$ *and* $x \in \mathcal{X}$*, and* $\|F(\mathbf{0}, x)\|_2 \leq f_n(\gamma c_{\max})$ *for any* $x \in \mathcal{X}$*,*

*(2)* $\max_{x \in \mathcal{X}} \left\| P_X^{k+n+1}(x, \cdot) - \nu(\cdot) \right\|_{TV} \leq C\sigma^k$ *for all* $k \geq 0$*, where* $P_X$ *is the transition probability matrix of the Markov chain* $\{X_k\}$ *under policy* $\pi_b$*,*

*(3)* $(w - w_{c,\rho}^\pi)^\top \bar{F}(w) \leq -(1 - \gamma_c)\lambda_{\min}\|w - w_{c,\rho}^\pi\|_2^2$ *for any* $w \in \mathbb{R}^d$*.*

Proposition 9.5.1 (1) establishes the Lipschitz continuity of the operator $F(\cdot, \cdot)$, Proposition 9.5.1 (2) establishes the geometric mixing of the auxiliary Markov chain $\{X_k\}$, and Proposition 9.5.1 (3) essentially guarantees that the ODE $\dot{x}(t) = \bar{F}(x(t))$ associated with SA algorithm (Equation 9.9) is globally geometrically stable. The rest of the proof follows by applying Theorem 3.2.1 to Algorithm 7.

## 9.6 Proof of All Theoretical Results

### 9.6.1 Proof of Lemma 9.3.1

We begin by introducing some notation. Let $\pi_c$ and $\pi_\rho$ be two policies defined by

$$\pi_c(a|s) = \frac{\pi_b(a|s)c(s,a)}{\sum_{a'\in\mathcal{A}}\pi_b(a'|s)c(s,a')}, \quad \text{and} \quad \pi_\rho(a|s) = \frac{\pi_b(a|s)\rho(s,a)}{\sum_{a'\in\mathcal{A}}\pi_b(a'|s)\rho(s,a')}, \ \forall\,(s,a).$$

Let $P_{\pi_c}$ and $P_{\pi_\rho}$ be the transition probability matrices of the Markov chain $\{S_k\}$ induced by the policies $\pi_c$ and $\pi_\rho$, respectively. Then, Equation 9.1 can be compactly written in vector form as

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

where $R\in\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ is defined by $R(s,a)=\mathcal{R}(s,a)$ for all $(s,a)$. Observe that the above equation is further equivalent to

$$\Phi(\Phi^\top\mathcal{K}_{SA}\Phi)^{-1}\Phi^\top \mathcal{K}_{SA}\sum_{i=0}^{n-1}(\gamma P_{\pi_c}D_c)^i(R+\gamma P_{\pi_\rho}D_\rho\Phi w-\Phi w)=0. \tag{9.10}$$

To see this, since the matrix $\Phi$ has full column-rank, and the matrix $\Phi^\top\mathcal{K}_{SA}\Phi$ is positive definite and hence invertible, we have $x=0$ if and only if $\Phi(\Phi^\top\mathcal{K}_{SA}\Phi)^{-1}x=0$.

To rewrite Equation 9.10 in the desired form of the generalized PBE (cf. Equation 9.2), we use the following three observations.

(1) The projection operator $\text{Proj}_\mathcal{Q}(\cdot)$ is explicitly given by

$$\text{Proj}_\mathcal{Q}(\cdot)=\Phi(\Phi^\top\mathcal{K}_{SA}\Phi)^{-1}\Phi^\top\mathcal{K}_{SA}(\cdot),$$

(2) The operator $\mathcal{T}_c(\cdot)$ is explicitly given by $\mathcal{T}_c(\cdot)=\sum_{i=0}^{n-1}(\gamma P_{\pi_c}D_c)^i(\cdot)$,

(3) The operator $\mathcal{H}_\rho(\cdot)$ is explicitly given by $\mathcal{H}_\rho(\cdot)=R+\gamma P_{\pi_\rho}D_\rho(\cdot)$.

Therefore, Equation 9.10 is equivalent to

$$\text{Proj}_{\mathcal{Q}}[\mathcal{T}_c(\mathcal{H}_\rho(\Phi w) - \Phi w)] = 0. \tag{9.11}$$

Finally, adding and subtracting $\Phi w$ on both sides of the previous inequality and we obtain the desired generalized PBE:

$$\begin{aligned}
\Phi w &= \text{Proj}_{\mathcal{Q}}[\mathcal{T}_c(\mathcal{H}_\rho(\Phi w) - \Phi w)] + \Phi w \\
&= \text{Proj}_{\mathcal{Q}}[\mathcal{T}_c(\mathcal{H}_\rho(\Phi w) - \Phi w) + \Phi w] \\
&= \text{Proj}_{\mathcal{Q}}\mathcal{B}_{c,\rho}(\Phi w),
\end{aligned}$$

where the second equality follows from (1) $\Phi w \in \mathcal{Q}$ and (2) $\text{Proj}_{\mathcal{Q}}(\cdot)$ is a linear operator.

### 9.6.2 Proof of Lemma 9.3.2

For any $Q_1, Q_2 \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, we have

$$\begin{aligned}
\|\text{Proj}_{\mathcal{Q}}\mathcal{B}_{c,\rho}(Q_1) - \text{Proj}_{\mathcal{Q}}\mathcal{B}_{c,\rho}(Q_2)\|_{\mathcal{K}_{SA}} &\leq \|\mathcal{B}_{c,\rho}(Q_1) - \mathcal{B}_{c,\rho}(Q_2)\|_{\mathcal{K}_{SA}} \\
&\leq \|\mathcal{B}_{c,\rho}(Q_1) - \mathcal{B}_{c,\rho}(Q_2)\|_\infty \quad (\|\cdot\|_{\mathcal{K}_{SA}} \leq \|\cdot\|_\infty) \\
&\leq \tilde{\gamma}(n)\|Q_1 - Q_2\|_\infty \\
&\leq \frac{\tilde{\gamma}(n)}{\sqrt{\mathcal{K}_{SA,\min}}}\|Q_1 - Q_2\|_{\mathcal{K}_{SA}}, \\
&\qquad\qquad (\|\cdot\|_\infty \leq \frac{1}{\sqrt{\mathcal{K}_{SA,\min}}}\|\cdot\|_{\mathcal{K}_{SA}})
\end{aligned}$$

where the first inequality follows from $\text{Proj}_{\mathcal{Q}}$ being non-expansive with respect to $\|\cdot\|_{\mathcal{K}_{SA}}$, and the third inequality follows from $\mathcal{B}_{c,\rho}(\cdot)$ being a $\tilde{\gamma}(n)$-contraction operator with respect to $\|\cdot\|_\infty$ [138] [1].

[1] In Chapter 7, we work with an asynchronous variant of the generalized Bellman operator, which is shown to be a contraction mapping with respect to $\|\cdot\|_\infty$ with contraction factor $1 - \mathcal{K}_{SA,\min}f_n(\gamma D_{c,\min})(1 - \gamma D_{\rho,\max})$. In this paper we work with the synchronous generalized Bellman operator $\mathcal{B}_{c,\rho}(\cdot)$. In this case, one can easily verify that the corresponding contraction factor can be obtained by simply dropping the factor

### 9.6.3 Proof of Lemma 9.3.3

We first show that under Condition 9.3.1 (3), we have $\lim_{n\rightarrow\infty}\tilde{\gamma}(n)/\sqrt{\mathcal{K}_{SA,\min}}<1$. Using the explicit expression of $\tilde{\gamma}(n)$, we have

$$\begin{aligned}
\lim_{n\rightarrow\infty}\frac{\tilde{\gamma}(n)}{\sqrt{\mathcal{K}_{SA,\min}}} &= \lim_{n\rightarrow\infty}\frac{1-f_n(\gamma D_{c,\min})(1-\gamma D_{\rho,\max})}{\sqrt{\mathcal{K}_{SA,\min}}}\\
&= \lim_{n\rightarrow\infty}\frac{1-\frac{1-(\gamma D_{c,\min})^n}{1-\gamma D_{c,\min}}(1-\gamma D_{\rho,\max})}{\sqrt{\mathcal{K}_{SA,\min}}}\\
&\qquad\qquad (f_n(x)=\textstyle\sum_{i=0}^{n-1}x^i \text{ and } \gamma D_{c,\min}<1)\\
&= \frac{\gamma(D_{\rho,\max}-D_{c,\min})}{(1-\gamma D_{c,\min})\sqrt{\mathcal{K}_{SA,\min}}}\\
&< 1. \qquad\qquad \text{(Condition 9.3.1 (3))}
\end{aligned}$$

Therefore, when $n$ is chosen such that $\gamma_c=\frac{\tilde{\gamma}(n)}{\sqrt{\mathcal{K}_{SA,\min}}}<1$, we have by Lemma 9.3.2 that

$$\|\text{Proj}_{\mathcal{Q}}\mathcal{B}_{c,\rho}(Q_1)-\text{Proj}_{\mathcal{Q}}\leq\gamma_c\|Q_1-Q_2\|_{\mathcal{K}_{SA}},\;\forall\,Q_1,Q_2\in\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}.$$

It follows that the composed operator $\text{Proj}_{\mathcal{Q}}\mathcal{B}_{c,\rho}(\cdot)$ is a contraction mapping with respect to $\|\cdot\|_{\mathcal{K}_{SA}}$, with contraction factor $\gamma_c$.

Next consider the difference between $Q^\pi$ and $\Phi w_{c,\rho}^\pi$. First of all, we have by triangle inequality that

$$\begin{aligned}
\|Q^\pi-\Phi w_{c,\rho}^\pi\|_{\mathcal{K}_{SA}} &= \|Q^\pi-Q_{c,\rho}^\pi+Q_{c,\rho}^\pi-\Phi w_{c,\rho}^\pi\|_{\mathcal{K}_{SA}}\\
&\leq \|Q^\pi-Q_{c,\rho}^\pi\|_{\mathcal{K}_{SA}}+\|Q_{c,\rho}^\pi-\Phi w_{c,\rho}^\pi\|_{\mathcal{K}_{SA}}. \qquad (9.12)
\end{aligned}$$

We next bound each term on the RHS of the previous inequality. For the first term, it was

$\mathcal{K}_{SA,\min}$.

already established in Proposition 2.1 of [138] that

$$\|Q^\pi - Q^\pi_{c,\rho}\|_{\mathcal{K}_{SA}} \leq \|Q^\pi - Q^\pi_{c,\rho}\|_\infty \leq \frac{\gamma \max_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} |\pi(a|s) - \pi_b(a|s)\rho(s,a)|}{(1-\gamma)(1-\gamma D_{\rho,\max})}. \quad (9.13)$$

Now consider the second term on the RHS of Equation 9.12. First note that

$$\begin{aligned}
\|Q^\pi_{c,\rho} - \Phi w^\pi_{c,\rho}\|^2_{\mathcal{K}_{SA}} &= \|Q^\pi_{c,\rho} - \text{Proj}_{\mathcal{Q}} Q^\pi_{c,\rho} + \text{Proj}_{\mathcal{Q}} Q^\pi_{c,\rho} - \Phi w^\pi_{c,\rho}\|^2_{\mathcal{K}_{SA}} \\
&= \|Q^\pi_{c,\rho} - \text{Proj}_{\mathcal{Q}} Q^\pi_{c,\rho}\|^2_{\mathcal{K}_{SA}} + \|\text{Proj}_{\mathcal{Q}} Q^\pi_{c,\rho} - \Phi w^\pi_{c,\rho}\|^2_{\mathcal{K}_{SA}} \qquad (*) \\
&= \|Q^\pi_{c,\rho} - \text{Proj}_{\mathcal{Q}} Q^\pi_{c,\rho}\|^2_{\mathcal{K}_{SA}} \\
&\quad + \|\text{Proj}_{\mathcal{Q}} \mathcal{B}_{c,\rho}(Q^\pi_{c,\rho}) - \text{Proj}_{\mathcal{Q}} \mathcal{B}_{c,\rho}(\Phi w^\pi_{c,\rho})\|^2_{\mathcal{K}_{SA}} \\
&\leq \|Q^\pi_{c,\rho} - \text{Proj}_{\mathcal{Q}} Q^\pi_{c,\rho}\|^2_{\mathcal{K}_{SA}} + \gamma_c^2 \|Q^\pi_{c,\rho} - \Phi w^\pi_{c,\rho}\|^2_{\mathcal{K}_{SA}},
\end{aligned}$$

where Eq. $(*)$ follows from the Babylonian–Pythagorean theorem (i.e., $Q^\pi_{c,\rho} - \text{Proj}_{\mathcal{Q}} Q^\pi_{c,\rho} \perp \mathcal{Q}$ and $\text{Proj}_{\mathcal{Q}} Q^\pi_{c,\rho} - \Phi w^\pi_{c,\rho} \in \mathcal{Q}$). Rearrange the previous inequality and we have

$$\|Q^\pi_{c,\rho} - \Phi w^\pi_{c,\rho}\|_{\mathcal{K}_{SA}} \leq \frac{1}{\sqrt{1-\gamma_c^2}} \|Q^\pi_{c,\rho} - \text{Proj}_{\mathcal{Q}} Q^\pi_{c,\rho}\|_{\mathcal{K}_{SA}}. \quad (9.14)$$

Substituting Equation 9.13 and Equation 9.14 into the RHS of Equation 9.12 and we finally obtain

$$\begin{aligned}
\|Q^\pi - \Phi w^\pi_{c,\rho}\|_{\mathcal{K}_{SA}} \leq\ & \frac{\gamma \max_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} |\pi(a|s) - \pi_b(a|s)\rho(s,a)|}{(1-\gamma)(1-\gamma D_{\rho,\max})} \\
&+ \frac{1}{\sqrt{1-\gamma_c^2}} \|Q^\pi_{c,\rho} - \text{Proj}_{\mathcal{Q}} Q^\pi_{c,\rho}\|_{\mathcal{K}_{SA}}.
\end{aligned}$$

### 9.6.4 Proof of Proposition 9.5.1

(1) (a) We first rewrite the operator $F(\cdot,\cdot)$ in the following equivalent way. For any $w \in \mathbb{R}^d$ and $x = (s_0, a_0, ..., s_n, a_n) \in \mathcal{X}$, we have

$$\begin{aligned}
F(w,x) = {} & \phi(s_0,a_0)\sum_{i=0}^{n-1}\gamma^i\prod_{j=1}^{i}c(s_j,a_j)\times \\
& (\mathcal{R}(s_i,a_i)+\gamma\rho(s_{i+1},a_{i+1})\phi(s_{i+1},a_{i+1})^\top w-\phi(s_i,a_i)^\top w) \\
= {} & \phi(s_0,a_0)\sum_{i=0}^{n-1}\gamma^i\prod_{j=1}^{i}c(s_j,a_j)\mathcal{R}(s_i,a_i)-\phi(s_0,a_0)\sum_{i=0}^{n-1}\gamma^i\prod_{j=1}^{i}c(s_j,a_j)\phi(s_i,a_i)^\top w \\
& +\phi(s_0,a_0)\sum_{i=0}^{n-1}\gamma^{i+1}\prod_{j=1}^{i}c(s_j,a_j)\rho(s_{i+1},a_{i+1})\phi(s_{i+1},a_{i+1})^\top w \\
= {} & \phi(s_0,a_0)\sum_{i=0}^{n-1}\gamma^i\prod_{j=1}^{i}c(s_j,a_j)\mathcal{R}(s_i,a_i)-\phi(s_0,a_0)\sum_{i=0}^{n-1}\gamma^i\prod_{j=1}^{i}c(s_j,a_j)\phi(s_i,a_i)^\top w \\
& +\phi(s_0,a_0)\sum_{i=1}^{n}\gamma^{i}\prod_{j=1}^{i-1}c(s_j,a_j)\rho(s_i,a_i)\phi(s_i,a_i)^\top w \\
= {} & \phi(s_0,a_0)\sum_{i=0}^{n-1}\gamma^i\prod_{j=1}^{i}c(s_j,a_j)\mathcal{R}(s_i,a_i)-\phi(s_0,a_0)\phi(s_0,a_0)^\top w \\
& +\phi(s_0,a_0)\sum_{i=1}^{n-1}\gamma^{i}\prod_{j=1}^{i-1}c(s_j,a_j)(\rho(s_i,a_i)-c(s_i,a_i))\phi(s_i,a_i)^\top w \\
& +\phi(s_0,a_0)\gamma^n\prod_{j=1}^{n-1}c(s_j,a_j)\rho(s_n,a_n)\phi(s_n,a_n)^\top w.
\end{aligned}$$

We now proceed and show the Lipschitz property.

For any $w_1, w_2 \in \mathbb{R}^d$ and $x = (s_0, a_0, ..., s_n, a_n) \in \mathcal{X}$, using the fact that $\|\phi(s,a)\|_2 \leq \|\phi(s,a)\|_1 \leq \|\Phi\|_\infty \leq 1$, we have

$$\begin{aligned}
& \|F(w_1,x)-F(w_2,x)\|_2 \\
\leq {} & \|\phi(s_0,a_0)\phi(s_0,a_0)^\top(w_1-w_2)\|_2 \\
& +\left\|\phi(s_0,a_0)\sum_{i=1}^{n-1}\gamma^i\prod_{j=1}^{i-1}c(s_j,a_j)(\rho(s_i,a_i)-c(s_i,a_i))\phi(s_i,a_i)^\top(w_1-w_2)\right\|_2
\end{aligned}$$

$$
\begin{aligned}
&+\left\|\phi(s_0,a_0)\gamma^n\prod_{j=1}^{n-1}c(s_j,a_j)\rho(s_n,a_n)\phi(s_n,a_n)^\top(w_1-w_2)\right\|_2\\
\leq\;& \|w_1-w_2\|_2+\sum_{i=1}^{n-1}\gamma^i c_{\max}^{i-1}\max_{s,a}|\rho(s,a)-c(s,a)|\|w_1-w_2\|_2\\
&+\gamma^n c_{\max}^{n-1}\rho_{\max}\|w_1-w_2\|_2\\
=\;&\left(1+\gamma\max_{s,a}|\rho(s,a)-c(s,a)|\frac{1-(\gamma c_{\max})^{n-1}}{1-\gamma c_{\max}}+\gamma^n c_{\max}^{n-1}\rho_{\max}\right)\|w_1-w_2\|_2\\
\leq\;&\begin{cases}(1+(\gamma\rho_{\max})^n)\|w_1-w_2\|_2, & c(\cdot,\cdot)=\rho(\cdot,\cdot)\\ (1+\gamma\rho_{\max})f_n(\gamma c_{\max})\|w_1-w_2\|_2, & c(\cdot,\cdot)\neq\rho(\cdot,\cdot).\end{cases}
\end{aligned}
$$

(1) (b) For any $x=(s_0,a_0,...,s_n,a_n)\in\mathcal{X}$, we have

$$
\|F(w,\mathbf{0})\|_2=\left\|\phi(s_0,a_0)\sum_{i=0}^{n-1}\gamma^i\prod_{j=1}^{i}c(s_j,a_j)\mathcal{R}(s_i,a_i)\right\|_2\leq\sum_{i=0}^{n-1}\gamma^i c_{\max}^i\leq f_n(\gamma c_{\max}).
$$

(2) It is clear that the stationary distribution $\nu$ of the Markov chain $\{X_k\}$ is given by

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

for all $(s_0,a_0,...,s_n,a_n)\in\mathcal{X}$. Moreover, for any $x=(s_0,a_0,...,s_n,a_n)\in\mathcal{X}$, we have for any $k\geq 0$ that

$$
\begin{aligned}
\left\|P_{\pi_b}^{k+n+1}(x,\cdot)-\nu(\cdot)\right\|_{\text{TV}}=\;&\frac{1}{2}\sum_{s_0',a_0',\cdots,s_n',a_n'}\left|\sum_s P_{a_n}(s_n,s)P_{\pi_b}^k(s,s_0')-\kappa_S(s_0')\right|\times\\
&\left[\prod_{i=0}^{n-1}\pi_b(a_i'\mid s_i')P_{a_i'}(s_i',s_{i+1}')\right]\pi_b(a_n'\mid s_n')\\
=\;&\frac{1}{2}\sum_{s_0'}\left|\sum_s P_{a_n}(s_n,s)P_{\pi_b}^k(s,s_0')-\kappa_S(s_0')\right|\\
\leq\;&\frac{1}{2}\sum_s P_{a_n}(s_n,s)\sum_{s_0'}\left|P_{\pi_b}^k(s,s_0')-\kappa_S(s_0')\right|\\
\leq\;&\max_{s\in\mathcal{S}}\|P_{\pi_b}^k(s,\cdot)-\kappa_S(\cdot)\|_{\text{TV}}
\end{aligned}
$$

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

Therefore, we have $\max_{x\in\mathcal{X}} \left\|P_{\pi_b}^{k+n+1}(x,\cdot)-\nu(\cdot)\right\|_{\text{TV}} \leq C\sigma^k$ for all $k\geq 0$.

(3) Using the fact that $\mathcal{B}_{c,\rho}(\cdot)$ is a linear operator, we have for any $w\in\mathbb{R}^d$ that

$$\begin{aligned}
&(w-w_{c,\rho}^\pi)^\top \bar{F}(w)\\
=\;&(w-w_{c,\rho}^\pi)^\top\Phi^\top\mathcal{K}_{SA}\left(\mathcal{B}_{c,\rho}(\Phi w)-\Phi w\right)\\
=\;&(w-w_{c,\rho}^\pi)^\top\Phi^\top\mathcal{K}_{SA}\left(\mathcal{B}_{c,\rho}(\Phi w)-\mathcal{B}_{c,\rho}(\Phi w_{c,\rho}^\pi)\right)-(w-w_{c,\rho}^\pi)^\top\Phi^\top\mathcal{K}_{SA}\Phi(w-w_{c,\rho}^\pi)\\
=\;&(w-w_{c,\rho}^\pi)^\top\Phi^\top\mathcal{K}_{SA}\Phi(\Phi^\top\mathcal{K}_{SA}\Phi)^{-1}\Phi^\top\mathcal{K}_{SA}\mathcal{B}_{c,\rho}(\Phi(w-w_{c,\rho}^\pi))\\
&-(w-w_{c,\rho}^\pi)^\top\Phi^\top\mathcal{K}_{SA}\Phi(w-w_{c,\rho}^\pi)\\
=\;&(w-w_{c,\rho}^\pi)^\top\Phi^\top\mathcal{K}_{SA}\Phi(\Phi^\top\mathcal{K}_{SA}\Phi)^{-1}\Phi^\top\mathcal{K}_{SA}\mathcal{B}_{c,\rho}(\Phi(w-w_{c,\rho}^\pi))\\
&-(w-w_{c,\rho}^\pi)^\top\Phi^\top\mathcal{K}_{SA}\Phi(w-w_{c,\rho}^\pi)\\
\leq\;&\|\Phi(w-w_{c,\rho}^\pi)\|_{\mathcal{K}_{SA}}\|\Phi(\Phi^\top\mathcal{K}_{SA}\Phi)^{-1}\Phi^\top\mathcal{K}_{SA}\mathcal{B}_{c,\rho}(\Phi(w-w_{c,\rho}^\pi))\|_{\mathcal{K}_{SA}}\\
&-\|\Phi(w-w_{c,\rho}^\pi)\|_{\mathcal{K}_{SA}}^2\\
=\;&\|\Phi(w-w_{c,\rho}^\pi)\|_{\mathcal{K}_{SA}}\|\text{Proj}_{\mathcal{Q}}\mathcal{B}_{c,\rho}(\Phi(w-w_{c,\rho}^\pi))\|_{\mathcal{K}_{SA}}-\|\Phi(w-w_{c,\rho}^\pi)\|_{\mathcal{K}_{SA}}^2\\
\leq\;&\gamma_c\|\Phi(w-w_{c,\rho}^\pi)\|_{\mathcal{K}_{SA}}\|\Phi(w-w_{c,\rho}^\pi)\|_{\mathcal{K}_{SA}}-\|\Phi(w-w_{c,\rho}^\pi)\|_{\mathcal{K}_{SA}}^2\\
=\;&-(1-\gamma_c)\|\Phi(w-w_{c,\rho}^\pi)\|_{\mathcal{K}_{SA}}^2\\
\leq\;&-(1-\gamma_c)\lambda_{\min}\|w-w_{c,\rho}^\pi\|_2^2
\end{aligned}$$

### 9.6.5 Proof of Theorem 9.4.1

The finite-sample bound (i.e., Equation 9.4) follows directly from Theorem 9.5.1. To show the performance bound (cf. Equation 9.5) on the limit point $w_{c,\rho}^\pi$, we apply Lemma 9.3.3 to the $\lambda$-averaged $Q$-trace algorithm. Note that when $c(s,a)=\rho(s,a)=\lambda(s)\frac{\pi(a|s)}{\pi_b(a|s)}+1-\lambda(s)$

for all $(s,a)$, we have for any $s\in\mathcal{S}$ that

$$\begin{aligned}\sum_{a\in\mathcal{A}}|\pi(a|s)-\pi_b(a|s)\rho(s,a)| &= (1-\lambda(s))\sum_{a\in\mathcal{A}}|\pi(a|s)-\pi_b(a|s)|\\ &= (1-\lambda(s))\|\pi(\cdot|s)-\pi_b(\cdot|s)\|_1.\end{aligned}$$

This proves the result.

### 9.6.6 Proof of Theorem 9.4.2

The finite-sample bound (i.e., Equation 9.6) follows directly from Theorem 9.5.1. To show the performance bound (cf. Equation 9.7) on the limit point $w_{c,\rho}^{\pi}$, we apply Lemma 9.3.3 to the two-sided $Q$-trace algorithm. Note that when $c(s,a)=\rho(s,a)=g_{\ell(s),u(s)}(\pi(a|s)/\pi_b(a|s))$ for all $(s,a)$, we have

$$\begin{aligned}\sum_{a\in\mathcal{A}}|\pi(a|s)-\pi_b(a|s)\rho(s,a)| =& \sum_{a\in\mathcal{A}}|(\pi(a|s)-\pi_b(a|s)\ell(s))\mathbb{I}\{\pi(a|s)<\ell(s)\pi_b(a|s)\}\\ &+(\pi(a|s)-\pi_b(a|s)u(s))\mathbb{I}\{\pi(a|s)>u(s)\pi_b(a|s)\}|\\ \leq& \sum_{a\in\mathcal{A}}|(\pi(a|s)-\pi_b(a|s)\ell(s))\mathbb{I}\{\pi(a|s)<\ell(s)\pi_b(a|s)\}|\\ &+\sum_{a\in\mathcal{A}}|(\pi(a|s)-\pi_b(a|s)u(s))\mathbb{I}\{\pi(a|s)>u(s)\pi_b(a|s)\}|\\ =& \sum_{a\in\mathcal{A}}\max(\pi(a|s)-\pi_b(a|s)u(s),0)\\ &-\sum_{a\in\mathcal{A}}\min(\pi(a|s)-\pi_b(a|s)\ell(s),0)\\ =& \sum_{a\in\mathcal{A}}(u_{\pi,\pi_b}(s,a)-\ell_{\pi,\pi_b}(s,a)).\end{aligned}$$

This proves the result.

## 9.7 Conclusion and Future Work

In this chapter, we focus on TD-learning with off-policy sampling and linear function approximation, and designed a convergent multi-step TD-learning algorithm. To overcome the high variance issue in off-policy learning, we propose using generalized importance sampling ratios. However, the variance reduction is achieved at a cost of an asymptotic bias. Therefore, a potential future direction of this line of work is to investigate whether variance reduction is possible without introducing bias.

# CHAPTER 10

# POLICY-BASED METHODS UNDER OFF-POLICY SAMPLING AND LINEAR FUNCTION APPROXIMATION

## 10.1 Introduction

So far we have been focusing on value-based RL algorithms, such as TD-learning for policy evaluation and $Q$-learning for control. In this chapter, we switch our focus to policy-based methods.

Unlike value-based methods, policy-based methods directly work with the policies, and in general consist of two phases: namely policy evaluation and policy improvement. Typical policy-based methods are approximate policy iteration and various actor-critic (AC) algorithms. Approximate policy iteration updates the policy by performing the $\arg\max$ operator to the latest $Q$-function estimate, while AC updates the policy using gradient ascent with preconditioning. Specifically, an identity pre-conditioner corresponds to regular AC, while a pre-conditioning with fisher information results in natural actor-critic (NAC) [144]. As for policy evaluation, it usually uses the TD-learning method and its variants, such as TD$(0)$, $n$-step TD [56], TD$(\lambda)$, or Monte Carlo method.

While at a high level, all policy-based methods iteratively perform policy evaluation and policy improvement, the actual implementation, however, has many variants. For example, policy-based algorithms can be implemented in a two-loop manner or a two time-scale manner. Depending on the sample collection procedure, there are on-policy learning and off-policy learning [145]. Also, policy-based algorithms can be incorporated with function approximation to overcome the curse of dimensionality in RL.

In this chapter, we focus on policy-based methods under off-policy sampling and linear function approximation, where the policy evaluation sub-problem is solved with the TD-learning algorithm studied in Chapter 9.

### 10.1.1 Main Contributions

We propose a general policy-based framework that uses linear function approximation and off-policy sampling. The framework subsumes popular existing algorithms such as approximate policy iteration and NAC as its special cases. We establish an overall $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity of the general policy-based method up to a function approximation error. This is the first time that $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity is established in the off-policy linear function approximation setting. Importantly, our results do not require strong exploration assumptions as in existing literature.

### 10.1.2 Related Literature

**Approximate Policy Iteration.** In the MDP setting (i.e., known environmental model), policy iteration is a popular method for finding an optimal policy [9], and is known to find an optimal policy in finitely many steps. In the RL setting, policy iteration becomes approximate policy iteration due to the possible error in solving the policy evaluation subproblem. See [155] for a survey about approximate policy iteration and its variants.

**On-Policy AC.** Several variants of AC were proposed in [156, 157, 158, 144, 159]. In the tabular setting, [160, 33, 157] studied the asymptotic convergence of AC algorithm. Furthermore, [146, 152] characterize the asymptotic convergence of on-policy AC under function approximation. Recently, there has been a flurry of work studying the finite-sample convergence of AC and NAC [161]. [162, 163, 137] perform the finite sample analysis of NAC under tabular setting, and [164, 148, 149, 165, 147, 166, 167, 168, 169] establish the finite-sample bounds of AC in function approximation setting. To the best of our knowledge, the best sample complexity bound of AC algorithms is provided in [163], where the authors characterize an $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity. However, [163] only studies tabular RL in the on-policy setting.

Table 10.1: Sample Complexity Bounds of the AC-Type Algorithms Using Function Approximation

| Algorithm | Sampling Procedure | References | Sample Complexity [1,2] | Single Trajectory |
|---|---|---|---|---|
| Actor Critic | On-Policy | [146] | Asymptotic | ✓ |
| | | [147] | $\tilde{\mathcal{O}}(\epsilon^{-6})$ | ✗ |
| | | [148, 149] | $\tilde{\mathcal{O}}(\epsilon^{-4})$ | ✗ |
| | Off-Policy | [150, 151] | Asymptotic | ✓ |
| Natural Actor Critic | On-Policy | [152] | Asymptotic | ✓ |
| | | [147] | $\tilde{\mathcal{O}}(\epsilon^{-14})$ | ✗ |
| | | [153] | $\tilde{\mathcal{O}}(\epsilon^{-6})$ | ✗ |
| | Off-Policy | [154] | $\tilde{\mathcal{O}}(\epsilon^{-4})$ | ✗ |
| | | This work | $\tilde{\mathcal{O}}(\epsilon^{-3})$ | ✓ |

[1] In this table, for the AC (respectively NAC) algorithms, sample complexity is the number of samples needed to find a policy $\pi$ such that $\mathbb{E}[\|\nabla V^\pi(\mu)\|^2] \leq \epsilon + \mathcal{E}_{\text{bias}}$ (respectively $\mathbb{E}[V^*(\mu) - V^\pi(\mu)] \leq \epsilon + \mathcal{E}_{bias}$), where $\mathcal{E}_{bias}$ is a non-vanishing error due to the function approximation.
[2] Here $\tilde{O}(\cdot)$ ignores all the logarithmic terms.

**Off-policy AC.** Off-policy AC, was first proposed in [145]. After that, there has been numerous extensions to that work such as DPG [170], DDPG [171], ACER [172], TD3 [173], IMPALA [25], ACE [174], etc. The asymptotic convergence of off-policy AC was established for Gradient-AC in [150], and for AC with emphasis in [151]. The first finite-sample bound of off-policy NAC was established in [175]. However, in [175] only tabular setting was studied. In the function approximation setting, [154] provided the finite sample analysis of a doubly robust off-policy AC. [176] also provided a convergence bound for off-policy AC, however their convergence bound does not involve a bound for the critic. A detailed comparison between our results and the related literature on off-policy AC-type algorithms with function approximation is presented in Table 10.1.

## 10.2 Policy Update Rules

We begin by presenting a generic policy-based algorithm in the following, where the policy evaluation sub-problem is solved with Algorithm 7. For simplicity of notation, for a given target policy $\pi$, behavior policy $\pi_b$, constant stepsize $\alpha$, initialization $w_0$, and samples $\{(S_k, A_k)\}_{0\leq k\leq K+n-1}$, we denote the output of Algorithm 7 after $K$ iterations by

$$w = \text{ALG}(w_0, \pi, \pi_b, \alpha, K, \{(S_k, A_k)\}_{0\leq k\leq K+n-1}).$$

**Algorithm 8** A Generic Policy-Based Algorithm

```
1: Input: Integers $T$, $K$, initial policy $\pi_0$, sample trajectory $\{(S_t, A_t)\}_{0\leq t\leq T(K+n)}$ collected under the behavior policy $\pi_b$.
2: for $t = 0, 1, \ldots, T-1$ do
3:     dataset $= \{(S_k, A_k)\}_{t(K+n)\leq k\leq (t+1)(K+n)-1}$
4:     $w_t = \text{ALG}(\mathbf{0}, \pi_t, \pi_b, \alpha, K, \text{dataset})$
5:     $\pi_{t+1} = G(\Phi w_t, \pi_t)$
6: end for
7: Output: $\pi_T$
```

Although Algorithm 8 is presented with a fixed behavior policy $\pi_b$, our results can be easily generalized to the case where the behavior policy is updated across $t$. The only requirement on the behavior policy is that it should enable the agent to sufficiently explore the state-action space. In Algorithm 8 line 5, the function $G(\cdot, \cdot)$ represents the policy update rule, which takes the current policy iterate $\pi_t$ and the $Q$-function estimate $\Phi w_t$ as inputs. Many existing policy update rules fit into this framework, as elaborated below.

**$1/\beta_1$-Greedy Update.** Let $\beta_1 \in [1, \infty]$ be a tunable parameter. For any $t \geq 0$ and state-action pair $(s, a)$, we update the policy by

$$\pi_{t+1}(a|s) = \begin{cases} \dfrac{1}{\beta_1|\mathcal{A}|}, & a \neq \arg\max_{a'\in\mathcal{A}} \phi(s, a')^\top w_t \\ \dfrac{1}{\beta_1|\mathcal{A}|} + 1 - \dfrac{1}{\beta_1}, & a = \arg\max_{a'\in\mathcal{A}} \phi(s, a')^\top w_t. \end{cases}$$

In this chapter, whenever the $\arg\max$ is not unique, we break tie arbitrarily. More generally, we allow the tunable parameter $\beta_1$ to be time-dependent (i.e., $\beta_1$ is a function of the iteration index $t$) and/or state-dependent (i.e., $\beta_1$ is a function of the state $s$).

**$\boldsymbol{\beta_2}$-Softmax Update.** Let $\beta_2 > 0$ be a tunable parameter, which is allowed to be time varying and state-dependent. Then the policy is updated by

$$\pi_{t+1}(a|s) = \frac{\exp(\beta_2 \phi(s,a)^\top w_t)}{\sum_{a' \in \mathcal{A}} \exp(\beta_2 \phi(s,a')^\top w_t)}, \ \forall\, (s,a).$$

In $1/\beta_1$-greedy update or $\beta_2$-softmax update, there is no need to parametrize the policy because it is uniquely determined by the estimate of the $Q$-function, which already uses linear function approximation.

At a first glance of Algorithm 8 line 5, it seems that we need to work with $|\mathcal{S}||\mathcal{A}|$-dimensional objects to update the policy at each state-action pair, which contradicts to the motivation of using function approximation. However, there is an equivalent way of implementing Algorithm 8 without explicitly executing line 5. To see this, first note that the target policy $\pi_t$ in each iteration is only used in the policy evaluation step (Algorithm 8 line 4). To view of our policy evaluation algorithm (cf. Algorithm 7), we only need to compute the policy value of $\pi_t$ at state-action pairs that are visited by the sample trajectory $\{(S_k, A_k)\}$.

When using $1/\beta_1$-greedy update or $\beta_2$-softmax update, Algorithm 8 subsumes the popular value-based method SARSA [11] as its special case. To see this, suppose that we are in the on-policy setting (i.e., $\pi = \pi_b$), and the inner-loop iteration number $K$ is set to 1. Then Algorithm 8 corresponds to SARSA with $1/\beta_1$-greedy exploration policy or Boltzmann exploration policy. However, we need to point out that our result does NOT imply finite-sample bounds for SARSA since we need a relatively large $K$ to provide a sufficiently accurate estimate of the value function before using it in policy improvement.

**$\boldsymbol{\beta_3}$-NPG Update.** Unlike $1/\beta_1$-greedy update or $\beta_2$-softmax update, where we need only the estimate of the $Q$-function to perform to update, in NPG, to update the policy, we need both the current policy and the estimate of its $Q$-function. Therefore, to keep track of the policy, in this case we also need to parametrize the policy using softmax parametrization and compatible linear function approximation. Specifically, with parameter $\theta \in \mathbb{R}^d$, the policy $\pi$ associated with parameter $\theta$ is given by $\pi_\theta(a|s) = \frac{\exp(\phi(s,a)^\top \theta)}{\sum_{a' \in \mathcal{A}} \exp(\phi(s,a')^\top \theta)}$.

Let $\beta_3 > 0$ be a tunable parameter, which is allowed to be time varying. Then NPG updates the parameter $\theta_t$ of the policy according to the formula

$$\theta_{t+1} = \theta_t + \beta_3 w_t. \tag{10.1}$$

See [177] for more details about this update rule. Denote $\pi_t$ as $\pi_{\theta_t}$ for simplicity of notation. Then the update equation can be equivalently written in terms of the policy update (and also in the form of Algorithm 8 line 5) as

$$\pi_{t+1}(a|s) = \frac{\pi_t(a|s)\exp(\beta_3 \phi(s,a)^\top w_t)}{\sum_{a' \in \mathcal{A}} \pi_t(a'|s)\exp(\beta_3 \phi(s,a')^\top w_t)}, \ \forall\, (s,a).$$

This enables us to use the previous equation for our analysis of NPG while using Equation 10.1 for the implementation of Algorithm 8.

### 10.2.1 Finite-Sample Analysis

In this section, we present the finite-sample guarantees of Algorithm 8. For ease of exposition, we implement line 4 of Algorithm 8 with the $\lambda$-averaged $Q$-trace algorithm. The results for using either two-sided $Q$-trace algorithm or Algorithm 7 with more general choices of $c(\cdot,\cdot)$ and $\rho(\cdot,\cdot)$ (as long as Condition 9.3.1 is satisfied) are straightforward extensions. As for the policy improvement (cf. line 5 of Algorithm 8), we use either $1/\beta_1$-greedy policy update, or $\beta_2$-softmax policy update, or $\beta_3$-NPG policy update, with the corresponding parameters satisfying the following condition. Denote $a_{t,s} = \arg\max_{a' \in \mathcal{A}} \phi(s,a')^\top w_t$.

**Condition 10.2.1.** Let $\beta > 0$ be a tunable parameter. (1) The parameter $\beta_1$ is time-varying and state-dependent, and is chosen such that $\beta_1(t,s) \geq \frac{2\gamma}{\beta} \max_{a\in\mathcal{A}} |\phi(s,a)^\top w_t|$ for all $s$ and $t$. (2) The parameter $\beta_2$ is chosen such that $\beta_2 \geq \frac{\gamma}{\beta}\log(|\mathcal{A}|)$. (3) The parameter $\beta_3$ is time-varying, and is chosen such that $\beta_3(t) \geq \frac{\gamma}{\beta}\log(1/\min_{s\in\mathcal{S}} \pi_t(a_{t,s}|s))$ for all $t$.

**Theorem 10.2.1.** *Consider $\pi_t$ of Algorithm 8. Suppose that the assumptions for applying Theorem Theorem 9.4.1 are satisfied, and the choices of $\beta_1$, $\beta_2$, and $\beta_3$ satisfy Condition 10.2.1. Then we have for any $T \geq 0$:*

$$\begin{aligned}\mathbb{E}[\|Q^* - Q^{\pi_T}\|_\infty] \leq{}& \underbrace{\frac{2\gamma\mathcal{E}_{\textit{approx}}}{(1-\gamma)^2}}_{N_1} + \underbrace{\frac{2\gamma^2\mathcal{E}_{\textit{bias}}}{(1-\gamma)^4}}_{N_2} + \underbrace{\gamma^T\|Q^* - Q^{\pi_0}\|_\infty}_{N_3\text{: \textit{Convergence bias in the actor}}} \\ &+ \underbrace{6\tilde{c}(1-(1-\gamma_c)\lambda_{\min}\alpha)^{\frac{1}{2}[K-(t_\alpha+n+1)]}}_{N_4\text{: \textit{Convergence bias in the critic}}} \\ &+ \underbrace{70L\tilde{c}\frac{[\alpha(t_\alpha+n+1)]^{1/2}}{\sqrt{1-\gamma_c}\sqrt{\lambda_{\min}}}}_{N_5\text{: \textit{Critic variance}}} + \underbrace{\frac{2\gamma\beta}{(1-\gamma)^2}}_{N_6},\end{aligned} \tag{10.2}$$

*where*

$$\begin{aligned}\tilde{c} &= \frac{\gamma}{\sqrt{\lambda_{\min}}\sqrt{1-\gamma_c}(1-\gamma)^3}, \\ \mathcal{E}_{\textit{approx}} &= \sup_\pi \|Q^\pi_{c,\rho} - \Phi w^\pi_{c,\rho}\|_\infty, \\ \mathcal{E}_{\textit{bias}} &= \max_{0\leq t\leq T}\max_{s\in\mathcal{S}}(1-\lambda(s))\|\pi_t(\cdot|s) - \pi_b(\cdot|s)\|_1.\end{aligned}$$

Notably on the LHS, our finite-sample guarantees are stated for the last policy iterate $\pi_T$, while in many existing literature it was stated for the best policy among $\{\pi_t\}_{0\leq t\leq T}$ [177].

**The Terms $N_1$ and $N_2$.** The term $N_1$ represents the function approximation bias, and is present in all existing literature that study policy-based methods under function approximation [177]. Note that $N_1 = 0$ when we use a complete basis. The term $N_2$ represents the bias introduced to the algorithm by using generalized importance sampling ratios

$c(\cdot,\cdot)$ and $\rho(\cdot,\cdot)$. Note that we have $N_2 = 0$ when $c(s,a) = \rho(s,a) = \pi(a|s)/\pi_b(a|s)$, which corresponds to using $\lambda = \mathbf{1}$ in the $\lambda$-averaged $Q$-trace algorithm, and using $u(s) \geq \max_{s,a} \pi(a|s)/\pi_b(a|s)$ and $\ell(s) \leq \min_{s,a} \pi(a|s)/\pi_b(a|s)$ for all $s$ in the two-sided $Q$-trace algorithm. However, this choice of $\lambda$ (or $u$ and $\ell$) might lead to a high variance. In particular, the parameter $L$ within the term $N_5$ could be large.

**The terms $N_3$ and $N_4$.** The term $N_3$ represents the convergence bias in the actor, and goes to zero geometrically fast as the outer loop iteration number $T$ goes to infinity. Such geometric convergence is the main reason why we obtain improved sample complexity of $\beta_3$-NPG compared to [141], where the convergence rate of the actor is $\mathcal{O}(1/T)$. The term $N_4$ represents the convergence bias in the critic, and goes to zero geometrically fast as the inner loop iteration number $K$ goes to infinity.

**The terms $N_5$ and $N_6$.** The term $N_5$ represents the variance in the critic, and is proportional to $\sqrt{\alpha t_\alpha} = \mathcal{O}(\sqrt{\alpha \log(1/\alpha)})$. Therefore, $N_5$ can be made arbitrarily small by using small enough stepsize $\alpha$. The term $N_6$ captures the error introduced to the algorithm by the policy update rule $G(\cdot,\cdot)$. To elaborate, consider the following example. Suppose that the underlying MDP model has a unique optimal policy, and suppose we use $1/\beta_1$-greedy update (with a fixed $\beta_1$) in Algorithm 8 line 5. Then as long as $\beta_1$ is finite, we can never truly find the optimal policy $\pi^*$ because of the deterministic nature of $\pi^*$. As a result, the difference between $Q^*$ and $Q^{\pi_t}$ will always be above some threshold, which depends on the choice of $\beta_1$, and is captured by $N_6$. Observe that $N_6$ can be made arbitrarily small by using small enough $\beta$.

Based on Theorem 10.2.1, we next derive the sample complexity of Algorithm 8. To enable fair comparison with existing literature, we choose $\lambda = \mathbf{1}$ to eliminate the error due to using generalized importance sampling ratios. Note that $\lambda = \mathbf{1}$ implies $\mathcal{E}_{\text{bias}} = 0$ (and hence $N_2 = 0$) in Theorem 10.2.1.

**Corollary 10.2.1.** *For a given accuracy level $\epsilon > 0$, to achieve $\mathbb{E}[\|Q^* - Q^{\pi_T}\|_\infty] \leq \epsilon + N_1$,* *the number of samples (e.g. the integer $TK$) required is of the size*

$$\mathcal{O}\left(\frac{\log^3(1/\epsilon)}{\epsilon^2}\right)\tilde{\mathcal{O}}\left(\frac{L^2 n}{(1-\gamma)^7(1-\gamma_c)^3\lambda_{\min}^3}\right).$$

Notably, we obtain $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity for policy-based methods, which matches with the sample complexity of value-based algorithms such as $Q$-learning [134]. In the case of $\beta_3$-NPG update, to our knowledge, [178, 163] establishes the $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity of on-policy NAC under regularization, and [141] establishes the $\tilde{\mathcal{O}}(\epsilon^{-3})$ sample complexity of a variant of off-policy NAC (where the infamous deadly triad is present). We improve the sample complexity in [141] by a factor of $\epsilon^{-1}$, and we do not use regularization.

In addition to the dependence on $\epsilon$, the dependence on $1/(1-\gamma)$ (which is usually called the effective horizon) is also improved by a factor of $1/(1-\gamma)$ compared to existing work [177, 141]. The bootstrapping parameter $n$ appears linearly in our sample complexity bound. This matches with the results for $n$-step TD-learning in the on-policy tabular setting [16].

## 10.3 Proof Sketch of Theorem 10.2.1

We first introduce some notation. Let $\mathcal{H}:\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}\mapsto\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ be the Bellman optimality operator defined by $[\mathcal{H}(Q)](s,a)=\mathcal{R}(s,a)+\gamma\mathbb{E}[\max_{a'\in\mathcal{A}}Q(S_{k+1},a')\mid S_k=s,A_k=a]$ for all $(s,a)$, and let $\mathcal{H}_\pi:\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}\mapsto\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ be the Bellman operator associated with policy $\pi$ defined by $[\mathcal{H}_\pi(Q)](s,a)=\mathcal{R}(s,a)+\gamma\mathbb{E}_\pi[Q(S_{k+1},A_{k+1})\mid S_k=s,A_k=a]$ for all $(s,a)$.

The key to prove Theorem 10.2.1 is the following proposition.

**Proposition 10.3.1.** *Consider $\{\pi_T\}$ of Algorithm 8. The following inequality holds for any $T\geq 0$:*

$$\mathbb{E}[\|Q^*-Q^{\pi_T}\|_\infty]\leq\gamma^T\|Q^*-Q^{\pi_0}\|_\infty \qquad A_1$$

$$+\frac{2\gamma}{1-\gamma}\sum_{i=0}^{T-1}\gamma^{T-1-i}\mathbb{E}[\|Q^{\pi_i}-\Phi w_i\|_\infty] \qquad A_2$$

$$+\frac{2\gamma}{1-\gamma}\sum_{i=0}^{T-1}\gamma^{T-1-i}\mathbb{E}[\|\mathcal{H}_{\pi_{i+1}}(\Phi w_i)-\mathcal{H}(\Phi w_i)\|_\infty]. \qquad A_3$$

In light of Proposition 10.3.1, to proceed and establish finite-sample bound of Algorithm 8, it remains to control the terms $A_2$ and $A_3$ when the policy evaluation algorithm and the policy update rule are specified. Specifically, we control $A_2$ by using Theorem Theorem 9.4.1, and control $A_3$ by using Condition 10.2.1 on the parameters $\beta_1$, $\beta_2$, and $\beta_3$ for various policy update rules. See the next section for more details.

Before we present the key steps to prove Proposition 10.3.1, consider a special case of tabular RL, and choosing $c(s,a)=\rho(s,a)=\frac{\pi(a|s)}{\pi_b(a|s)}$ in Algorithm 7. Note that the term $A_2$ vanishes. Since the term $A_3$ can be made arbitrarily small by using large enough $\beta$, Proposition 10.3.1 implies *geometric* convergence for NPG. The geometric convergence of NPG was previously established in [178, 163] under regularization, and in [179] in the asymptotic region. We do not require regularization to establish the result, and our result holds for all $T\geq 0$.

Next we present the proof sketch of Proposition 10.3.1. In most of the existing literature, for policy-based type of algorithms, the analysis is usually based on the mirror descent analysis in optimization [43], where the $\mathcal{KL}$-divergence was chosen as a potential/Lyapunov function, and the performance difference lemma was extensively used [177, 178]. To establish Proposition 10.3.1, we use a completely different approach, where we only exploit the contraction and the monotonicity of the Bellman operators $\mathcal{H}_\pi(\cdot)$ and $\mathcal{H}(\cdot)$. Such proof technique was inspired by [11] Section 6.2. However, only asymptotic error bound of approximate policy iteration was established in [11], while we establish finite-sample bounds for various policy update rules. Proposition 10.3.1 builds on the following two lemmas.

**Lemma 10.3.1.** *It holds for all $t \geq 0$ that*

$$\max_{s,a}(Q^{\pi_t}(s,a) - Q^{\pi_{t+1}}(s,a)) \leq \frac{2\gamma\|Q^{\pi_t} - \Phi w_t\|_\infty + \|\mathcal{H}_{\pi_{t+1}}(\Phi w_t) - \mathcal{H}(\Phi w_t)\|_\infty}{1-\gamma}.$$

**Lemma 10.3.2.** *It holds for all $t \geq 0$ that*

$$\|Q^* - Q^{\pi_{t+1}}\|_\infty \leq \gamma\|Q^* - Q^{\pi_t}\|_\infty + \frac{2\gamma\|Q^{\pi_t} - \Phi w_t\|_\infty + \|\mathcal{H}_{\pi_{t+1}}(\Phi w_t) - \mathcal{H}(\Phi w_t)\|_\infty}{1-\gamma}.$$

Proposition 10.3.1 then follows by repeatedly using Lemma 10.3.2 and then taking expectation on both sides of the resulting inequality.

## 10.4 Proof of All Theoretical Results

### 10.4.1 Proof of Theorem 10.2.1

We begin with the result of Proposition 10.3.1:

$$\begin{aligned}\mathbb{E}[\|Q^* - Q^{\pi_T}\|_\infty] \leq \gamma^T\|Q^* - Q^{\pi_0}\|_\infty + \underbrace{\frac{2\gamma}{1-\gamma}\sum_{i=0}^{T-1}\gamma^{T-1-i}\mathbb{E}[\|Q^{\pi_i} - \Phi w_i\|_\infty]}_{A_2} \\ + \underbrace{\frac{2\gamma}{1-\gamma}\sum_{i=0}^{T-1}\gamma^{T-1-i}\mathbb{E}[\|\mathcal{H}_{\pi_{i+1}}(\Phi w_i) - \mathcal{H}(\Phi w_i)\|_\infty]}_{A_3}.\end{aligned} \tag{10.3}$$

*The Term $A_2$*

To control the term $A_2$, using triangle inequality and we have for any $0 \leq i \leq T-1$:

$$\begin{aligned}\mathbb{E}[\|Q^{\pi_i} - \Phi w_i\|_\infty] &\leq \mathbb{E}[\|Q^{\pi_i} - \Phi w_{c,\rho}^{\pi_i} + \Phi w_{c,\rho}^{\pi_i} - \Phi w_i\|_\infty] \\ &\leq \mathbb{E}[\|Q^{\pi_i} - \Phi w_{c,\rho}^{\pi_i}\|_\infty] + \mathbb{E}[\|\Phi(w_{c,\rho}^{\pi_i} - w_i)\|_\infty] \\ &\leq \mathbb{E}[\|Q^{\pi_i} - \Phi w_{c,\rho}^{\pi_i}\|_\infty] + \|\Phi\|_\infty\mathbb{E}[\|w_{c,\rho}^{\pi_i} - w_i\|_\infty]\end{aligned}$$

$$
\begin{aligned}
&\leq \mathbb{E}[\|Q^{\pi_i} - \Phi w_{c,\rho}^{\pi_i}\|_\infty] + \mathbb{E}[\|w_{c,\rho}^{\pi_i} - w_i\|_\infty] \\
&\leq \mathbb{E}[\|Q_{c,\rho}^{\pi_i} - \Phi w_{c,\rho}^{\pi_i}\|_\infty] + \mathbb{E}[\|Q_{c,\rho}^{\pi_i} - Q^{\pi_i}\|_\infty] + \mathbb{E}[\|w_{c,\rho}^{\pi_i} - w_i\|_\infty] \\
&\hspace{20em} (\|\Phi\|_\infty \leq 1) \\
&\leq \mathcal{E}_{\text{approx}} + \frac{\gamma}{(1-\gamma)^2} \max_{s\in\mathcal{S}} (1-\lambda(s))\|\pi_i(\cdot|s) - \pi_b(\cdot|s)\|_1 \\
&\quad + \mathbb{E}[\|w_{c,\rho}^{\pi_i} - w_i\|_\infty] \qquad (10.4) \\
&\leq \mathcal{E}_{\text{approx}} + \frac{\gamma}{(1-\gamma)^2}\mathcal{E}_{\text{bias}} + \mathbb{E}[\|w_{c,\rho}^{\pi_i} - w_i\|_\infty]. \qquad (10.5)
\end{aligned}
$$

It remains to control $\mathbb{E}[\|w_{c,\rho}^{\pi_i} - w_i\|_\infty]$. For any $0 \leq i \leq T-1$, we have by Theorem 9.4.1 that

$$
\begin{aligned}
\mathbb{E}[\|w_{c,\rho}^{\pi_i} - w_i\|_\infty] &\leq \mathbb{E}[\|w_{c,\rho}^{\pi_i} - w_i\|_2] \\
&\leq (\mathbb{E}[\|w_{c,\rho}^{\pi_i} - w_i\|_2^2])^{1/2} \qquad \text{(Jensen's Inequality)} \\
&\leq c_{1,i}(1-(1-\gamma_c)\lambda_{\min}\alpha)^{\frac{1}{2}[K-(t_\alpha+n+1)]} + c_{2,i}\frac{[\alpha(t_\alpha+n+1)]^{1/2}}{\sqrt{1-\gamma_c}\sqrt{\lambda_{\min}}},
\end{aligned}
$$

where the last line follows from $\sqrt{a+b} \leq \sqrt{a}+\sqrt{b}$ for any $a, b \geq 0$, and $c_{1,i} = \|w_{c,\rho}^{\pi_i}\|_2+1$ and $c_{2,i} = 11.5L(\|w_{c,\rho}^{\pi_i}\|_2 + 1)$. To further control the constants $c_{1,i}$ and $c_{2,i}$, note that we have for any policy $\pi$ that

$$
\begin{aligned}
\|w_{c,\rho}^{\pi}\|_2 &\leq \frac{1}{\sqrt{\lambda_{\min}}}\|\Phi w_{c,\rho}^{\pi}\|_{\mathcal{K}_{SA}} \\
&\leq \frac{1}{\sqrt{\lambda_{\min}}}\left(\|Q_{c,\rho}^{\pi}\|_{\mathcal{K}_{SA}} + \frac{1}{\sqrt{1-\gamma_c^2}(1-\gamma)}\right) \\
&\leq \frac{1}{\sqrt{\lambda_{\min}}}\left(\frac{1}{1-\gamma} + \frac{1}{\sqrt{1-\gamma_c^2}(1-\gamma)}\right) \\
&\leq \frac{2}{\sqrt{\lambda_{\min}}(1-\gamma)\sqrt{1-\gamma_c}}.
\end{aligned}
$$

Therefore we have $c_{1,i} \leq \frac{3}{\sqrt{\lambda_{\min}}(1-\gamma)\sqrt{1-\gamma_c}}$ and $c_{2,i} \leq \frac{35L}{\sqrt{\lambda_{\min}}(1-\gamma)\sqrt{1-\gamma_c}}$ for any $0 \leq i \leq T-1$. Substituting the upper bound we obtained for $\mathbb{E}[\|w_{c,\rho}^{\pi_i} - w_i\|_\infty]$ into Equation 10.5 and we have for any $0 \leq i \leq T-1$:

$$\begin{aligned}\mathbb{E}[\|Q^{\pi_i} - \Phi w_i\|_\infty] \leq{}& \mathcal{E}_{\text{approx}} + \frac{\gamma}{(1-\gamma)^2}\mathcal{E}_{\text{bias}} \\ &+ \frac{3}{\sqrt{\lambda_{\min}}(1-\gamma)\sqrt{1-\gamma_c}}(1-(1-\gamma_c)\lambda_{\min}\alpha)^{\frac{1}{2}[K-(t_\alpha+n+1)]} \\ &+ \frac{35L[\alpha(t_\alpha+n+1)]^{1/2}}{(1-\gamma)(1-\gamma_c)\lambda_{\min}}.\end{aligned}$$

Finally, using the previous inequality and we obtain the following bound on the term $A_2$:

$$\begin{aligned}A_2 ={}& \frac{2\gamma}{1-\gamma}\sum_{i=0}^{T-1}\gamma^{T-1-i}\mathbb{E}[\|Q^{\pi_i} - \Phi w_i\|_\infty] \\ \leq{}& \frac{2\gamma\mathcal{E}_{\text{approx}}}{(1-\gamma)^2} + \frac{2\gamma^2\mathcal{E}_{\text{bias}}}{(1-\gamma)^4} + 6\tilde{c}(1-(1-\gamma_c)\lambda_{\min}\alpha)^{\frac{1}{2}[K-(t_\alpha+n+1)]} \\ &+ \frac{70\tilde{c}L[\alpha(t_\alpha+n+1)]^{1/2}}{\sqrt{\lambda_{\min}}\sqrt{1-\gamma_c}},\end{aligned}$$

where $\tilde{c} = \frac{\gamma}{\sqrt{\lambda_{\min}}\sqrt{1-\gamma_c}(1-\gamma)^3}$.

*The Term $A_3$*

Now consider the term $A_3$, whose upper bound depends on which policy update rule we use.

**$1/\beta_1$-Greedy Update** For simplicity of notation, denote $Q_t = \Phi w_t$. Then we have for any $0 \leq t \leq T-1$ and state-action pair $(s,a)$ that

$$\begin{aligned}0 \leq{}& [\mathcal{H}(Q_t)](s,a) - [\mathcal{H}_{\pi_{t+1}}(Q_t)](s,a) \\ ={}& \left[\mathcal{R}(s,a) + \gamma\sum_{s'\in\mathcal{S}}P_a(s,s')Q_t(s',a_{t,s'})\right] \quad \text{(Recall that } a_{t,s'} = \arg\max_{a'\in\mathcal{A}} Q_t(s',a')\text{)} \\ &- \left\{\mathcal{R}(s,a) + \gamma\sum_{s'\in\mathcal{S}}P_a(s,s')\left[\left(1 - \frac{1}{\beta_1(t,s')} + \frac{1}{|\mathcal{A}|\beta_1(t,s')}\right)Q_t(s',a_{t,s'})\right.\right. \\ &\left.\left. + \sum_{a'\neq a_{t,s'}}\frac{1}{|\mathcal{A}|\beta_1(t,s')}Q_t(s',a')\right]\right\}\end{aligned}$$

$$
\begin{aligned}
&= \gamma \sum_{s'} P_a(s,s') \left( \left( \frac{1}{\beta_1(t,s')} - \frac{1}{|\mathcal{A}|\beta_1(t,s')} \right) Q_t(s', a_{t,s'}) - \sum_{a' \neq a_{t,s'}} \frac{1}{|\mathcal{A}|\beta_1(t,s')} Q_t(s',a') \right) \\
&\leq \gamma \sum_{s'} P_a(s,s') \frac{2}{\beta_1(t,s')} \max_{a' \in \mathcal{A}} |Q_t(s',a')| \\
&\leq \beta,
\end{aligned}
$$

where the last line follows from $\beta_1(t,s) \geq \frac{2\gamma}{\beta} \max_{a \in \mathcal{A}} |Q_t(s,a)|$ for all $s \in \mathcal{S}$ (cf. Condition 10.2.1). Therefore, we have

$$
A_3 \leq \frac{2\gamma}{1-\gamma} \sum_{i=0}^{T-1} \gamma^{T-1-i} \beta \leq \frac{2\gamma\beta}{(1-\gamma)^2}.
$$

**$\boldsymbol{\beta_2}$-Softmax Update** The following lemma is needed for us to control the term $A_3$.

**Lemma 10.4.1.** *For any $x \in \mathbb{R}^d$ and $y \in \Delta^d$ satisfying $y_i > 0$ for all $i$, denote $i_{\max} = \arg\max_{1 \leq i \leq d} x_i$, then the following inequality holds for any $\beta > 0$:*

$$
\max_{1 \leq i \leq d} x_i - \frac{\sum_{i=1}^d x_i y_i e^{\beta x_i}}{\sum_{j=1}^d y_j e^{\beta x_j}} \leq \frac{1}{\beta} \log \left( \frac{1}{y_{i_{\max}}} \right).
$$

*Proof of Lemma 10.4.1.* For any $\beta > 0$, consider the function $h_\beta : \mathbb{R}^d \mapsto \mathbb{R}$ defined by

$$
h_\beta(x) = \frac{1}{\beta} \log \left( \sum_{i=1}^d y_i e^{\beta x_i} \right).
$$

Assume without loss of generality that $i_{\max} = 1$. Then it is clear that $h_\beta(x) \leq x_1$. On the other hand, we have

$$
x_1 \leq \frac{1}{\beta} \log \left( \sum_{i=1}^d \frac{y_i}{y_1} e^{\beta x_i} \right) = h_\beta(x) + \frac{1}{\beta} \log \left( \frac{1}{y_1} \right). \tag{10.6}
$$

Since it is well-known that $h_\beta(x)$ is a convex differentiable function, we have for any

$x \in \mathbb{R}^d$ that $h_\beta(0) - h_\beta(x) \geq \langle \nabla h_\beta(x), -x \rangle$, which implies

$$\langle \nabla h_\beta(x), x \rangle = \frac{\sum_{i=1}^d x_i y_i e^{\beta x_i}}{\sum_{j=1}^d y_j e^{\beta x_j}} \geq h_\beta(x) - h_\beta(0) = h_\beta(x). \tag{10.7}$$

Using Equation 10.6 and Equation 10.7 and we finally obtain

$$\max_{1 \leq i \leq d} x_i - \frac{\sum_{i=1}^d x_i y_i e^{\beta x_i}}{\sum_{j=1}^d y_j e^{\beta x_j}} \leq x_1 - h_\beta(x) \leq \frac{1}{\beta} \log\left(\frac{1}{y_1}\right).$$

□

We now proceed to control the term $A_3$ when using the $\beta_2$-softmax update. For any $0 \leq t \leq T-1$ and state-action pair $(s, a)$, we have

$$\begin{aligned}
&0 \leq [\mathcal{H}(Q_t)](s,a) - [\mathcal{H}_{\pi_{t+1}}(Q_t)](s,a) \\
&= \gamma \sum_{s'} P_a(s,s') \left( \max_{a' \in \mathcal{A}} Q_t(s',a') - \sum_{a' \in \mathcal{A}} \frac{\exp(\beta_2 Q_t(s',a'))}{\sum_{a'' \in \mathcal{A}} \exp(\beta_2 Q_t(s',a''))} Q_t(s',a') \right) \\
&= \gamma \sum_{s'} P_a(s,s') \left( \max_{a' \in \mathcal{A}} Q_t(s',a') - \sum_{a' \in \mathcal{A}} \frac{\exp(\beta_2 Q_t(s',a'))/|\mathcal{A}|}{\sum_{a'' \in \mathcal{A}} \exp(\beta_2 Q_t(s',a''))/|\mathcal{A}|} Q_t(s',a') \right) \\
&\leq \frac{\gamma}{\beta_2} \log(|\mathcal{A}|) \qquad \text{(Lemma 10.4.1)} \\
&\leq \beta,
\end{aligned}$$

where the last line follows from $\beta_2 \geq \frac{\gamma}{\beta} \log(|\mathcal{A}|)$. Therefore, we have

$$A_3 \leq \frac{2\gamma}{1-\gamma} \sum_{i=0}^{T-1} \gamma^{T-1-i} \beta \leq \frac{2\gamma\beta}{(1-\gamma)^2}.$$

**$\beta_3$-NPG Update** Recall that $\beta_3$-NPG updates the policy according to

$$\pi_{t+1}(a|s) = \frac{\pi_t(a|s) \exp(\beta_3(t) Q_t(s,a))}{\sum_{a' \in \mathcal{A}} \pi_t(a'|s) \exp(\beta_3(t) Q_t(s,a'))}, \ \forall \ (s,a).$$

Therefore, for any $0 \leq t \leq T-1$ and state-action pair $(s,a)$, we have

$$
\begin{aligned}
0 &\leq [\mathcal{H}(Q_t)](s,a) - [\mathcal{H}_{\pi_{t+1}}(Q_t)](s,a) \\
&= \gamma \sum_{s'} P_a(s,s') \left( \max_{a' \in \mathcal{A}} Q_t(s',a') - \sum_{a' \in \mathcal{A}} \frac{\pi_t(a'|s') \exp(\beta_3(t) Q_t(s',a'))}{\sum_{a'' \in \mathcal{A}} \pi_t(a''|s') \exp(\beta_3(t) Q_t(s',a''))} Q_t(s',a') \right) \\
&\leq \frac{\gamma}{\beta_3(t)} \log \left( \frac{1}{\pi_t(a_{t,s'}|s')} \right) \\
&\leq \beta,
\end{aligned}
$$

where the last line follows from $\beta_3(t) \geq \frac{\gamma}{\beta} \log(1/\min_{s \in \mathcal{S}} \pi_t(a_{t,s}|s))$. Therefore, we have

$$
A_3 \leq \frac{2\gamma}{1-\gamma} \sum_{i=0}^{T-1} \gamma^{T-1-i} \beta \leq \frac{2\gamma\beta}{(1-\gamma)^2}.
$$

### 10.4.2 Putting Together

Using the upper bounds we obtained for the terms $A_2$ and $A_3$ in Equation 10.3 and we have for any $K \geq t_\alpha + n + 1$ and $T \geq 0$ that

$$
\begin{aligned}
\mathbb{E}[\|Q^* - Q^{\pi_T}\|_\infty] \leq \gamma^T \|Q^* - Q^{\pi_0}\|_\infty &+ \frac{2\gamma \mathcal{E}_{\text{approx}}}{(1-\gamma)^2} + \frac{2\gamma^2 \mathcal{E}_{\text{bias}}}{(1-\gamma)^4} \\
&+ 6\tilde{c}(1-(1-\gamma_c)\lambda_{\min}\alpha)^{\frac{1}{2}[K-(t_\alpha+n+1)]} \\
&+ \frac{70\tilde{c}L[\alpha(t_\alpha+n+1)]^{1/2}}{\sqrt{\lambda_{\min}}\sqrt{1-\gamma_c}} + \frac{2\gamma\beta}{(1-\gamma)^2},
\end{aligned}
$$

where $\tilde{c} = \frac{\gamma}{\sqrt{\lambda_{\min}}\sqrt{1-\gamma_c}(1-\gamma)^3}$.

### 10.4.3 Proof of Lemma 10.3.1

For simplicity of notation, denote $\delta_t = \max_{s,a}(Q^{\pi_t}(s,a) - Q^{\pi_{t+1}}(s,a))$. Then we have by definition of $\delta_t$ that $Q^{\pi_{t+1}} \geq Q^{\pi_t} - \delta_t \mathbf{1}$. Using the monotonicity of the Bellman operator

[11, Lemma 2.1 and Lemma 2.2] and we have

$$Q^{\pi_{t+1}} = \mathcal{H}_{\pi_{t+1}}(Q^{\pi_{t+1}}) \geq \mathcal{H}_{\pi_{t+1}}(Q^{\pi_t} - \delta_t \mathbf{1}) = \mathcal{H}_{\pi_{t+1}}(Q^{\pi_t}) - \gamma\delta_t \mathbf{1}.$$

It follows that

$$\begin{aligned}
&Q^{\pi_t} - Q^{\pi_{t+1}} \\
\leq\, & Q^{\pi_t} - \mathcal{H}_{\pi_{t+1}}(Q^{\pi_t}) + \gamma\delta_t \mathbf{1} \\
=\, & Q^{\pi_t} - \mathcal{H}_{\pi_{t+1}}(Q^{\pi_t}) + \mathcal{H}_{\pi_{t+1}}(Q_t) - \mathcal{H}_{\pi_{t+1}}(Q_t) + \mathcal{H}(Q_t) - \mathcal{H}(Q_t) + \gamma\delta_t \mathbf{1} \\
\leq\, & \mathcal{H}_{\pi_t}(Q^{\pi_t}) - \mathcal{H}_{\pi_t}(Q_t) - \mathcal{H}_{\pi_{t+1}}(Q^{\pi_t}) + \mathcal{H}_{\pi_{t+1}}(Q_t) - \mathcal{H}_{\pi_{t+1}}(Q_t) + \mathcal{H}(Q_t) + \gamma\delta_t \mathbf{1} \\
\leq\, & 2\gamma\|Q^{\pi_t} - Q_t\|_\infty \mathbf{1} + \|\mathcal{H}_{\pi_{t+1}}(Q_t) - \mathcal{H}(Q_t)\|_\infty \mathbf{1} + \gamma\delta_t \mathbf{1}.
\end{aligned}$$

Therefore, we have

$$\delta_t \leq 2\gamma\|Q^{\pi_t} - Q_t\|_\infty + \|\mathcal{H}_{\pi_{t+1}}(Q_t) - \mathcal{H}(Q_t)\|_\infty + \gamma\delta_t,$$

which implies

$$\delta_t \leq \frac{2\gamma\|Q^{\pi_t} - Q_t\|_\infty + \|\mathcal{H}_{\pi_{t+1}}(Q_t) - \mathcal{H}(Q_t)\|_\infty}{1-\gamma}.$$

### 10.4.4 Proof of Lemma 10.3.2

For simplicity of notation, denote $\zeta_t = \max_{s,a}(Q^*(s,a) - Q^{\pi_t}(s,a)) = \|Q^* - Q^{\pi_t}\|_\infty$. Then we have by definition of $\zeta_t$ that $Q^{\pi_t} \geq Q^* - \zeta_t \mathbf{1}$. Using the monotonicity of the Bellman operator and we have

$$\begin{aligned}
Q^{\pi_{t+1}} &= \mathcal{H}_{\pi_{t+1}}(Q^{\pi_{t+1}}) \\
&\geq \mathcal{H}_{\pi_{t+1}}(Q^{\pi_t} - \max_{s,a}(Q^{\pi_t}(s,a) - Q^{\pi_{t+1}}(s,a))\mathbf{1})
\end{aligned}$$

$$
\begin{aligned}
&= \mathcal{H}_{\pi_{t+1}}(Q^{\pi_t}) - \gamma \max_{s,a}(Q^{\pi_t}(s,a) - Q^{\pi_{t+1}}(s,a))\mathbf{1} \\
&\geq \mathcal{H}_{\pi_{t+1}}(Q^{\pi_t}) - \frac{2\gamma^2\|Q^{\pi_t} - Q_t\|_\infty + \gamma\|\mathcal{H}_{\pi_{t+1}}(Q_t) - \mathcal{H}(Q_t)\|_\infty}{1-\gamma}\mathbf{1}, \qquad (10.8)
\end{aligned}
$$

where the last line follows from Lemma 10.3.1. We next control $\mathcal{H}_{\pi_{t+1}}(Q^{\pi_t})$ from below in the following. Again by monotonicity of the Bellman operator we have

$$
\begin{aligned}
\mathcal{H}_{\pi_{t+1}}(Q^{\pi_t}) &\geq \mathcal{H}_{\pi_{t+1}}(Q_t - \|Q_t - Q^{\pi_t}\|_\infty \mathbf{1}) \\
&= \mathcal{H}_{\pi_{t+1}}(Q_t) - \gamma\|Q_t - Q^{\pi_t}\|_\infty \mathbf{1} \\
&= \mathcal{H}_{\pi_{t+1}}(Q_t) - \mathcal{H}(Q_t) + \mathcal{H}(Q_t) - \gamma\|Q_t - Q^{\pi_t}\|_\infty \mathbf{1} \\
&\geq \mathcal{H}_{\pi_{t+1}}(Q_t) - \mathcal{H}(Q_t) + \mathcal{H}(Q^{\pi_t} - \|Q_t - Q^{\pi_t}\|_\infty \mathbf{1}) - \gamma\|Q_t - Q^{\pi_t}\|_\infty \mathbf{1} \\
&= \mathcal{H}_{\pi_{t+1}}(Q_t) - \mathcal{H}(Q_t) + \mathcal{H}(Q^{\pi_t}) - 2\gamma\|Q_t - Q^{\pi_t}\|_\infty \mathbf{1} \\
&\geq \mathcal{H}_{\pi_{t+1}}(Q_t) - \mathcal{H}(Q_t) + \mathcal{H}(Q^* - \zeta_t \mathbf{1}) - 2\gamma\|Q_t - Q^{\pi_t}\|_\infty \mathbf{1} \\
&= \mathcal{H}_{\pi_{t+1}}(Q_t) - \mathcal{H}(Q_t) + \mathcal{H}(Q^*) - \gamma\zeta_t \mathbf{1} - 2\gamma\|Q_t - Q^{\pi_t}\|_\infty \mathbf{1} \\
&\geq -\|\mathcal{H}_{\pi_{t+1}}(Q_t) - \mathcal{H}(Q_t)\|_\infty \mathbf{1} + Q^* - \gamma\zeta_t \mathbf{1} - 2\gamma\|Q_t - Q^{\pi_t}\|_\infty \mathbf{1}.
\end{aligned}
$$

Using the previous inequality in Equation 10.8 and we have

$$
Q^{\pi_{t+1}} - Q^* \geq -\gamma\zeta_t \mathbf{1} - \frac{2\gamma\|Q^{\pi_t} - Q_t\|_\infty + \|\mathcal{H}_{\pi_{t+1}}(Q_t) - \mathcal{H}(Q_t)\|_\infty}{1-\gamma}\mathbf{1},
$$

which implies

$$
\zeta_{t+1} \leq \gamma\zeta_t + \frac{2\gamma\|Q^{\pi_t} - Q_t\|_\infty + \|\mathcal{H}_{\pi_{t+1}}(Q_t) - \mathcal{H}(Q_t)\|_\infty}{1-\gamma}.
$$

## 10.5 Conclusion and Future Work

In this chapter, we focus on general policy-based methods under off-policy sampling and linear function approximation, and establish an $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity, which matches with the sample complexity of value-based algorithms such as $Q$-learning. Note that our generic algorithm is a two-loop algorithm, while in practical applications, two time-scale algorithms are more preferred. However, analyzing two time-scale algorithms is fundamentally more challenging, and the state-of-the-art sample complexity there is worse than that of two-loop type of algorithms. Studying two time-scale AC algorithms and getting improved sample complexity are interesting future directions.

# CHAPTER 11

# $Q$-LEARNING WITH LINEAR FUNCTION APPROXIMATION

## 11.1 Introduction

Recall from Chapter 8 that the goal of $Q$-learning is to learn the optimal $Q$-function $Q^*$, and once $Q^*$ is obtained, we can immediately find an optimal policy by computing $\pi^*(s) \in \arg\max_{a' \in \mathcal{A}} Q^*(s,a')$ for all $s \in \mathcal{S}$. While $Q$-learning provably converges, due to the fact that $Q$-learning performs asynchronous update, it lacks computational tractability when the size of the state-action space is large. In this chapter, to overcome the aforementioned computational challenge, we consider $Q$-learning with linear function approximation.

We begin by describing the linear parametric architecture. Let $\phi_i \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, $i = 1, 2, ..., d$, be a set of basis vectors, and denote $\phi(s,a) = (\phi_1(s,a), \cdots, \phi_d(s,a)) \in \mathbb{R}^d$ for all $(s,a)$, which can be viewed as the feature associated with state-action pair $(s,a)$. We assume without loss of generality that the basis vectors $\{\phi_i\}_{1 \leq i \leq d}$ are linearly independent, and are normalized so that $\|\phi(s,a)\|_2 \leq 1$ for all $(s,a)$. Let the feature matrix $\Phi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times d}$ be defined by

$$\Phi = \begin{bmatrix} | & & | \\ \phi_1 & ... & \phi_d \\ | & & | \end{bmatrix} = \begin{bmatrix} — & \phi(s_1, a_1)^\top & — \\ ... & ... & ... \\ — & \phi(s_{|\mathcal{S}|}, a_{|\mathcal{A}|})^\top & — \end{bmatrix}.$$

Using the feature matrix $\Phi$, the linear sub-space spanned by $\{\phi_i\}_{1 \leq i \leq d}$ can be compactly written as $\mathcal{W} = \{Q_\theta \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \mid Q_\theta = \Phi\theta,\ \theta \in \mathbb{R}^d\}$. The goal of $Q$-learning with linear function approximation is to design a stable algorithm that provably finds an approximation of the optimal $Q$-function $Q^*$ from the linear sub-space $\mathcal{W}$.

## 11.2 Classical Semi-Gradient $Q$-Learning with Linear Function Approximation

In this section, we present the classical $Q$-learning algorithm under linear function approximation [11], and provide its finite-sample bounds.

### 11.2.1 The Algorithm

We begin with the semi-gradient $Q$-learning algorithm presented in the following.

**Algorithm 9** Classical Semi-Gradient $Q$-Learning with Linear Function Approximation

```
**Input:** Integer $K$, initialization $\theta_0 \in \mathbb{R}^d$, and behavior policy $\pi_b$
**for** $k = 0, 1, \cdots, K-1$ **do**
    Sample $A_k \sim \pi_b(\cdot|S_k)$, observe $S_{k+1} \sim P_{A_k}(S_k, \cdot)$
    $\theta_{k+1} = \theta_k + \alpha_k \phi(S_k, A_k)(\mathcal{R}(S_k, A_k) + \gamma \max_{a' \in \mathcal{A}} \phi(S_{k+1}, a')^\top \theta_k - \phi(S_k, A_k)^\top \theta_k)$
**end for**
**Output:** $\theta_K$
```

The reason that Algorithm 9 is called semi-gradient $Q$-learning is that it can be interpreted as a one step stochastic semi-gradient descent for minimizing the projected Bellman error. See [11] for more details.

Alternatively, Algorithm 9 can be viewed as an SA algorithm for solving the equation

$$\mathbb{E}_{\kappa_S, \pi_b}\left[\phi(S, A)(\mathcal{R}(S, A) + \gamma \max_{a' \in \mathcal{A}} \phi(S', a')^\top \theta - \phi(S, A)^\top \theta)\right] = 0, \tag{11.1}$$

where $\kappa_S$ stands for the stationary distribution of the Markov chain $\{S_k\}$ under policy $\pi_b$ (provided that it exists and is unique). Under some mild conditions, Equation 11.1 is equivalent to a so-called *projected Bellman equation* [180]. In the special case where the feature matrix $\Phi$ is an identity matrix, Algorithm 9 reduces to the tabular $Q$-learning algorithm, and Equation 11.1 becomes the regular Bellman equation for $Q^*$.

In general, Equation 11.1 may not necessarily admit a solution [181], and the iteration in Algorithm 9 may diverge [8]. However, it was shown in [180] that under an assumption on the behavior policy $\pi_b$, $\theta_k$ converges to the solution of Equation 11.1, denoted by $\theta^*$, almost surely. In this chapter, we work with a similar condition, and focus on establishing the finite-sample bounds of Algorithm 9. We begin by stating our assumptions.

**Assumption 11.2.1.** The behavior policy $\pi_b$ satisfies $\pi_b(a|s) > 0$ for all $(s, a)$, and the Markov chain $\{S_k\}$ induced by $\pi_b$ is irreducible and aperiodic.

Assumption 11.2.1 essentially requires that the behavior policy $\pi_b$ has enough exploration, and is standard in studying off-policy value-based RL algorithms [92, 135]. Under Assumption 11.2.1, the Markov chain $\{S_k\}$ has a unique stationary distribution, which we have denoted by $\kappa_S$. In addition, since the state-space $\mathcal{S}$ is finite, the Markov chain $\{S_k\}$ mixes geometrically fast in that there exist $C \geq 1$ and $\sigma \in (0, 1)$ such that $\max_{s\in\mathcal{S}} \|P^k_{\pi_b}(s, \cdot) - \kappa_S(\cdot)\|_{\text{TV}} \leq C\sigma^k$ for all $k \geq 0$ [48].

**Assumption 11.2.2.** The target equation (cf. Equation 11.1) has a unique solution $\theta^*$, and there exists $\kappa > 0$ such that the following inequality holds for all $\theta \in \mathbb{R}^d$:

$$\gamma^2\mathbb{E}_{\kappa_S}[\max_{a\in\mathcal{A}} Q_\theta(S, a)^2] - \mathbb{E}_{\kappa_S,\pi_b}[Q_\theta(S, A)^2] \leq -\kappa\|\theta\|_2^2. \tag{11.2}$$

We make Assumption 11.2.2 and especially Equation 11.2 to ensure the stability of Algorithm 9, which is in the same spirit to the conditions proposed in [180]. A detailed discussion about this assumption and comparison to related conditions are presented in Subsection 11.2.3.

### 11.2.2 Finite-Sample Guarantees

To establish the finite-sample guarantees of Algorithm 9 using our SA results, we begin by modeling Algorithm 9 in the form of the SA algorithm presented in Chapter 3 (cf. Algorithm 2).

Define $Y_k = (S_k, A_k, S_{k+1})$ for all $k \geq 0$. It is clear that $\{Y_k\}$ is also a Markov chain with finite state-space $\mathcal{Y} = \{(s, a, s') \mid s \in \mathcal{S}, \pi_b(a|s) > 0, P_a(s, s') > 0\}$. Moreover, under Assumption 11.2.1, the Markov chain $\{Y_k\}$ also has a unique stationary distribution, which we denote by $\mu_Y$, and is given by $\mu_Y(s,a,s') = \kappa_S(s)\pi_b(a|s)P_a(s,s')$ for all $(s,a,s') \in \mathcal{Y}$. Define an operator $F : \mathbb{R}^d \times \mathcal{Y} \mapsto \mathbb{R}^d$ by

$$F(\theta,y) = F(\theta,s,a,s') = \phi(s,a)\left(\mathcal{R}(s,a) + \gamma \max_{a'\in\mathcal{A}} \phi(s',a')^\top\theta - \phi(s,a)^\top\theta\right) \tag{11.3}$$

for all $\theta$ and $y = (s,a,s') \in \mathcal{Y}$. Then the update equation of Algorithm 9 can be written in the same form as Algorithm 2 with the additive noise $w_k$ being identically equal to zero. Let $\bar{F}(\theta) = \mathbb{E}_{\mu_Y}[F(\theta,Y)]$. We see that $\bar{F}(\theta) = 0$ is exactly the targeting equation (cf. Equation 11.1).

To apply Theorem Theorem 3.2.1, we first show in the following proposition that Assumptions 3.1.1 and 3.1.2 are satisfied in the context of $Q$-learning with linear function approximation.

**Proposition 11.2.1.** *Suppose that Assumptions 11.2.1 and 11.2.2 are satisfied, then we have the following results.*

*(1) The Markov chain $\{Y_k\}$ has a unique stationary distribution $\mu_Y$. In addition, we have $\max_{y\in\mathcal{Y}} \|P_{\pi_b}^{k+1}(y,\cdot) - \mu_Y(\cdot)\|_{TV} \leq C\sigma^k$ for all $k \geq 0$.*

*(2) The operator $F(\cdot,\cdot)$ satisfies*

*(a) $\|F(\theta_1,y) - F(\theta_2,y)\|_2 \leq 3\|\theta_1 - \theta_2\|_2$ for all $\theta_1, \theta_2 \in \mathbb{R}^d$ and $y \in \mathcal{Y}$.*

*(b) $\|F(\mathbf{0},y)\|_2 \leq 3$ for all $y \in \mathcal{Y}$.*

*(3) The equation $\bar{F}(\theta) = 0$ has a unique solution $\theta^*$, and we have*

$$(\theta - \theta^*)^\top(\bar{F}(\theta) - \bar{F}(\theta^*)) \leq -\frac{\kappa}{2}\|\theta - \theta^*\|_2^2,\ \forall\, \theta \in \mathbb{R}^d.$$

Similarly as in previous chapters, given precision $\delta > 0$, we define $t_\delta$ as the mixing time of the Markov chain $\{Y_k\}$ with precision $\delta > 0$. Observe that Proposition 11.2.1 (1) implies that there exists a constant $L_1 = \frac{\log(C/\sigma)}{\log(1/\sigma)}$ such that $t_\delta \leq L_1(\log(1/\delta)+1)$ for any $\delta > 0$.

We next use Theorem Theorem 3.2.1 to establish the finite-sample bounds of Algorithm 9. Let $c_1 = (\|\theta_0\|_2 + \|\theta_0 - \theta^*\|_2 + 1)^2$ and $c_2 = 1170(\|\theta^*\|_2+1)^2$. The following theorem is a direct implication of Theorem Theorem 3.2.1, hence we omit its proof.

**Theorem 11.2.1.** *Consider $\{\theta_k\}$ of Algorithm 9. Suppose that Assumptions 11.2.1 and 11.2.2 are satisfied, Then we have the following results.*

*(1) When $\alpha_k \equiv \alpha$ with $\alpha$ chosen such that $\alpha t_\alpha \leq \frac{\kappa}{2340}$, we have for all $k \geq t_\alpha$:*

$$\mathbb{E}[\|\theta_k - \theta^*\|_2^2] \leq c_1 \left(1 - \frac{\kappa\alpha}{2}\right)^{k-t_\alpha} + 2c_2 \frac{\alpha t_\alpha}{\kappa}.$$

*(2) When $\alpha_k = \alpha/(k+h)$, where $\alpha > 2/\kappa$ and $h$ is appropriatly chosen, there exists $K' > 0$ such that we have for all $k \geq K'$:*

$$\mathbb{E}[\|\theta_k - \theta^*\|_2^2] \leq c_1 \left(\frac{K'+h}{k+h}\right)^{\frac{\kappa\alpha}{2}} + \frac{16ec_2\alpha^2 L_1}{\kappa\alpha - 2} \frac{\left[\log\left(\frac{k+h}{\alpha}\right)+1\right]}{k+h}.$$

*(3) When $\alpha_k = \alpha/(k+h)^\xi$, where $\xi \in (0,1)$, $\alpha > 0$, and $h$ is appropriatly chosen, there exists $K' \geq [4\xi/(\kappa\alpha)]^{1/(1-\xi)}$ such that we have for all $k \geq K'$:*

$$\mathbb{E}[\|\theta_k - \theta^*\|_2^2] \leq c_1 e^{-\frac{\kappa\alpha}{2(1-\xi)}\left((k+h)^{1-\xi} - (K'+h)^{1-\xi}\right)} + \frac{8c_2\alpha L_1}{\kappa} \frac{\left[\log\left(\frac{k+h}{\alpha}\right)+1\right]}{(k+h)^\xi}.$$

Theorem 11.2.1 (1) is qualitatively similar to Corollary 3.2.1 (1) in that the iterates of $Q$-learning converge exponentially fast to a ball centered at $\theta^*$, and the size of the ball is proportional to $\alpha t_\alpha$. This agrees with results in [12, 40], where the popular TD-learning with linear function approximation algorithm was studied. Theorem 11.2.1 (2) suggests that for properly chosen diminishing stepsizes, the optimal convergence rate is roughly $O(\log(k)/k)$. The $\log(k)$ factor is a consequence of performing Markovian sampling of $\{(S_k, A_k)\}$.

### 11.2.3 Discussion

In this subsection, we take a closer look at Assumption 11.2.2 and especially Equation 11.2, which is made for the stability of the $Q$-learning with linear function approximation algorithm. First note that Equation 11.2 is equivalent to

$$\gamma^2\mathbb{E}_{\kappa_S}[\max_{a\in\mathcal{A}} Q_\theta(S,a)^2] < \mathbb{E}_{\kappa_S,\pi_b}[Q_\theta(S,A)^2] \tag{11.4}$$

for all nonzero $\theta$. The direction Equation 11.2 implying Equation 11.4 is trivial. As for the other direction, let

$$\kappa = -\max_{\theta:\|\theta\|_2=1}\{\gamma^2\mathbb{E}_{\kappa_S}[\max_{a\in\mathcal{A}} Q_\theta(S,a)^2] - \mathbb{E}_{\kappa_S,\pi_b}[Q_\theta(S,A)^2]\}.$$

By Weierstrass extreme value theorem [182], $\kappa$ is well-defined and strictly positive because it is the maximum of a continuous function over a compact set. This immediately gives Equation 11.2.

Similar assumptions on the behavior policy were also proposed in [180, 183]. Although the exact form of the conditions are different, they all follow the same spirit. That is, with a chosen Lyapunov function, the condition should enable us to show that the corresponding ODE

$$\dot{\theta}(t) = \bar{F}(\theta(t)) \tag{11.5}$$

of the $Q$-learning algorithm (cf. Algorithm 9) is globally asymptotically stable (GAS). We next briefly compare our condition to those proposed in [180, 183]. The condition in [180] (i.e., their Eq. (7)) implies

$$2\gamma^2\mathbb{E}_{\kappa_S}[(\max_{a\in\mathcal{A}} Q_\theta(S,a))^2] < \mathbb{E}_{\kappa_S,\pi_b}[Q_\theta(S,A)^2] \tag{11.6}$$

for all nonzero $\theta$ [1]. The RHS is the same for both Equation 11.6 and Equation 11.4. On the LHS, Equation 11.6 has an additional factor of 2, and the square is outside the max operator. Although they are similar, our condition and the condition proposed in [180] do not imply each other. As for the condition proposed in [183], while it is not clear if it is less restrictive than ours, it is shown that the condition in [183] implies the condition in [180] under more restrictive assumptions. However, [183] assumes i.i.d. sampling, and studies only the asymptotic convergence rather than finite-sample error bounds.

We next analyze how the discount factor, the basis vectors $\{\phi_i\}$, and the behavior policy $\pi_b$ impact Equation 11.4. In terms of the dependence on the discount factor, it is clear that Equation 11.4 is easier to satisfy for smaller discount factor. This agrees with our numerical simulations provided in the next subsection. Utility of smaller discount factors in RL was also noted in [184], albeit in a completely different context of generalization. To see the impact of the basis vectors and the behavior policy, consider the following two examples.

**Uni-Dimension Case.** Suppose that $d = 1$. That is, there is only one basis vector $\phi_1$, and the weight $\theta$ is a scalar. Equation 11.4 reduces to

$$\gamma^2 \mathbb{E}_{\kappa_S}[\max_{a\in\mathcal{A}} \phi(S,a)^2] < \mathbb{E}_{\kappa_S,\pi_b}[\phi(S,A)^2]. \tag{11.7}$$

Define

$$\begin{aligned} h^+ &= \mathbb{E}_{\kappa_S,\pi_b}[\gamma\phi(S,A)\max_{a'\in\mathcal{A}}\phi(S',a') - \phi(S,A)^2], \\ h^- &= \mathbb{E}_{\kappa_S,\pi_b}[\gamma\phi(S,A)\min_{a'\in\mathcal{A}}\phi(S',a') - \phi(S,A)^2], \end{aligned}$$

and $r_\pi = \mathbb{E}_{\kappa_S,\pi_b}[\phi(S,A)\mathcal{R}(S,A)]$. Then we have the following result.

**Proposition 11.2.2.** *Equation 11.7 implies $h^+ < 0$ and $h^- < 0$, and the following statements regarding the relation between the stability of ODE (cf. Equation 11.5) and the sign*

[1] The factor of 2 appears to be missing in [180].

*of $h^+$ and $h^-$ hold:*

$$\textit{ODE (cf. Equation 11.5) is GAS} \iff \begin{cases} h^+ < 0, h^- < 0, & \textit{when } r_\pi = 0, \\ h^+ < 0, h^- \leq 0, & \textit{when } r_\pi > 0, \\ h^+ \leq 0, h^- < 0, & \textit{when } r_\pi < 0. \end{cases}$$

Proposition 11.2.2 implies that Equation 11.7 is "almost necessary" for the GAS of the ODE given in Equation 11.5. Moreover, it is clear from Equation 11.7 that when $d = 1$, there always exists a behavior policy $\pi_b$ such that Equation 11.7 is satisfied. For example, $\pi_b(s) \in \arg\max_{a \in \mathcal{A}} \phi(s, a)^2$ is a feasible behavior policy.

**Full-Dimension Case.** Suppose that $d = |\mathcal{S}||\mathcal{A}|$, i.e., there is no dimension reduction at all. We want to emphasize that this is not equivalent to the tabular $Q$-learning. Even when $\Phi$ is a full-rank square matrix, the $Q$-learning with linear function approximation algorithm does not coincide with the tabular $Q$-learning algorithm. In fact, the divergence counterexample provided in [8] belongs to this setting. We show in the following proposition that, in the full-dimension case, Equation 11.4 is feasible in terms of the behavior policy $\pi_b$ only when the discount factor $\gamma$ is sufficiently small.

**Proposition 11.2.3.** *When $d = |\mathcal{S}||\mathcal{A}|$ and $\gamma^2 \geq 1/|\mathcal{A}|$, Equation 11.4 is infeasible for any behavior policy $\pi_b$.*

We now compare the results for the two extreme cases, i.e., $d = 1$ and $d = |\mathcal{S}||\mathcal{A}|$. We see that in the uni-dimensional case, Equation 11.4 implies a condition which is almost sufficient and necessary for the GAS of the equilibrium $\theta^*$ to ODE given in Equation 11.5. Moreover, there always exists a behavior policy $\pi_b$ satisfying Equation 11.4. However, in the full-dimensional case, Equation 11.4 is infeasible in terms of the behavior policy $\pi_b$ when $\gamma^2 \geq 1/|\mathcal{A}|$, which can usually happen in practice.

### 11.2.4 Numerical Simulations

In this subsection, we present numerical experiments to demonstrate the sufficiency of Equation 11.4, as well as the resulting convergence rates of the $Q$-learning with linear function approximation algorithm.

We begin by verifying the sufficiency of Equation 11.2. Let

$$\omega(\pi) = \min_{\{\theta:\|\theta\|_2=1\}} \frac{\mathbb{E}_{\kappa_S,\pi_b}[Q_\theta(S,A)^2]}{\mathbb{E}_{\kappa_S}[\max_{a\in\mathcal{A}} Q_\theta(S,a)^2]}. \tag{11.8}$$

Then Equation 11.2 is equivalent to $\omega(\pi) > \gamma^2$. One way to compute $\omega(\pi)$ is presented in Subsection 11.3.4.

In our simulation, we consider the divergent example of $Q$-learning with linear function approximation introduced in [8], which is an MDP with 7 states and 2 actions. To demonstrate the effectiveness of Equation 11.2 for the stability of $Q$-learning, in our first set of simulations, the reward function is set to zero. Since the reward function is identically zero, $Q^*$ is zero, implying $\theta^*$ is zero. We choose the behavior policy $\pi$ which takes each action with equal probability. In this case, we have $\omega(\pi) \approx 0.5$, giving the threshold for $\gamma$ to satisfy Equation 11.2 being $\omega(\pi)^{1/2} \approx 0.7$. In our simulation, we choose constant stepsize $\alpha = 0.01$, discount factor $\gamma \in \{0.7, 0.9, 0.97\}$, and plot $\|\theta_k\|_2$ as a function of the number of iterations $k$ in Figure 11.1. Here, $\theta_k$ converges when $\gamma = 0.7, 0.9$, but diverges when $\gamma = 0.97$. This demonstrates that Equation 11.2 is sufficient but not necessary for convergence. This also shows that when Equation 11.2 is satisfied, the counter-example from [8] converges.

To show the exponential convergence rate for using constant stepsize, we consider the convergence of $\theta_k$ when $\gamma = 0.7$ given in Figure 11.2, where we plot $\log \mathbb{E}[\|\theta_k\|_2^2]$ as a function of the number of iterations $k$. In this case, $\theta_k$ seems to converge geometrically, which agrees with Theorem 11.2.1 (1).

We next numerically verify the convergence rates of $Q$-learning with linear function

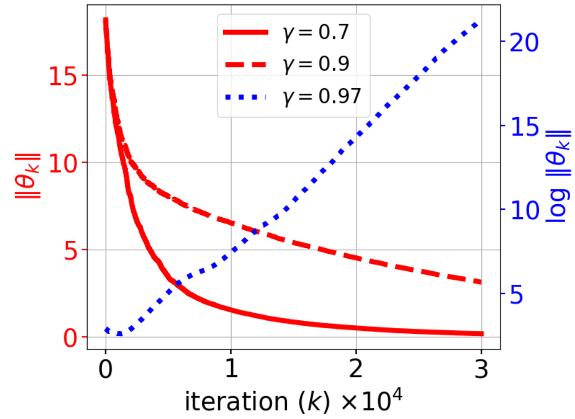

Figure 11.1: Convergence of $Q$-Learning with Linear Function Approximation for Different Discount Factor $\gamma$

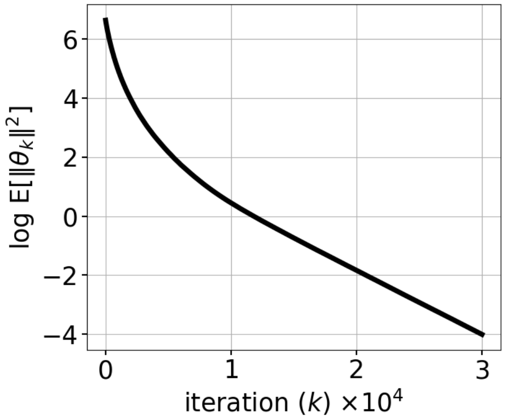

Figure 11.2: Exponentially Fast Convergence of $Q$-Learning with Linear Function Approximation for $\gamma = 0.7$

approximation for using diminishing stepsizes $\alpha_k = \alpha/(k+h)^\xi$. We use the same MDP model and behavior policy. The only difference is that the reward is no longer set to zero, but is sampled independently from a uniform distribution on $(0, 1)$ for all state-action pairs. The constant $\kappa$ given in Equation 11.2 is estimated by numerical optimization, and the discount factor $\gamma$ is set to be $0.7$ to ensure convergence. In Figure 11.3, we plot $\mathbb{E}[\|\theta_k - \theta^*\|_2^2]$ as a function of $k$ for $\xi \in \{0.4, 0.6, 0.8, 1\}$. In the case where $\xi = 1$, the constant coefficient $\alpha$ is chosen such that $\kappa\alpha \geq 2$ in order to achieve the optimal convergence rate. We see that the iterates converge for all $\xi \in (0, 1]$. Moreover, the larger the value of $\xi$ is, the faster $\theta_k$ converges.

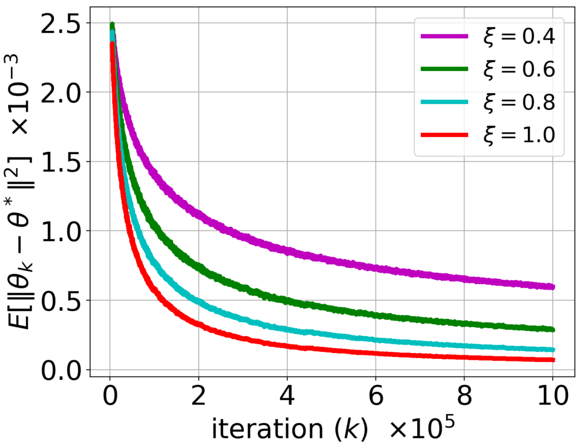

Figure 11.3: Convergence for Diminishing Stepsizes

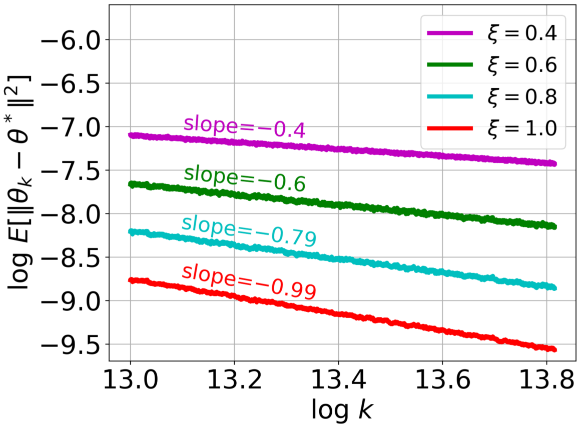

Figure 11.4: Asymptotic Convergence Rate for Diminishing Stepsizes

To further verify the convergence rates, we plot $\log \mathbb{E}[\|\theta_k - \theta^*\|_2^2]$ as a function of $\log k$ in Figure 11.4 and look at its asymptotic behavior. We see that the slope is approximately $-\xi$, which agrees with Theorem 11.2.1 (3).

## 11.3 Proof of All Theoretical Results

### 11.3.1 Proof of Proposition 11.2.1

(1) Under Assumption 11.2.2, it is easy to verify that the Markov chain $\{Y_k\}$ admits a unique stationary distribution, which we have denoted by $\mu_Y$. In view of the definition of $Y_k$, it is clear that $\mu_Y(s,a,s') = \kappa_S(s)\pi(a|s)P_a(s,s')$ for all $(s,a,s')$. Therefore, for any

$y = (s, a, s') \in \mathcal{Y}$, we have by definition of the total variation distance that

$$\begin{aligned}
\|P_{\pi_b}^{k+1}(y, \cdot) - \mu_Y(\cdot)\|_{\text{TV}} &= \frac{1}{2} \sum_{y_0 \in \mathcal{Y}} \left| P_{\pi_b}^{k+1}(y, y_0) - \mu_Y(y_0) \right| \\
&= \frac{1}{2} \sum_{(s_0, a_0, s_1) \in \mathcal{Y}} \left| P_{\pi_b}^{k}(s', s_0) - \kappa_S(s_0) \right| \pi(a_0|s_0) P_{a_0}(s_0, s_1) \\
&= \frac{1}{2} \sum_{(s_0, a_0, s_1) \in \mathcal{Y}} \left| P_{\pi_b}^{k}(s', s_0) - \kappa_S(s_0) \right| \\
&\leq \|P_{\pi_b}^{k}(s_0, \cdot) - \kappa_S(\cdot)\|_{\text{TV}} \\
&\leq C\sigma^k
\end{aligned}$$

for all $k \geq 0$. It follows that $\max_{y \in \mathcal{Y}} \|P_{\pi_b}^{k+1}(y, \cdot) - \mu_Y(\cdot)\|_{\text{TV}} \leq C\sigma^k$ for all $k \geq 0$.

(2) Using Cauchy-Schwarz inequality, and our assumption that $\|\phi(s, a)\|_1 \leq 1$ for all state-action pairs, we have for any $\theta_1, \theta_2$ and $y = (s, a, s')$ that

$$\begin{aligned}
&\|F(\theta_1, y) - F(\theta_2, y)\|_2 \\
={}& \|\phi(s, a)(\mathcal{R}(s, a) + \gamma \max_{a_1 \in \mathcal{A}} \phi(s', a_1)^\top \theta_1 - \phi(s, a)^\top \theta_1) \\
&- \phi(s, a)(\mathcal{R}(s, a) + \gamma \max_{a_2 \in \mathcal{A}} \phi(s', a_2)^\top \theta_2 - \phi(s, a)^\top \theta_2)\|_2 \\
\leq{}& \gamma \|\phi(s, a)(\max_{a_1 \in \mathcal{A}} \phi(s', a_1)^\top \theta_1 - \max_{a_2 \in \mathcal{A}} \phi(s', a_2)^\top \theta_2)\|_2 \\
&+ \|\phi(s, a)\phi(s, a)^\top (\theta_1 - \theta_2)\|_2 \\
\leq{}& \gamma |\max_{a_1 \in \mathcal{A}} \phi(s', a_1)^\top \theta_1 - \max_{a_2 \in \mathcal{A}} \phi(s', a_2)^\top \theta_2| + \|\theta_1 - \theta_2\|_2.
\end{aligned}$$

Since

$$\begin{aligned}
|\max_{a_1 \in \mathcal{A}} \phi(s', a_1)^\top \theta_1 - \max_{a_2 \in \mathcal{A}} \phi(s', a_1)^\top \theta_2| &\leq \max_{a' \in \mathcal{A}} |\phi(s', a')^\top (\theta_1 - \theta_2)| \\
&\leq \max_{a' \in \mathcal{A}} \|\phi(s', a')\| \|\theta_1 - \theta_2\|_2
\end{aligned} \tag{11.9}$$

$$\leq \|\theta_1 - \theta_2\|_2,$$

we have for any $\theta_1, \theta_2$ and $y$:

$$\|F(\theta_1, y) - F(\theta_2, y)\|_2 \leq (\gamma + 1)\|\theta_1 - \theta_2\|_2 \leq 2\|\theta_1 - \theta_2\|_2.$$

Moreover, we have

$$\|F(\mathbf{0}, y)\|_2 = \|\phi(s, a)\mathcal{R}(s, a)\|_2 \leq 2$$

for any $y = (s, a, s') \in \mathcal{Y}$.

(3) Using the fact that $\bar{F}(\theta^*) = 0$, we have

$$\begin{aligned}
&(\theta - \theta^*)^\top(\bar{F}(\theta) - \bar{F}(\theta^*)) \\
=& \gamma(\theta - \theta^*)^\top \mathbb{E}_{\kappa_S}[\phi(S, A)(\max_{a_1 \in \mathcal{A}} \phi(S', a_1)^\top \theta - \max_{a_2 \in \mathcal{A}} \phi(S', a_2)^\top \theta^*)] \\
&- \mathbb{E}_{\kappa_S, \pi_b}[(\phi(S, A)^\top(\theta - \theta^*))^2] \\
\leq& \gamma \mathbb{E}_{\kappa_S, \pi_b}[|\phi(S, A)^\top(\theta - \theta^*)| \max_{a' \in \mathcal{A}} |\phi(S', a')^\top(\theta - \theta^*)|] \\
&- \mathbb{E}_{\kappa_S, \pi_b}[(\phi(S, A)^\top(\theta - \theta^*))^2] \qquad (11.10) \\
\leq& \gamma \sqrt{\mathbb{E}_{\kappa_S, \pi_b}[(\phi(S, A)^\top(\theta - \theta^*))^2]} \sqrt{\mathbb{E}_{\kappa_S}[\max_{a \in \mathcal{A}}(\phi(S, a)^\top(\theta - \theta^*))^2]} \\
&- \mathbb{E}_{\kappa_S, \pi_b}[(\phi(S, A)^\top(\theta - \theta^*))^2]. \qquad (11.11)
\end{aligned}$$

Equation 11.10 follows from Equation 11.9. Equation 11.11 follows from the fact that when $S \sim \kappa_S$, we have $S' \sim \kappa_S$. For simplicity of notation, denote

$$A = \sqrt{\mathbb{E}_{\kappa_S, \pi_b}[(\phi(S, A)^\top(\theta - \theta^*))^2]}, \text{ and } B = \sqrt{\mathbb{E}_{\kappa_S}[\max_{a \in \mathcal{A}}(\phi(S, a)^\top(\theta - \theta^*))^2]}.$$

Since Assumption 11.2.2 gives $\gamma^2 B^2 - A^2 \leq -\kappa\|\theta - \theta^*\|^2$, we have

$$(\theta - \theta^*)^\top(\bar{F}(\theta) - \bar{F}(\theta^*)) \leq \frac{\gamma^2 B^2 - A^2}{\gamma B/A + 1} \leq -\frac{\kappa}{2}\|\theta - \theta^*\|^2.$$

### 11.3.2 Proof of Proposition 11.2.2

We first show that Equation 11.7 implies $h^+ < 0$, and $h^- < 0$. Note that Jensen's inequality implies

$$\begin{aligned}\mathbb{E}_{\kappa_S}[\max_{a'\in\mathcal{A}} \phi(S,a')^2] &= \mathbb{E}_{\kappa_S}\left\{\max\left[(\max_{a'\in\mathcal{A}}\phi(S,a'))^2, (\min_{a'\in\mathcal{A}}\phi(S,a'))^2\right]\right\} \\ &\geq \max\left\{\mathbb{E}_{\kappa_S}[(\max_{a'\in\mathcal{A}}\phi(S,a'))^2], \mathbb{E}_{\kappa_S}[(\min_{a'\in\mathcal{A}}\phi(S,a'))^2]\right\}. \qquad (11.12)\end{aligned}$$

Thus, using Equation 11.7, we have

$$\begin{aligned} h^+ &= \mathbb{E}_{\kappa_S,\pi_b}[\gamma\phi(S,A)\max_{a'\in\mathcal{A}}\phi(S',a')] - \mathbb{E}_{\kappa_S}[\phi(S,A)^2] \\ &= \mathbb{E}_{\kappa_S,\pi_b}[\gamma\phi(S,A)\max_{a'\in\mathcal{A}}\phi(S',a')] - \sqrt{\mathbb{E}_{\kappa_S,\pi_b}[\phi(S,A)^2]\mathbb{E}_{\kappa_S}[\phi(S,A)^2]} \\ &< \mathbb{E}_{\kappa_S,\pi_b}[\gamma\phi(S,A)\max_{a'\in\mathcal{A}}\phi(S',a')] - \gamma\sqrt{\mathbb{E}_{\kappa_S,\pi_b}[\max_{a'\in\mathcal{A}}\phi(S,a')^2]\mathbb{E}_{\kappa_S}[\phi(S,A)^2]} \\ &\leq 0, \end{aligned}$$

where the last inequality follows from Cauchy-Schwarz inquality and the fact that $S'$ and $S$ are equal in distribution if $S \sim \kappa_S$. Similarly, we also have $h^- < 0$.

We next prove the equivalence stated in Proposition 11.2.2. By definition of $h^+$ and $h^-$, in uni-dimensional case, the ODE given in Equation 11.5 can be equivalently written as

$$\dot{\theta}(t) = \begin{cases} h^+\theta(t) + r_\pi, & \theta(t) \geq 0, \\ h^-\theta(t) + r_\pi, & \theta(t) < 0. \end{cases}$$

In the case where $r_\pi = 0$, it is easy to see that the ODE is globally asymptotically stable if and only if $h^+, h^- < 0$. Now we assume without loss of generality that $r_\pi > 0$. The proof for the other case is entirely similar.

**Sufficiency:** We first note that $\theta^* = -r_\pi/h^+ > 0$. Let $W(\theta) = \frac{1}{2}(\theta - \theta^*)^2$ be a candidate Lyapunov function. It is clear that $W(\theta) \geq 0$ for all $\theta \in \mathbb{R}$, and $W(\theta) = 0$ if and only if $\theta = \theta^*$. Moreover, we have

$$\begin{aligned}\dot{W}(\theta(t)) &= (\theta(t) - \theta^*)\dot{\theta}(t) \\ &= \begin{cases} h^+(\theta(t) - \theta^*)^2, & \theta(t) \geq 0 \\ (\theta(t) - \theta^*)(h^-\theta(t) - h^+\theta^*), & \theta(t) < 0. \end{cases}\end{aligned}$$

It is clear that $\dot{W}(\theta(t)) < 0$ when $\theta(t) \in [0, \theta^*) \cup (\theta^*, \infty)$. For $\theta(t) < 0$, since $\theta(t) - \theta^* < 0$, $h^+\theta^* = -r_\pi < 0$, and $h^-\theta(t) \geq 0$, we must also have $\dot{W}(\theta(t)) < 0$. Therefore, the time derivative of the Lyapunov function $W(\theta)$ along the trajectory of the ODE is strictly negative when $\theta(t) \neq \theta^*$. It then follows from the Lyapunov stability theorem [91, 90] that $\theta^*$ is globally asymptotically stable.

**Necessity:** We prove by contradiction. Suppose that the equilibrium point $\theta^*$ is globally asymptotically stable, but $h^+ \geq 0$ or $h^- > 0$. Suppose that $h^+ \geq 0$. When $\theta(0) > \max(0, \theta^*)$, we have $\dot{\theta}(t) = h^+\theta(t) + r_\pi \geq r_\pi > 0$. It follows that $\theta(t) > \theta(0) > \theta^*$ for all $t \geq 0$, which contradict to the fact that $\theta^*$ is a globally asymptotically stable equilibrium point. Suppose that $h^- > 0$. When $\theta(0) < \min(\theta^*, -(1 + r_\pi)/h^-)$, we have $\dot{\theta}(t) = h^-\theta(t) + r_\pi \leq -1 < 0$. It follows that $\theta(t) < \theta(0) < \theta^*$ for all $t \geq 0$, which also contradict to the fact $\theta^*$ being globally asymptotically stable.

### 11.3.3 Proof of Proposition 11.2.3

When $d = |\mathcal{S}||\mathcal{A}|$, the feature matrix $\Phi$ is a square matrix. Define

$$\Theta_{s,a} = \text{span}\left(\{\phi(s',a')|(s',a') \in \mathcal{S} \times \mathcal{A},\ (s',a') \neq (s,a)\}\right)^{\perp}.$$

Note that $\Theta_{s,a}$ exists for all state-action pairs since $\Phi$ is full rank. Now for a given state-action pair $(s,a)$, let $\theta \neq 0$ be in $\Theta_{s,a}$, Equation 11.4 implies

$$\gamma^2 \kappa_S(s)(\phi(s,a)^\top \theta)^2 < \kappa_S(s)\pi(a|s)(\phi(s,a)^\top \theta)^2,$$

which further gives $\gamma^2 < \pi(a|s)$. Therefore, by running $(s,a)$ though all state-action pairs, we have $\gamma^2 < \min_{(s,a)\in\mathcal{S}\times\mathcal{A}} \pi(a|s) \leq \frac{1}{|\mathcal{A}|}$. Thus, if $\gamma^2 \geq 1/|\mathcal{A}|$, there is no behavior policy $\pi$ that satisfies Equation 11.4.

### 11.3.4 Computing $\omega(\pi)$

We here present one way to compute $\omega(\pi)$ for an MDP with a chosen policy $\pi$ when the underlying model is known. Before that, the following definitions are needed.

**Definition 11.3.1.** Let $D \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times|\mathcal{S}||\mathcal{A}|}$ be a diagonal matrix with diagonal entries being $\{\kappa_S(s)\pi_b(a|s)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$, and let $\Sigma = \Phi^\top D \Phi \in \mathbb{R}^{d\times d}$, where $\Phi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times d}$ is the feature matrix.

**Definition 11.3.2.** Let $\mathcal{B} = \mathcal{A}^n \subseteq \mathbb{R}^n$ be the set of all deterministic policies.

**Definition 11.3.3.** Let $H \in \mathbb{R}^{n\times n}$ be a diagonal matrix with diagonal entries $\{\kappa_S(s)\}_{s\in\mathcal{S}}$, and let $\Sigma_b = \Phi_b^\top H \Phi_b \in \mathbb{R}^{d\times d}$, where $\Phi_b \in \mathbb{R}^{n\times d}$ ($b \in \mathcal{B}$) is defined by:

$$\Phi_b = \begin{bmatrix} — & \phi(s_1,b)^\top & — \\ ... & ... & ... \\ — & \phi(s_n,b)^\top & — \end{bmatrix}.$$

We now compute $\omega(\pi)$ given in the following lemma. Let $\lambda_{\max}(\cdot)$ return the largest eigenvalue of a positive semi-definite matrix

**Lemma 11.3.1.** $\omega(\pi) = \min_{b\in\mathcal{B}} \left[1/\lambda_{\max}(\Sigma^{-1/2}\Sigma_b\Sigma^{-1/2})\right]$.

*Proof of Lemma 11.3.1*: Recall our definition for $\omega(\pi)$:

$$\omega(\pi) = \min_{\theta\neq 0} \frac{\sum_{s\in\mathcal{S}} \kappa_S(s) \sum_{a\in\mathcal{A}} \pi(a|s)(\phi(s,a)^\top\theta)^2}{\sum_{s\in\mathcal{S}} \kappa_S(s) \max_{a\in\mathcal{A}}(\phi(s,a)^\top\theta)^2}. \tag{11.13}$$

Let $f(\theta)$ be the numerator. Then we have

$$\begin{aligned} f(\theta) &= \sum_{s\in\mathcal{S}} \kappa_S(s) \sum_{a\in\mathcal{A}} \pi(a|s)(\phi(s,a)^\top\theta)^2 \\ &= \theta^\top\Phi^\top D\Phi\theta = \theta^\top\Sigma\theta. \end{aligned}$$

Since the diagonal entries of $D$ are all positive, and $\Phi$ is full column rank, the matrix $\Sigma$ is symmetric and positive definite. To represent the denominator of Equation 11.13 in a similar form, let

$$\begin{aligned} g(\theta,b) &= \sum_{s} \kappa_S(s)(\phi(s,b)^\top\theta)^2 \\ &= \theta^\top\Phi_b^\top H\Phi_b\theta = \theta^\top\Sigma_b\theta, \end{aligned}$$

where $b \in \mathcal{B}$. Since the columns of $\Phi_b$ can be dependent, the matrix $\Sigma_b$ is in general only symmetric and positive semi-definite. Using the definition of $f(\theta)$ and $g(\theta,b)$, we can rewrite $\omega(\pi)$ as

$$\begin{aligned} \omega(\pi) &= \min_{\theta\neq 0} \frac{f(\theta)}{\max_{b\in\mathcal{B}} g(\theta,b)} \\ &= \min_{\theta\neq 0} \min_{b\in\mathcal{B}} \frac{f(\theta)}{g(\theta,b)} \\ &= \min_{b\in\mathcal{B}} \min_{\theta\neq 0} \frac{f(\theta)}{g(\theta,b)}. \end{aligned}$$

Now since $\Sigma$ is positive definite, $\Sigma^{1/2}$ and $\Sigma^{-1/2}$ are both well-defined and positive definite, we have

$$\begin{aligned}
\min_{\theta \neq 0} \frac{f(\theta)}{g(\theta, b)} &= \left[\max_{\theta \neq 0} \frac{g(\theta, b)}{f(\theta)}\right]^{-1} \\
&= \left[\max_{\theta \neq 0} \frac{\theta^\top \Sigma_{\mu,b} \theta}{\theta^\top \Sigma_{\mu,\pi} \theta}\right]^{-1} \\
&= \left[\left(\max_{x \neq 0} \frac{\|\Sigma_{\mu,b}^{1/2} \Sigma_{\mu,\pi}^{-1/2} x\|_2}{\|x\|_2}\right)^2\right]^{-1} \\
&= \frac{1}{\lambda_{\max}(\Sigma_{\mu,\pi}^{-1/2} \Sigma_{\mu,b} \Sigma_{\mu,\pi}^{-1/2})}.
\end{aligned}$$

It follows that

$$\omega(\pi) = \min_{b \in \mathcal{B}} [1/\lambda_{\max}(\Sigma^{-1/2} \Sigma_b \Sigma^{-1/2})].$$

# CHAPTER 12

# TARGET NETWORK AND TRUNCATION OVERCOME THE DEADLY TRIAD IN $Q$-LEARNING

## 12.1 Introduction

In the previous chapter, we studied the classical semi-gradient $Q$-learning algorithm under linear function approximation, and established its finite-sample guarantees. However, there are several limitations with the results. A major limitation is that we require 11.2.2 (which is highly restrictive) to establish the finite-sample bounds. In addition, there is no characterization on where the limit point is relative to the optimal $Q$-function. Since classical $Q$-learning with linear function approximation has divergent counter-examples [8], the divergence issue is not because of the artifact of proof. This motivates us to design new variants of $Q$-learning under linear function approximation.

While theoretically unclear, it was empirically evident from [7] that the following three ingredients: *experience replay*, *target network*, and *truncation* together overcome the divergence of $Q$-learning with function approximation. In this chapter, we show theoretically that target network together with truncation is sufficient to provably stabilize $Q$-learning.

### 12.1.1 Main Contributions

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

- **Finite-Sample Guarantees.** We establish finite-sample guarantees of the output of $Q$-learning with target network and truncation to the optimal $Q$-function $Q^*$ up to a function approximation error. This is the first variant of $Q$-learning with linear function approximation that is provably stable (without needing strong assumptions), and uses a single trajectory of Markovian samples. The result implies an $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity, which matches with the sample complexity of $Q$-learning in the tabular setting, and is known to be optimal up to a logarithmic factor. The function approximation error in our finite-sample bound well captures the approximation power of the chosen function class. In the special case of tabular setting, or assuming the function class is closed under Bellman operator, our result implies asymptotic convergence in the mean-square sense to the optimal $Q$-function $Q^*$.

- **Broad Applicability.** In existing literature, to stabilize $Q$-learning with linear function approximation, one usually requires strong assumptions on the underlying MDP and/or the approximating function class. Those assumptions include but not limited to the function class being complete with respect to the Bellman operator, the MDP being linear (or close to linear), and a so-called strong negative drift assumption, etc. In this work, we do not require any of those assumptions. Specifically, our result holds as long as the policy used to collect samples enables the agent to sufficiently explore the state-action space, which is to some extent a necessary requirement to find an optimal policy in RL.

### 12.2 Related Literature

When using function approximation, the infamous deadly triad (i.e., function approximation, off-policy sampling, and bootstrapping) [1] appears in $Q$-learning, and the algorithm can be unstable even when linear function approximation is used. This is evident from the divergent MDP example constructed in [8]. Over the past $20$ years, there are many attempts to stabilize $Q$-learning with linear function approximation, which are summarised in the following.

**Strong Negative Drift Assumption.** The asymptotic convergence of $Q$-learning with linear function approximation was established in [180] under a “negative drift” assumption. Under similar assumptions, the finite-sample analysis of $Q$-learning, as well as its on-policy variant SARSA, was performed in [49, 185, 183, 186] for using linear function approximation, and in [187, 118] for using neural network approximation. However, such negative drift assumption is highly artificial, highly restrictive, and is impossible to satisfy unless the discount factor of the MDP is extremely small (see Chapter 11). In this chapter, we do not require such negative drift assumption or any of its variants to stabilize $Q$-learning with linear function approximation.

**Modifying the Problem Discount Factor.** Very recently, new convergent variants of $Q$-learning with linear function approximation were proposed in [188, 189], where target network was used in the algorithm. However, as we will see later in Section Subsection 12.5.3, target network alone is not sufficient to break the deadly triad. The reason that [188, 189] achieve convergence of $Q$-learning is by implicitly modifying the discount factor. In fact, the problem they are effectively solving is no longer the original MDP, but an MDP with a much smaller discount factor, which is the reason why their algorithms do not converge to the optimal $Q$-function $Q^*$ in the tabular setting. In this chapter we do not modify the original problem parameters to achieve stability, and in the special case of tabular RL, our algorithm converges to $Q^*$.

**The Greedy-GQ Algorithm.** A two time-scale variant of $Q$-learning with linear function approximation, known as Greedy-GQ, was proposed in [190]. The algorithm is designed based on minimizing the projected Bellman error using stochastic gradient descent. Although the Greedy-GQ algorithm is stable without needing the negative drift assumption, since the Bellman error is in general non-convex, Greedy-GQ algorithm can only guarantee convergence to stationary points. As a result, there are no performance guarantees on how well the limit point approximates the optimal $Q$-function $Q^*$. Although finite-sample bounds for Greedy-GQ were recently established in [191, 192, 193], due to the lack of global optimality, the finite-sample bounds were only on the gradient of the Bellman error rather than the distance to $Q^*$. In this work we provide finite-sample guarantees of our algorithm to the optimal $Q$-function $Q^*$ (up to a function approximation error).

**Fitted $Q$-Iteration and Its Variants.** Fitted $Q$-iteration is proposed in [127] as an offline variant of $Q$-learning. The finite-sample guarantees of fitted $Q$-iteration (or more generally fitted value iteration) were established in [194, 195]. More recently, [196] proposes a variant of batch RL algorithms called BVFT, where the authors establish an $\tilde{\mathcal{O}}(\epsilon^{-4})$ sample complexity under the realizability assumption. Notably, [194, 195] employed truncation technique to ensure the boundedness of the function approximation class. Such truncation technique dates back to [197]. We use the same truncation technique in this paper. In the special case of linear function approximation, $Q$-learning with target network can be viewed as an approximate way of implementing the fitted $Q$-iteration, where stochastic gradient descent was used as a way of performing such fitting. Compared to [194, 195], the main difference of this work is that our algorithm is implemented in an online manner, and is driven by a single trajectory of Markovian samples.

Another variant of fitted $Q$-iteration targeting finite horizon MDPs was proposed in [198] using a distribution shift checking oracle. However, [198] requires the approximating function class to contain the optimal $Q$-function, and only polynomial sample complexity, i.e., $\tilde{\mathcal{O}}(\epsilon^{-n})$ for some positive integer $n$, was established. In this work, we do not require $Q^*$ to be within our chosen function class, and our algorithm achieves the optimal $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity.

**Linear MDP Model.** In the special case that the MDP has linear (or approximately linear) transition dynamics and linear reward, convergent variants of $Q$-learning with linear function approximation were designed and analyzed in [199, 200, 201, 202, 203, 204]. In this work, we do not assume the underlying MDP is linear.

**Other Work.** [205] studies $Q$-learning with function approximation for deterministic MDPs. The Deep $Q$-Network was studied in [206]. See Appendix Subsection 12.7.2 for a more detailed discussion about the Deep $Q$-Network.

### 12.3 A Stable Algorithm Design

In this section, we present the algorithm of $Q$-learning with linear function approximation using target network and truncation. Before that, we introduce the truncation operator $\lceil \cdot \rceil$ in the following. For any vector $x$, let $\lceil x \rceil$ be the resulting vector of $x$ component-wisely truncated from both above and below at $r = 1/(1-\gamma)$, i.e., for each component $\lceil x \rceil_i$ of $\lceil x \rceil$, we have $\lceil x \rceil_i = r$ if $x_i > r$, $\lceil x \rceil_i = x_i$ if $x_i \in [-r, r]$, and $\lceil x \rceil_i = -r$ if $x_i < -r$. As will become clear later, the reason that we pick the truncation level $r$ to be $1/(1-\gamma)$ is that $\|Q^\pi\|_\infty \leq 1/(1-\gamma)$ for any policy $\pi$.

**Algorithm 10** $Q$-Learning with Linear Function Approximation: Target Network and Truncation

```
1: Input: Integers $T$, $K$, initializations $\theta_{t,0} = 0$ for all $t = 0, 1, ..., T-1$ and $\hat{\theta}_0 = 0$, behavior policy $\pi_b$
2: for $t = 0, 1, \cdots, T-1$ do
3:     for $k = 0, 1, \cdots, K-1$ do
4:         Sample $A_k \sim \pi_b(\cdot|S_k)$, $S_{k+1} \sim P_{A_k}(S_k, \cdot)$
5:         $\theta_{t,k+1} = \theta_{t,k} + \alpha_k \phi(S_k, A_k)(\mathcal{R}(S_k, A_k) + \gamma \max_{a' \in \mathcal{A}} \lceil \phi(S_{k+1}, a')^\top \hat{\theta}_t \rceil - \phi(S_k, A_k)^\top \theta_{t,k})$
6:     end for
7:     $\hat{\theta}_{t+1} = \theta_{t,K}$
8:     $S_0 = S_K$
9: end for
10: Output: $\hat{\theta}_T$
```

Several remarks are in order. First of all, Algorithm 10 is simple, easy to implement, and can be generalized to using arbitrary parametric function approximation in a straightforward manner (see Subsection 12.7.2). Second, in addition to $\{\theta_{t,k}\}$, we introduce $\{\hat{\theta}_t\}$ as the target network parameter, which is fixed in the inner loop where we update $\theta_{t,k}$, and is synchronized to the last iterate $\theta_{t,K}$ in the outer loop. Target network was first introduced in [7] for the design of the celebrated Deep $Q$-Network. Finally, before using the $Q$-function estimate associated with the target network in the inner-loop, we first truncate it at level $r$ (see line 5 of Algorithm 10).

Note that the location where we impose the truncation operator is different from that in the Deep $Q$-Network [7], where instead of only truncating $\phi(S_{k+1},a')^\top\hat{\theta}_t$, truncation is performed for the entire temporal difference $\mathcal{R}(S_k,A_k)+\gamma\max_{a'\in\mathcal{A}}\phi(S_{k+1},a')^\top\hat{\theta}_t-\phi(S_k,A_k)^\top\theta_{t,k}$. Similar truncation technique has been employed in [195, 130]. The reason that target network and truncation together ensure the stability of $Q$-learning with linear function approximation will be illustrated in detail in Section 12.5.

On the practical side, Algorithm 10 uses a *single trajectory* of Markovian samples generated by the behavior policy $\pi_b$ (see line 4 and line 8 of Algorithm 10). Therefore, the agent does not have to constantly reset the system. Our result can be easily generalized to the case where one uses time-varying behavior policy (i.e., the behavior policy is updated across the iterations of the target network) as long as it ensures sufficient exploration. For example, one can use the $\epsilon$-greedy policy or the Boltzmann exploration policy (aka. softmax policy) with respect to the $Q$-function estimate associated with the target network $Q_{\hat{\theta}_t}$ as the behavior policy.

## 12.4 Finite-Sample Guarantees

To present the finite-sample guarantees of Algorithm 10, we first formally state our assumption about the behavior policy $\pi_b$ and introduce necessary notation.

**Assumption 12.4.1.** The behavior policy $\pi_b$ satisfies $\pi_b(a|s)>0$ for all $(s,a)$, and induces an irreducible and aperiodic Markov chain $\{S_k\}$.

This assumption ensures that the behavior policy sufficient explores the state-action space, and is commonly imposed for value-based RL algorithms in the literature [92]. Note that Assumption 12.4.1 implies that the Markov chain $\{S_k\}$ admits a unique stationary distribution, denoted by $\kappa_S\in\Delta^{|\mathcal{S}|}$, and mixes at a geometric rate [48]. As a result, letting $t_\delta=\min\{k\geq 0\ :\ \max_{s\in\mathcal{S}}\|P^k_{\pi_b}(s,\cdot)-\kappa_S(\cdot)\|_{\text{TV}}\leq\delta\}$ be the mixing time of the Markov chain $\{S_k\}$ (induced by $\pi_b$) with precision $\delta>0$, then under Assumption 12.4.1 we have $t_\delta=\mathcal{O}(\log(1/\delta))$.

Under Assumption 12.4.1, the Markov chain $\{(S_k, A_k)\}$ also has a unique stationary distribution. Let $D \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times|\mathcal{S}||\mathcal{A}|}$ be a diagonal matrix with the unique stationary distribution of $\{(S_k, A_k)\}$ on its diagonal, i.e., $D((s,a),(s,a)) = \kappa_S(s)\pi_b(a|s)$ for all $(s,a)$. Moreover, let a norm $\|\cdot\|_D$ be defined by $\|x\|_D = (x^\top Dx)^{1/2}$. Denote $\lambda_{\min}$ as the minimum eigenvalue of the positive definite matrix $\Phi^\top D\Phi$.

Let $\mathcal{E}_{\text{approx}} := \sup_{Q:\|Q\|_\infty\leq r} \|\lceil \text{Proj}_{\mathcal{W}}\mathcal{H}(Q)\rceil - \mathcal{H}(Q)\|_\infty$, which captures the approximation power of the chosen function class. Denote $\hat{Q}_t = \lceil \Phi\hat{\theta}_t\rceil$, which is the truncated $Q$-function estimate associated with the target network $\hat{\theta}_t$.

We next present the finite-sample bounds. For ease of exposition, we only present the case where we use constant stepsize in the inner-loop of Algorithm 10, i.e., $\alpha_k \equiv \alpha$. The results for using various diminishing stepsizes are straightforward extensions.

**Theorem 12.4.1.** *Consider $\hat{\theta}_T$ of Algorithm 10. Suppose that Assumption 12.4.1 is satisfied, the constant stepsize $\alpha$ is chosen such that $\alpha \leq \frac{\lambda_{\min}(1-\gamma)^2}{130}$, and $K \geq t_\alpha + 1$. Then we have for any $T \geq 0$ that*

$$\begin{aligned}\mathbb{E}[\|\hat{Q}_T - Q^*\|_\infty] \leq \; & \underbrace{\gamma^T\|\hat{Q}_0 - Q^*\|_\infty}_{E_1\text{: Error due to fixed-point iteration}} + \underbrace{\frac{2(1-\lambda_{\min}\alpha)^{\frac{K-t_\alpha-1}{2}}}{\lambda_{\min}^{1/2}(1-\gamma)^2}}_{E_2\text{: Bias in the inner-loop}} \\ & + \underbrace{\frac{24\sqrt{\alpha(t_\alpha+1)}}{\lambda_{\min}(1-\gamma)^2}}_{E_3\text{: Variance in the inner-loop}} + \underbrace{\frac{\mathcal{E}_{\textit{approx}}}{1-\gamma}}_{E_4\text{: Function approximation error}}. \end{aligned} \tag{12.1}$$

*As a result, to obtain $\mathbb{E}[\|\hat{Q}_T - Q^*\|_\infty] \leq \epsilon + \frac{\mathcal{E}_{\textit{approx}}}{1-\gamma}$ for a given accuracy $\epsilon$, the sample complexity is*

$$\mathcal{O}\left(\frac{\log^2(1/\epsilon)}{\epsilon^2}\right)\tilde{\mathcal{O}}\left(\frac{1}{(1-\gamma)^4}\right).$$

*Remark.* While commonly used in existing literature studying RL with function approximation, it was argued in [175] that sample complexity is strictly speaking not well-defined when the asymptotic error is non-zero. Here we present the "sample complexity" in the same sense as in existing literature to enable a fair comparison.

Theorem 12.4.1 is by far the strongest result of $Q$-learning with linear function approximation in the literature in that it achieves the optimal $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity without needing strong assumptions, and meets all the requirements described in the beginning of this chapter.

In our finite-sample bound, the term $E_1$ goes to zero geometrically fast as $T$ goes to infinity. In fact, the term $E_1$ captures the error due to fixed-point iteration. That is, if we had a complete basis (hence no function approximation error), and were able to perform value iteration to solve the Bellman equation $Q^* = \mathcal{H}(Q^*)$ (hence no stochastic error), $E_1$ is the only error term.

The terms $E_2$ and $E_3$ represent the bias and variance in the inner-loop of Algorithm 10. Since the target network parameter $\hat{\theta}_t$ is fixed in the inner-loop, the update equation in Algorithm 10 line 5 can be viewed as a linear stochastic approximation algorithm under Markovian noise. When using constant stepsize, the bias goes to zero geometrically fast as $K$ goes to infinity but the variance is a constant proportional to $\sqrt{\alpha t_\alpha}$. Since geometric mixing implies $t_\alpha = \mathcal{O}(\log(1/\alpha))$, the term $\sqrt{\alpha t_\alpha}$ can be made arbitrarily small by using small enough constant stepsize. This agrees with existing literature studying linear stochastic approximation [12]. When using diminishing stepsizes with a suitable decay rate, one can easily show using 3.2.1 that both $E_1$ and $E_2$ go to zero at a rate of $\mathcal{O}(1/\sqrt{K})$, therefore the resulting sample complexity is the same as when using constant stepsize.

The term $E_4$ captures the error due to using function approximation. Recall that we define $\mathcal{E}_{\text{approx}} = \sup_{Q:\|Q\|_\infty \leq r} \|\lceil \text{Proj}_{\mathcal{W}} \mathcal{H}(Q) \rceil - \mathcal{H}(Q)\|_\infty$. Therefore to make the function approximation error small, one only needs to approximate the functions that are one-step reachable under the Bellman operator. In addition, using truncation also helps reducing the function approximation error to some extend since $\|\lceil \text{Proj}_{\mathcal{W}} \mathcal{H}(Q) \rceil - \mathcal{H}(Q)\|_\infty \leq \|\text{Proj}_{\mathcal{W}} \mathcal{H}(Q) - \mathcal{H}(Q)\|_\infty$ for any $Q$ such that $\|Q\|_\infty \leq r$. The $1/(1-\gamma)$ factor in $E_4$ also appears in TD-learning with linear function approximation [92], where it was shown to be not removable in general. Observe that $E_4$ vanishes (and hence we have convergence to $Q^*$) when (1) we are in the tabular setting, or (2) we use a complete basis (i.e., $\Phi$ being an invertible matrix), or (3) under the completeness assumption in existing literature, which requires $\mathcal{H}(Q) \in \mathcal{W}$ whenever $Q \in \mathcal{W}$. In existing work [188, 189], the algorithm does not converge to $Q^*$ even in the tabular setting (see Section 12.2).

## 12.5 The reason that Target Network and Truncation Stabilize $Q$-Learning

In the previous section, we presented the algorithm and the finite-sample guarantees. In this section, we elaborate in detail why target network and truncation together are enough to stabilize $Q$-learning.

**Summary.** We start with the classical semi-gradient $Q$-learning with linear function approximation algorithm in Subsection 12.5.1, which unfortunately is not necessarily stable, as evidenced by the divergent counter-example constructed in [8]. In Subsection 12.5.2, We show that by adding target network to $Q$-learning, the resulting algorithm successfully overcomes the divergence in the MDP example in [8]. However, beyond the example in [8], target network alone is not sufficient to break the deadly triad. In fact, we show in Subsection 12.5.3 that $Q$-learning with target network diverges for another MDP example constructed in [49]. In Subsection 12.5.4, we propose the last ingredient needed to achieve a stable algorithm design: truncation, which leads to Algorithm 10. By truncating at the right place in the $Q$-learning with target network algorithm, the resulting algorithm is provably stable and achieves the optimal $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity. The reason that truncation successfully stabilizes $Q$-learning is due to an insightful observation regarding the relation between truncation and projection.

### 12.5.1 Classical Semi-Gradient $Q$-Learning

We begin with the classical semi-gradient $Q$-learning with linear function approximation algorithm [11, 1], which we have studied in detail in the previous chapter. With a trajectory of samples $\{(S_k, A_k)\}$ collected under the behavior policy $\pi_b$ and an arbitrary initialization $\theta_0$, the semi-gradient $Q$-learning algorithm updates the parameter $\theta_k$ according to the following formula:

$$\theta_{k+1} = \theta_k + \alpha_k \phi(S_k, A_k) \left( \mathcal{R}(S_k, A_k) + \gamma \max_{a' \in \mathcal{A}} \phi(S_{k+1}, a')^\top \theta_k - \phi(S_k, A_k)^\top \theta_k \right). \tag{12.2}$$

Unfortunately, Equation 12.2 does not necessarily converge, as evidenced by the divergent example provided in [8]. The MDP example contructed in [8] has $7$ states and $2$ actions. To perform linear function approximation, $14$ linearly independent basis vectors are chosen. The important thing to notice about this example is that the number of basis vectors is equal to the size of the state-action space, i.e., $d = |\mathcal{S}||\mathcal{A}|$. Hence rather than doing function approximation, we are essentially doing a change of basis. Surprisingly even in this setting, Equation 12.2 diverges. Due to the divergence nature, [180, 49, 183] impose strong negative drift assumptions to ensure its stability.

By viewing Equation 12.2 as a stochastic approximation algorithm, the target equation it is trying to solve is $\mathbb{E}_{S_k \sim \kappa_S, A_k \sim \pi_b(\cdot|S_k)}[\phi(S_k, A_k)(\mathcal{R}(S_k, A_k) + \gamma \max_{a' \in \mathcal{A}} \phi(S_{k+1}, a')^\top \theta - \phi(S_k, A_k)^\top \theta)] = 0$. The previous equation can be written compactly using the Bellman optimality operator $\mathcal{H}(\cdot)$ and the diagonal matrix $D$ as

$$\Phi^\top D(\mathcal{H}(\Phi\theta) - \Phi\theta) = 0, \tag{12.3}$$

and is further equivalent to the fixed-point equation

$$\theta = \mathcal{H}_\Phi(\theta), \tag{12.4}$$

where the operator $\mathcal{H}_\Phi : \mathbb{R}^d \mapsto \mathbb{R}^d$ is defined by $\mathcal{H}_\Phi(\theta) = (\Phi^\top D \Phi)^{-1} \Phi^\top D \mathcal{H}(\Phi\theta)$. Equation 12.4 is closely related to the so-called projected Bellman equation. To see this, since

$\Phi$ is assumed to have linearly independent columns, Equation 12.4 is equivalent to

$$\Phi\theta = \Phi(\Phi^\top D\Phi)^{-1}\Phi^\top D\mathcal{H}(\Phi\theta) = \text{Proj}_{\mathcal{W}}\mathcal{H}(\Phi\theta), \tag{12.5}$$

where $\text{Proj}_{\mathcal{W}}$ denotes the projection operator onto the linear subspace $\mathcal{W}$ (which is spanned by the columns of $\Phi$) with respect to the weighted $\ell_2$-norm $\|\cdot\|_D$.

We next show that in the complete basis setting, i.e., $d = |\mathcal{S}||\mathcal{A}|$, which covers the Baird's counter-example as a special case, the operator $\mathcal{H}_\Phi(\cdot)$ is in fact a contraction mapping with $\theta^* = \Phi^{-1}Q^*$ being its unique fixed-point. This implies that the design of the classical semi-gradient $Q$-learning algorithm (cf. Equation 12.2) is flawed because *if it were designed as a stochastic approximation algorithm which is in effect performing fixed-point iteration to solve Equation 12.4, it would converge [24].* Instead, it was designed as a stochastic approximation algorithm based on Equation 12.3. While Equation 12.3 is equivalent to Equation 12.4, their corresponding stochastic approximation algorithms have different behavior in terms of their convergence or divergence.

To show the contraction property of $\mathcal{H}_\Phi(\cdot)$, first observe that in the complete basis setting we have $\mathcal{H}_\Phi(\theta) = (\Phi^\top D\Phi)^{-1}\Phi^\top D\mathcal{H}(\Phi\theta) = \Phi^{-1}\mathcal{H}(\Phi\theta)$. Let $\|\cdot\|_{\Phi,\infty}$ be a norm on $\mathbb{R}^d$ defined by $\|\theta\|_{\Phi,\infty} = \|\Phi\theta\|_\infty$ (the fact that it is indeed a norm can be easily verified). Then we have

$$\|\mathcal{H}_\Phi(\theta_1) - \mathcal{H}_\Phi(\theta_2)\|_{\Phi,\infty} = \|\mathcal{H}(\Phi\theta_1) - \mathcal{H}(\Phi\theta_2)\|_\infty \leq \gamma\|\Phi(\theta_1 - \theta_2)\|_\infty = \gamma\|\theta_1 - \theta_2\|_{\Phi,\infty}$$

for all $\theta_1, \theta_2 \in \mathbb{R}^d$, where the inequality follows from the Bellman optimality operator $\mathcal{H}(\cdot)$ being an $\ell_\infty$-norm contraction mapping. It follows that the operator $\mathcal{H}_\Phi(\cdot)$ is a contraction mapping with respect to $\|\cdot\|_{\Phi,\infty}$. Moreover, since $\mathcal{H}_\Phi(\theta^*) = \Phi^{-1}\mathcal{H}(\Phi\theta^*) = \Phi^{-1}\mathcal{H}(Q^*) = \Phi^{-1}Q^* = \theta^*$, the point $\theta^*$ is the unique fixed-point of the operator $\mathcal{H}_\Phi(\cdot)$. The previous analysis suggests that we should aim at designing $Q$-learning with linear function approximation algorithm as a fixed-point iteration (implemented in a stochastic manner due to sampling in RL) to solve Equation 12.4. The resulting algorithm would at least converge for the Baird's MDP example.

### 12.5.2 Introducing Target Network

We begin with the following fixed-point iteration for solving Equation 12.4:

$$\theta_{k+1} = (\Phi^\top D\Phi)^{-1}\Phi^\top D\mathcal{H}(\Phi\theta_k), \tag{12.6}$$

where we write $\mathcal{H}_\Phi(\cdot)$ explicitly in terms of $\Phi$, $D$, and $\mathcal{H}(\cdot)$. Equation 12.6 is what we would like to perform if we had complete information about the dynamics of the underlying MDP. The question is that if there is a stochastic variant of such fixed-point iteration that can be actually implemented in the RL setting where the transition probabilities and the stationary distribution are unknown. The answer is $Q$-learning with target network.

**Algorithm 11** $Q$-Learning with Linear Function Approximation: Target Network and No Truncation

```
1: Input: Integers $T$, $K$, initializations $\theta_{t,0} = 0$ for all $t = 0, 1, ..., T-1$ and $\hat{\theta}_0 = 0$,
   behavior policy $\pi_b$
2: for $t = 0, 1, \cdots, T-1$ do
3:     for $k = 0, 1, \cdots, K-1$ do
4:         Sample $A_k \sim \pi_b(\cdot|S_k)$, $S_{k+1} \sim P_{A_k}(S_k, \cdot)$
5:         $\theta_{t,k+1} = \theta_{t,k} + \alpha_k\phi(S_k, A_k)(\mathcal{R}(S_k, A_k) + \gamma \max_{a' \in \mathcal{A}} \phi(S_{k+1}, a')^\top\hat{\theta}_t -$
           $\phi(S_k, A_k)^\top\theta_{t,k})$
6:     end for
7:     $\hat{\theta}_{t+1} = \theta_{t,K}$
8:     $S_0 = S_K$
9: end for
10: Output: $\hat{\theta}_T$
```

We next elaborate on why Algorithm 11 can be viewed as a stochastic variant of the fixed-point iteration given in Equation 12.6. Consider the update equation (line 5) in the inner-loop of Algorithm 11. Since the target network is fixed in the inner-loop, the update equation in terms of $\theta_{t,k}$ is in fact a linear stochastic approximation algorithm for solving the following linear system of equations:

$$-\Phi^\top D\Phi\theta + \Phi^\top D\mathcal{H}(\Phi\hat{\theta}_t) = 0. \tag{12.7}$$

Since the matrix $-\Phi^\top D\Phi$ is negative definite, the asymptotic convergence of the inner-loop update follows from standard results in the literature [11]. Therefore, when the stepsize sequence $\{\alpha_k\}$ is appropriately chosen and $K$ is large, we expect $\theta_{t,K}$ to approximate the solution of Equation 12.7, i.e., $\theta_{t,K} \approx (\Phi^\top D\Phi)^{-1}\Phi^\top D\mathcal{H}(\Phi\hat{\theta}_t)$. Now in view of line 7 of Algorithm 11, the target network $\hat{\theta}_{t+1}$ is synchronized to $\theta_{t,K}$. Therefore $Q$-learning with target network is in effect performing a stochastic variant of the fixed-point iteration in Equation 12.6.

Note that on an aside, $Q$-learning with target network can be viewed as an online version of fitted $Q$-iteration. To see this, recall that in the linear function approximation setting, fitted $Q$-iteration updates the corresponding parameter $\{\tilde{\theta}_t\}$ iteratively according to

$$\tilde{\theta}_{t+1} = \arg\min\nolimits_{\tilde{\theta}\in\mathbb{R}^d} \frac{1}{|\mathcal{N}|}\sum_{(s,a,s')\in\mathcal{N}} \left(\phi(s,a)^\top\tilde{\theta} - \mathcal{R}(s,a) - \gamma\max_{a'\in\mathcal{A}}\phi(s',a')^\top\tilde{\theta}_t\right)^2, \tag{12.8}$$

where $\mathcal{N} = \{(s,a,s')\}$ is a batch dataset generated in an i.i.d. manner as follows: $s\sim\mu(\cdot)$, $a\sim\pi_b(\cdot|s)$, and $s'\sim P_a(s,\cdot)$. Observe that Equation 12.8 is an empirical version of

$$\tilde{\theta}_{t+1} = \arg\min\nolimits_{\tilde{\theta}\in\mathbb{R}^d}\|\Phi\tilde{\theta} - \mathcal{H}(\Phi\tilde{\theta}_t)\|_D^2. \tag{12.9}$$

In light of Equation 12.9, the inner-loop of Algorithm Algorithm 11 can be viewed as a stochastic gradient descent algorithm for solving the optimization problem in Equation 12.9 with a single trajectory of Markovian samples.

Revisiting Baird's counter-example (where $d = |\mathcal{S}||\mathcal{A}|$), recall that the fixed-point iteration (cf. Equation 12.6) reduces to $\theta_{k+1} = \Phi^{-1}\mathcal{H}(\Phi\theta_k) = \mathcal{H}_\Phi(\theta_k)$. Since the operator $\mathcal{H}_\Phi(\cdot)$ is a contraction mapping as shown in Subsection 12.5.1, the fixed-point iteration in Equation 12.6 provably converges. As a result, $Q$-learning with target network as a stochastic variant of the fixed-point iteration in Equation 12.6 also converges.

**Proposition 12.5.1.** *Consider Algorithm 11. Suppose that Assumption 12.4.1 is satisfied, the feature matrix* $\Phi$ *is a square matrix (i.e.,* $d = |\mathcal{S}||\mathcal{A}|$*),* $\alpha_k \equiv \alpha \leq \frac{\lambda_{\min}(1-\gamma)^2}{130}$*, and* $K \geq t_\alpha + 1$*. Then the sample complexity to achieve* $\mathbb{E}[\|\Phi\hat{\theta}_T - Q^*\|_\infty] < \epsilon$ *is* $\tilde{\mathcal{O}}(\epsilon^{-2})$*.*

To further verify the stability, we conduct numerical simulations for the MDP example constructed in [8]. As we see, while classical semi-gradient $Q$-learning with linear function approximation diverges in Figure 12.1 (which agrees with [8]), $Q$-learning with target network converges as shown in Figure 12.2.

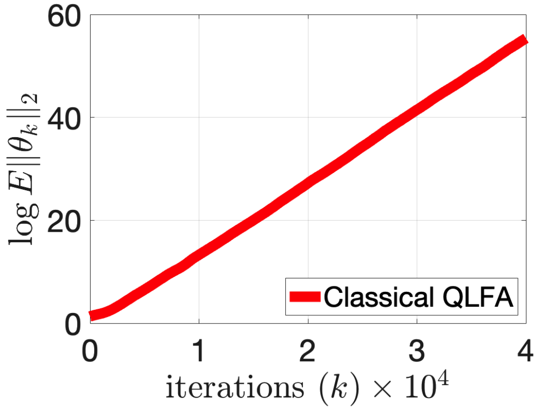

Figure 12.1: Classical Semi-Gradient $Q$-Learning

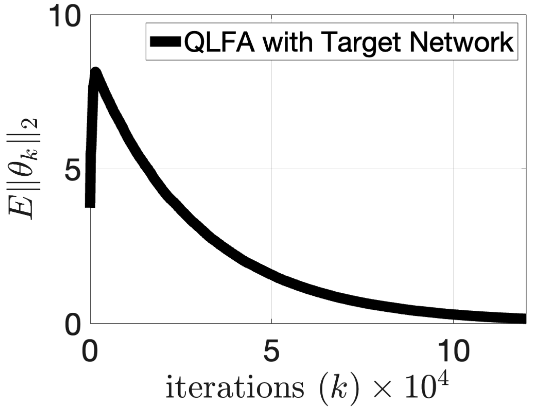

Figure 12.2: $Q$-Learning with Target Network

### 12.5.3 Insufficiency of Target Network

The reason that $Q$-learning with target network overcomes the divergence for Baird's MDP example is essentially that *the projected Bellman operator reduces to the regular Bellman operator (which is a contraction mapping) when we have a complete basis*. However, this is not the case in general. In the projected Bellman equation (cf. Equation 12.5), the Bellman operator $\mathcal{H}(\cdot)$ is a contraction mapping with respect to the $\ell_\infty$-norm $\|\cdot\|_\infty$, and the projection operator $\text{Proj}_{\mathcal{W}}$ is a non-expansive mapping with respect to the projection norm, in this case the weighted $\ell_2$-norm $\|\cdot\|_D$. Due to the norm mismatch, the composed operator $\text{Proj}_{\mathcal{W}}\mathcal{H}(\cdot)$ in general is not a contraction mapping with respect to any norm. This is the *fundamental reason* for the divergence of $Q$-learning with linear function approximation, and introducing target network alone does not overcome this issue, as evidenced by the following MDP example.

**Example 12.5.1.** Consider an MDP with state-space $\mathcal{S} = \{s_1, s_2\}$ and action-space $\mathcal{A} = \{a_1, a_2\}$. Regardless of the present state, taking action $a_1$ results in state $s_1$ with probability $1$, and taking action $a_2$ results in state $s_2$ with probability $1$. The reward function is defined as $\mathcal{R}(s_1, a_1) = 1$, $\mathcal{R}(s_1, a_2) = \mathcal{R}(s_2, a_1) = 2$, and $\mathcal{R}(s_2, a_2) = 4$. We construct the approximating linear sub-space with a single basis vector, which is given by $\Phi = [\phi(s_1, a_1), \phi(s_1, a_2), \phi(s_2, a_1), \phi(s_2, a_2)]^\top = [1, 2, 2, 4]^\top$. The behavior policy is to take each action with equal probability. In this example, after straightforward calculation, we have the following result.

**Lemma 12.5.1.** *Equation 12.4 is explicitly given by* $\theta = 1 + \frac{9\gamma}{10}\theta + \frac{3\gamma\theta}{10}(\mathbb{I}_{\{\theta \geq 0\}} - \mathbb{I}_{\{\theta < 0\}})$.

When the discount factor $\gamma$ is in the interval $(5/6, 1)$, for any positive initialization $\theta_0 > 0$, it is clear that performing fixed-point iteration to solve Equation 12.4 in this example leads to divergence. Since $Q$-learning with target network is a stochastic variant of such fixed-point iteration, it also diverges. Numerical simulations demonstrate that performing either classical semi-gradient $Q$-learning (cf. Figure 12.3) or $Q$-learning with target network (cf. Figure 12.4) leads to divergence for the MDP in Example 12.5.1.

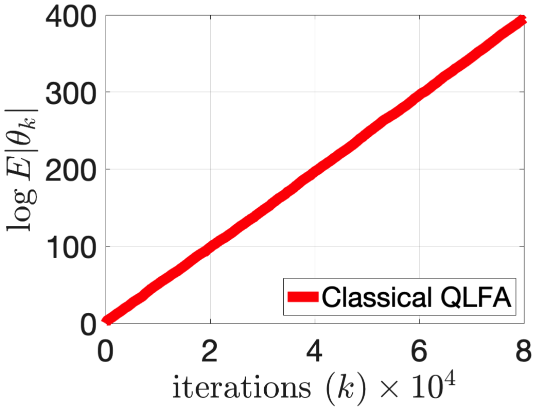

Figure 12.3: Classical Semi-Gradient $Q$-Learning

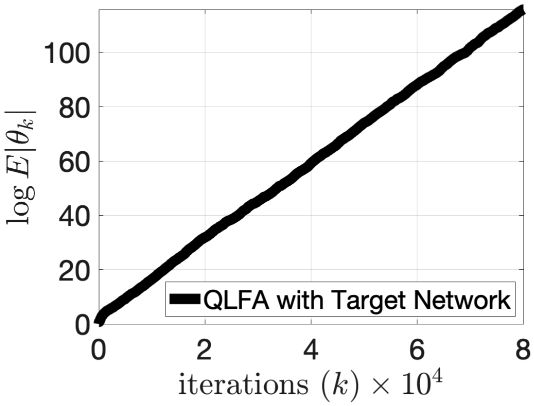

Figure 12.4: $Q$-Learning with Target Network

### 12.5.4 Truncation to the Rescue

The key idea that we use to further overcome the divergence of $Q$-learning with target network is truncation. Recall from the previous section that $Q$-learning with target network is trying to perform a stochastic variant of the fixed-point iteration in Equation 12.6, which can be equivalently written as

$$\tilde{Q}_{t+1} = \text{Proj}_{\mathcal{W}} \mathcal{H}(\tilde{Q}_t), \tag{12.10}$$

where we use $\tilde{Q}_t$ to denote the $Q$-function estimate associated with the target network $\hat{\theta}_t$, i.e., $\tilde{Q}_t = \Phi\hat{\theta}_t$. To motivate the truncation technique, we next analyze the update given in Equation 12.10, whose behavior in terms of stability aligns with the behavior of $Q$-learning with target network, as explained in the previous section. First note that Equation 12.10 is equivalent to

$$\tilde{Q}_{t+1} - Q^* = \mathcal{H}(\tilde{Q}_t) - \mathcal{H}(Q^*) + \text{Proj}_{\mathcal{W}}\mathcal{H}(\tilde{Q}_t) - \mathcal{H}(\tilde{Q}_t).$$

A simple calculation using triangle inequality, the contraction property of $\mathcal{H}(\cdot)$, and telescoping yields the following error bound of the iterative algorithm in Equation 12.10:

$$\|\tilde{Q}_{t+1} - Q^*\|_\infty \leq \gamma^{t+1}\|\tilde{Q}_0 - Q^*\|_\infty + \sum_{i=0}^{t}\gamma^{t-i}\underbrace{\|\text{Proj}_{\mathcal{W}}\mathcal{H}(\tilde{Q}_i) - \mathcal{H}(\tilde{Q}_i)\|_\infty}_{A_i}.$$

The problem with the previous analysis is that the term $A_i$ (which captures the error due to using linear function approximation) is not necessarily bounded unless using a complete basis or knowing in prior that $\{\tilde{Q}_t\}$ is always contained in a bounded set. The possibility that such function approximation error can be unbounded is an alternative explanation to the divergence of $Q$-learning with linear function approximation. This is true for arbitrary function approximation (including neural network) as well since it is in general not possible to uniformly approximate unbounded functions.

Suppose we are able to somehow control the size of the estimate $\tilde{Q}_t$ so that it is always contained in a bounded set. Then the term $A_i$ is guaranteed to be finite, and well captures the approximation power of the chosen function class. To achieve the boundedness of the associated $Q$-function estimate $\tilde{Q}_t$ of the target network, tracing back to Algorithm 11, a natural approach is to first project $\Phi\hat{\theta}_t$ onto the $\ell_\infty$-norm ball $B_r := \{Q \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \mid \|Q\|_\infty \leq r\}$ before using it as the target $Q$-function in the inner-loop, resulting in Algorithm 12 presented in the following.

In line 8 of Algorithm 12, the operator $\Pi_{B_r}$ stands for the projection onto the $\ell_\infty$-norm

**Algorithm 12** Impractical $Q$-Learning with Linear Function Approximation: Target Network and Projection

```
1: Input: Integers $T$, $K$, initializations $\theta_{t,0} = 0$ for all $t = 0, 1, ..., T-1$ and $\hat{\theta}_0 = 0$, behavior policy $\pi_b$
2: for $t = 0, 1, \cdots, T-1$ do
3:     for $k = 0, 1, \cdots, K-1$ do
4:         Sample $A_k \sim \pi_b(\cdot|S_k)$, $S_{k+1} \sim P_{A_k}(S_k, \cdot)$
5:         $\theta_{t,k+1} = \theta_{t,k} + \alpha_k \phi(S_k, A_k)(\mathcal{R}(S_k, A_k) + \gamma \max_{a' \in \mathcal{A}} \tilde{Q}_t(S_{k+1}, a') - \phi(S_k, A_k)^\top \theta_{t,k})$
6:     end for
7:     $\hat{\theta}_{t+1} = \theta_{t,K}$
8:     $\tilde{Q}_{t+1} = \Pi_{B_r} \Phi \hat{\theta}_{t+1}$
9:     $S_0 = S_K$
10: end for
11: Output: $\hat{\theta}_T$
```

ball $B_r$ with respect to some suitable norm $\| \cdot \|$. The specific norm $\| \cdot \|$ chosen to perform the projection turns out to be irrelevant as result of a key observation between truncation and projection.

Algorithm 12 although stabilizes the $Q$-function estimate $\tilde{Q}_t$, it is not implementable in practice. To see this, recall that the whole point of using linear function approximation is to avoid working with $|\mathcal{S}||\mathcal{A}|$ dimensional objects. However, to implement Algorithm 12 line 8, one has to first compute $\Phi \hat{\theta}_{t+1} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, and then project it onto $B_r$. Therefore, the last difficulty we need to overcome to achieve a stable algorithm design is to find a way to implement Algorithm 12 without working with $|\mathcal{S}||\mathcal{A}|$ dimensional objects. The solution relies on the following observation.

**Lemma 12.5.2.** *For any $x \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and any weighted $\ell_p$-norm $\| \cdot \|$ (the weights can be arbitrary and $p \in [1, \infty]$), we have $\lceil x \rfloor \in \arg\min_{y \in B_r} \|x - y\|$.*

*Remark.* Note that $\arg\min_{y \in B_r} \|x - y\|$ is in general a set because the projection may not be unique. As an example, observe that any point in the set $\{(x, 1) \mid x \in [-1, 1]\}$ is a projection of the point $(0, 2)$ onto the $\ell_\infty$-norm unit ball $\{(x, y) \mid x, y \in [-1, 1]\}$ with respect to the $\ell_\infty$-norm.

Lemma 12.5.2 states that for any $x \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, if we simply truncate $x$ at $r$, the resulting vector must belong to the projection set of $x$ onto the $\ell_\infty$-norm ball with radius $r$, for a wide class of projection norms. This seemingly simple but important result enables us to replace projection $\Pi_{B_r}(\cdot)$ by truncation $\lceil \cdot \rceil$ in line 8 of Algorithm 12:

$$\tilde{Q}_{t+1} = \Pi_{B_r} \Phi \hat{\theta}_{t+1} \quad \longrightarrow \quad \tilde{Q}_{t+1} = \lceil \Phi \hat{\theta}_{t+1} \rceil.$$

Unlike projection, truncation is a component-wise operation. Hence $\tilde{Q}_{t+1} = \lceil \Phi \hat{\theta}_{t+1} \rceil$ is equivalent to $\tilde{Q}_{t+1}(s,a) = \lceil \phi(s,a)^\top \hat{\theta}_{t+1} \rceil$ for all $(s,a)$.

The last issue is that we need to perform truncation for all state-action pairs $(s,a)$, which as illustrated earlier, violates the purpose of doing function approximation. However, observe that the target network is used in line 5 of Algorithm 12, where only the components of $\tilde{Q}_t$ visited by the sample trajectory is needed to perform the update. In light of this observation, instead of truncating $\phi(s,a)^\top \hat{\theta}_t$ for all $(s,a)$, we only need to truncate $\phi(S_{k+1}, a')^\top \hat{\theta}_t$ in Algorithm 12 line 5, which leads to our stable design of $Q$-learning with linear function approximation in Algorithm 10. The following proposition shows that target network and truncation together stabilized $Q$-learning with linear function approximation, and serves as a middle step to prove Theorem 12.4.1.

**Proposition 12.5.2.** *The following inequality holds:*

$$\begin{aligned} \mathbb{E}[\|\hat{Q}_T - Q^*\|_\infty] \leq \gamma^T \|\hat{Q}_0 - Q^*\|_\infty + \frac{\mathcal{E}_{approx}}{1-\gamma} \\ + \sum_{i=0}^{T-1} \gamma^{T-i-1} \mathbb{E}[\|\hat{Q}_{i+1} - \lceil Proj_{\mathcal{W}} \mathcal{H}(\hat{Q}_i) \rceil \|_\infty]. \end{aligned} \tag{12.11}$$

Because of truncation, the error due to using function approximation is bounded, and is captured by $\mathcal{E}_{\text{approx}}$. This is crucial to prevent the divergence of $Q$-learning with linear function approximation. The last term in Equation 12.11 captures the error in the inner-loop of Algorithm 10, and eventually contribute to the terms $E_2$ and $E_3$ in Equation 12.1.

Similar truncation technique was previously used in [195] to achieve a stable design of fitted $Q$-iteration. [195] studies fitted $Q$-iteration for general (possibly nonlinear) function approximation, where truncation is used to ensure the boundedness of the function approximation class.

Revisiting Example 12.5.1, where either semi-gradient $Q$-learning or $Q$-learning with target network diverges, Algorithm 10 converges as demonstrated in Figure 12.5. Moreover, observe that Algorithm 10 seems to converge to a positive scalar, which we denote by $\theta^*$. As a result, the policy $\pi$ induced greedily from $\Phi\theta^*$ is to always take action $a_2$. It can be easily verified that $\pi$ is indeed the optimal policy. This is an interesting observation since the optimal $Q$-function $Q^*$ in this case does not belong to the linear sub-space $\mathcal{W}$ (which is spanned by a single basis vector $(1, 2, 2, 4)^\top$). Nevertheless performing Algorithm 10 converges and the induced policy is optimal. Figure 12.6 shows that Algorithm 10 also converges for the Baird's MDP example.

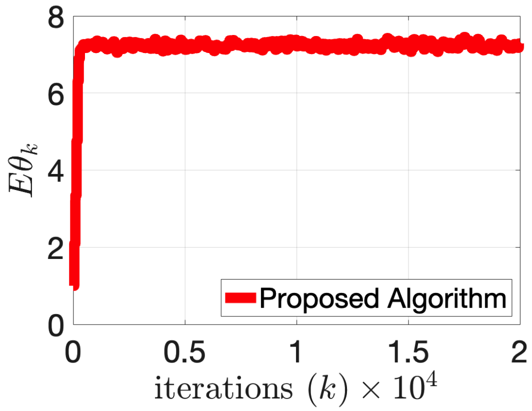

Figure 12.5: Algorithm 10 for Baird's MDP Example

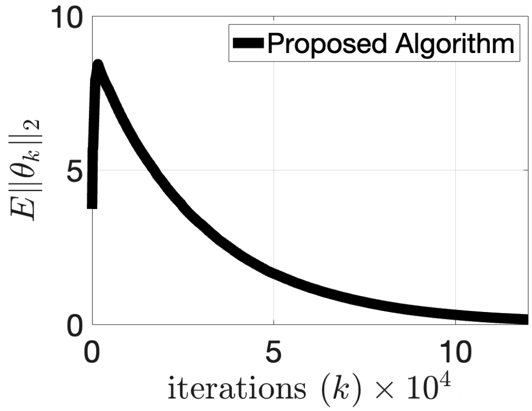

Figure 12.6: Algorithm 10 for Example 12.5.1

## 12.6 Proof of All Theoretical Results

### 12.6.1 Proof of Theorem 12.4.1

*Analysis of the Outer-Loop (Proof of Proposition 12.5.2)*

Recall that we denote $\hat{Q}_t = \lceil \Phi\hat{\theta}_t \rceil$. Using the fact that $Q^* = \mathcal{H}(Q^*)$, we have for any $t \geq 0$ that

$$\begin{aligned}\hat{Q}_t - Q^* &= \hat{Q}_t - \mathcal{H}(Q^*)\\ &= \mathcal{H}(\hat{Q}_{t-1}) - \mathcal{H}(Q^*) + \hat{Q}_t - \lceil \text{Proj}_{\mathcal{W}} \mathcal{H}(\hat{Q}_{t-1}) \rceil\\ &\quad + \lceil \text{Proj}_{\mathcal{W}} \mathcal{H}(\hat{Q}_{t-1}) \rceil - \mathcal{H}(\hat{Q}_{t-1}).\end{aligned}$$

It follows that

$$\begin{aligned}\|\hat{Q}_t - Q^*\|_\infty &\leq \|\mathcal{H}(\hat{Q}_{t-1}) - \mathcal{H}(Q^*)\|_\infty + \|\hat{Q}_t - \lceil \text{Proj}_{\mathcal{W}} \mathcal{H}(\hat{Q}_{t-1}) \rceil\|_\infty\\ &\quad + \|\lceil \text{Proj}_{\mathcal{W}} \mathcal{H}(\hat{Q}_{t-1}) \rceil - \mathcal{H}(\hat{Q}_{t-1})\|_\infty\\ &\leq \gamma \|\hat{Q}_{t-1} - Q^*\|_\infty + \|\hat{Q}_t - \lceil \text{Proj}_{\mathcal{W}} \mathcal{H}(\hat{Q}_{t-1}) \rceil\|_\infty + \mathcal{E}_{\text{approx}},\end{aligned}$$

where the last line follows from $\mathcal{H}(\cdot)$ being a $\gamma$-contraction mapping with respect to $\|\cdot\|_\infty$, and the definition of $\mathcal{E}_{\text{approx}}$.

Repeatedly using the previous inequality and then taking expectation on both sides of the resulting inequality, and we have for any $T \geq 0$:

$$\mathbb{E}[\|\hat{Q}_T - Q^*\|_\infty] \leq \gamma^T \|\hat{Q}_0 - Q^*\|_\infty + \sum_{i=0}^{T-1} \gamma^{T-i-1} \mathbb{E}[\|\hat{Q}_{i+1} - \lceil \text{Proj}_{\mathcal{W}} \mathcal{H}(\hat{Q}_i) \rceil \|_\infty] + \frac{\mathcal{E}_{\text{approx}}}{1-\gamma}. \tag{12.12}$$

This proves Proposition 12.5.2. The remaining task is to control $\mathbb{E}[\|\hat{Q}_{i+1} - \lceil \text{Proj}_{\mathcal{W}} \mathcal{H}(\hat{Q}_i) \rceil \|_\infty]$ for any $i = 0, ..., T-1$. First of all, since $\hat{Q}_t = \lceil \Phi \hat{\theta}_t \rceil = \lceil \Phi \theta_{t-1,K} \rceil$ and $\|\lceil Q_1 \rceil - \lceil Q_2 \rceil \|_\infty \leq \|Q_1 - Q_2\|_\infty$ for any $Q_1, Q_2 \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, we have

$$\mathbb{E}[\|\hat{Q}_{i+1} - \lceil \text{Proj}_{\mathcal{W}} \mathcal{H}(\hat{Q}_i) \rceil \|_\infty] \leq \mathbb{E}[\|\Phi \theta_{i,K} - \text{Proj}_{\mathcal{W}} \mathcal{H}(\hat{Q}_i)\|_\infty].$$

To further bound the RHS of the previous inequality, we need to analyze the inner-loop of Algorithm 10, which is done in the next section.

*Analysis of the Inner-Loop*

We begin by presenting the inner-loop of Algorithm 10.

**Algorithm 13** Inner-Loop of Algorithm 10

```
**Input:** Integer $K$, initialization $\theta_0 = 0$, target network $\hat{\theta}$, behavior policy $\pi_b$
**for** $k = 0, 1, \cdots, K-1$ **do**
    Sample $A_k \sim \pi_b(\cdot|S_k)$, $S_{k+1} \sim P_{A_k}(S_k, \cdot)$
    $\theta_{k+1} = \theta_k + \alpha_k \phi(S_k, A_k)(\mathcal{R}(S_k, A_k) + \gamma \max_{a' \in \mathcal{A}} \lceil \phi(S_{k+1}, a')^\top \hat{\theta} \rceil - \phi(S_k, A_k)^\top \theta_k)$
**end for**
**Output:** $\theta_K$
```

In view of the main update equation, Algorithm 13 is a Markovian linear stochastic approximation algorithm for solving the following linear system of equations:

$$-\Phi^\top D\Phi\theta + \Phi^\top D\mathcal{H}(\Phi\hat{\theta}) = 0.$$

Since the matrix $-\Phi^\top D\Phi$ is negative definite, the finite-sample guarantees follow from standard results in the literature [12, 49]. Specifically, we will apply Corollary 3.2.1 to establish the result.

To apply Corollary 3.2.1, we first rewrite the update equation in line 4 of Algorithm 13 in the form of the SA algorithm given in Algorithm 2. Then we verify that Assumptions 3.1.1 and 3.1.2 are satisfied.

- *Reformulation.* For any $k \geq 0$, let $Y_k = (S_k, A_k, S_{k+1})$, which is clearly a Markov chain with state-space given by $\mathcal{Y} = \{y = (s, a, s') \mid s \in \mathcal{S}, \pi_b(a|s) > 0, P_a(s, s') > 0\}$. Define the function $F : \mathbb{R}^d \times \mathcal{Y} \mapsto \mathbb{R}^d$ by

  $$F(\theta, s, a, s') = \phi(s, a)\left(\mathcal{R}(s, a) + \gamma \max_{a' \in \mathcal{A}} \lceil \phi(s', a')^\top \hat{\theta} \rceil - \phi(s, a)^\top \theta\right)$$

  for any $\theta \in \mathbb{R}^d$ and $y = (s, a, s') \in \mathcal{Y}$. Then the update equation of Algorithm 13 can be equivalently written as

  $$\theta_{k+1} = \theta_k + \alpha F(\theta_k, Y_k).$$

- *Verification of Assumption 3.1.2 (1).* For any $x_1, x_2 \in \mathbb{R}^d$ and $y = (s, a, s') \in \mathcal{Y}$, we have

  $$\begin{aligned} \|F(\theta_1, y) - F(\theta_2, y)\|_2 &= \|\phi(s, a)\phi(s, a)^\top(\theta_1 - \theta_2)\|_2 \\ &\leq \|\phi(s, a)\|_2^2 \|\theta_1 - \theta_2\|_2 \\ &\leq \|\theta_1 - \theta_2\|_2. \quad (\|\phi(s, a)\|_2 \leq \|\phi(s, a)\|_1 \leq 1 \text{ for all } (s, a)) \end{aligned}$$

Similarly, we have for any $y=(s,a,s')\in\mathcal{Y}$ that

$$\begin{aligned}\|F(0,y)\|_2&=\left\|\phi(s,a)\left(\mathcal{R}(s,a)+\gamma\max_{a'\in\mathcal{A}}\lceil\phi(s',a')^\top\hat{\theta}\rceil\right)\right\|_2\\&\leq\left(1+\frac{\gamma}{1-\gamma}\right)\|\phi(s,a)\|_2\\&\leq\frac{1}{1-\gamma}.\end{aligned}$$

- *Verification of Assumption 3.1.2 (2).* Under Assumption 12.4.1, the Markov chain $\{Y_k\}$ has a unique stationary distribution $\nu$, which is given by

$$\nu(s,a,s')=\mu(s)\pi(a|s)P_a(s,s')$$

for all $(s,a,s')\in\mathcal{Y}$. In addition, we have for any $y=(s,a,s')\in\mathcal{Y}$ that

$$\begin{aligned}\|P_{\pi_b}^k(y,\cdot)-\nu(\cdot)\|_{\text{TV}}&=\frac{1}{2}\sum_{(s_0,a_0,s_1)\in\mathcal{Y}}\left|P_{\pi_b}^{k-1}(s',s_0)-\mu(s_0)\right|\pi(a_0|s_0)P_{a_0}(s_0,s_1)\\&\leq\frac{1}{2}\sum_{s_0\in\mathcal{S}}\left|P_{\pi_b}^{k-1}(s',s_0)-\mu(s_0)\right|\\&\leq\max_{s\in\mathcal{S}}\|P_{\pi_b}^{k-1}(s,\cdot)-\mu(\cdot)\|_{\text{TV}}\\&\leq C\sigma^{k-1}.\end{aligned}$$

- *Verification of Assumption 3.1.1:* By definition of $F(\cdot,\cdot)$, we have

$$\bar{F}(\theta)=-\Phi^\top D\Phi\theta+\Phi^\top D\mathcal{H}(\lceil\Phi\hat{\theta}\rceil),$$

where we recall that $D\in\mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times|\mathcal{S}||\mathcal{A}|}$ is a diagonal matrix with diagonal entries $\{\mu(s)\pi_b(a|s)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$. Since $\Phi$ has linearly independent columns, the matrix $\Phi^\top D\Phi$ is invertible. Solving $\bar{F}(\theta)=0$ and we obtain $\theta^*=(\Phi^\top D\Phi)^{-1}\Phi^\top D\mathcal{H}(\lceil\Phi\hat{\theta}\rceil)$. Furthermore, note that the matrix $\Phi^\top D\Phi$ is positive definite, whose smallest eigenvalue is denoted by $\lambda_{\min}$. Therefore we have for any $\theta \in \mathbb{R}^d$:

$$\begin{aligned}(\theta-\theta^*)^\top \bar{F}(\theta) &= (\theta-\theta^*)^\top(\bar{F}(\theta)-\bar{F}(\theta^*))\\ &= -(\theta-\theta^*)^\top\Phi^\top D\Phi(\theta-\theta^*)\\ &\leq -\lambda_{\min}\|\theta-\theta^*\|_2^2.\end{aligned}$$

Now that Assumptions 3.1.1 and 3.1.2 are satisfied, Corollary 3.2.1 and we obtain for any $k \geq t_\alpha + 1$:

$$\begin{aligned}\mathbb{E}[\|\theta_k-\theta^*\|_2^2] \leq{}& (\|\theta^*\|_2+1)^2(1-\lambda_{\min}\alpha)^{k-t_\alpha-1}\\ &+\frac{130}{(1-\gamma)^2}((1-\gamma)\|\theta^*\|_2+1)^2\frac{\alpha(t_\alpha+1)}{\lambda_{\min}},\end{aligned} \tag{12.13}$$

where we used $\theta_0 = 0$ in Algorithm 13. The last step is to provide an upper bound on $\|\theta^*\|_2$. Note that

$$\begin{aligned}\|\theta^*\|_2 &= \frac{1}{\lambda_{\min}^{1/2}}\lambda_{\min}^{1/2}\|\theta^*\|_2\\ &\leq \frac{1}{\lambda_{\min}^{1/2}}\|\Phi\theta^*\|_D\\ &= \frac{1}{\lambda_{\min}^{1/2}}\|\Phi(\Phi^\top D\Phi)^{-1}\Phi^\top D\mathcal{H}(\lceil\Phi\hat{\theta}\rceil)\|_D && (\bar{F}(\theta^*)=0)\\ &= \frac{1}{\lambda_{\min}^{1/2}}\|\text{Proj}_{\mathcal{W}}\mathcal{H}(\lceil\Phi\hat{\theta}\rceil)\|_D\\ &\leq \frac{1}{\lambda_{\min}^{1/2}}\|\mathcal{H}(\lceil\Phi\hat{\theta}\rceil)\|_D && (\text{Proj}_{\mathcal{W}}(\cdot)\text{ is non-expansive with respect to }\|\cdot\|_D)\\ &\leq \frac{1}{\lambda_{\min}^{1/2}(1-\gamma)}\|\mathbf{1}\|_D && (-\tfrac{1}{1-\gamma}\mathbf{1}\leq\mathcal{H}(\lceil Q\rceil)\leq\tfrac{1}{1-\gamma}\mathbf{1}\text{ for any }Q\in\mathbb{R}^{|\mathcal{S}||\mathcal{A}|})\\ &= \frac{1}{\lambda_{\min}^{1/2}(1-\gamma)}.\end{aligned}$$

Substituting the previous upper bound we obtained for $\|\theta^*\|_2$ into Equation 12.13 and we finally have for all $k \geq t_\alpha + 1$:

$$\mathbb{E}[\|\theta_k - \theta^*\|_2^2] \leq \frac{4}{\lambda_{\min}(1-\gamma)^2}(1-\lambda_{\min}\alpha)^{k-t_\alpha-1} + \frac{520}{\lambda_{\min}^2(1-\gamma)^2}\alpha(t_\alpha+1). \tag{12.14}$$

*Putting Together*

We next combine the analysis of the outer-loop and the inner-loop to establish the overall finite-sample bounds of Algorithm 10. Denote $\theta_t^* = (\Phi^\top D\Phi)^{-1}\Phi^\top D\mathcal{H}(\hat{Q}_t)$. Note that we have $\Phi\theta_t^* = \text{Proj}_{\mathcal{W}}\mathcal{H}(\hat{Q}_t)$. Using the fact that $\|\cdot\|_\infty \leq \|\cdot\|_2$ and we obtain for any $0 \leq i \leq T$:

$$\begin{aligned}
\mathbb{E}[\|\Phi\theta_{i,K} - \text{Proj}_{\mathcal{W}}\mathcal{H}(\hat{Q}_i)\|_\infty] = \quad & \mathbb{E}[\|\Phi(\theta_{i,K} - \theta_i^*)\|_\infty] \\
&\leq \mathbb{E}[\|\Phi\|_\infty \|\theta_{i,K} - \theta_i^*\|_\infty] \\
&\leq \mathbb{E}[\|\theta_{i,K} - \theta_i^*\|_\infty] && (\|\phi(s,a)\|_1 \leq 1 \text{ for all } (s,a)) \\
&\leq \mathbb{E}[\|\theta_{i,K} - \theta_i^*\|_2] \\
&\leq \left(\mathbb{E}[\|\theta_{i,K} - \theta_i^*\|_2^2]\right)^{1/2} && (\text{Jensen's inequality}) \\
&\leq \left(\frac{4}{\lambda_{\min}(1-\gamma)^2}(1-\lambda_{\min}\alpha)^{K-t_\alpha-1} + \frac{520}{\lambda_{\min}^2(1-\gamma)^2}\alpha(t_\alpha+1)\right)^{1/2} && (\text{Equation 12.14}) \\
&\leq \frac{2}{\lambda_{\min}^{1/2}(1-\gamma)}(1-\lambda_{\min}\alpha)^{\frac{K-t_\alpha-1}{2}} + \frac{24}{\lambda_{\min}(1-\gamma)}\sqrt{\alpha(t_\alpha+1)},
\end{aligned}$$

where the last line follows from $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for any $a, b \geq 0$.

Substituting the previous inequality into Equation 12.12, and we obtain the overall finite-sample guarantees of Algorithm 10:

$$\begin{aligned}
\mathbb{E}[\|\hat{Q}_T - Q^*\|_\infty] \leq \gamma^T\|\hat{Q}_0 - Q^*\|_\infty &+ \frac{2}{\lambda_{\min}^{1/2}(1-\gamma)^2}(1-\lambda_{\min}\alpha)^{\frac{K-t_\alpha-1}{2}} \\
&+ \frac{24}{\lambda_{\min}(1-\gamma)^2}\sqrt{\alpha(t_\alpha+1)} + \frac{\mathcal{E}_{\text{approx}}}{1-\gamma}.
\end{aligned}$$

In view of the finite-sample guarantee, to obtain $\mathbb{E}[\|\hat{Q}_T - Q^*\|_\infty] \leq \epsilon + \frac{\mathcal{E}_{\text{approx}}}{1-\gamma}$ for a given accuracy $\epsilon$, the number of sample required is of the size

$$\mathcal{O}\left(\epsilon^{-2}\log^2(1/\epsilon)\right)\tilde{\mathcal{O}}\left(\frac{1}{(1-\gamma)^4}\right).$$

### 12.6.2 Proof of Proposition 12.5.1

The proof is identical to that of Theorem 12.4.1, and hence is omitted.

### 12.6.3 Proof of Proposition 12.5.2

See the analysis of the outer loop in Subsection 12.6.1.

### 12.6.4 Proof of Lemma 12.5.1

We first compute the transition probability matrix of the Markov chain $\{S_k\}$ under $\pi_b$. Since

$$P_{a_1} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad P_{a_2} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix},$$

and $\pi(a|s) = 1/2$ for any $a \in \{a_1, a_2\}$ and $s \in \{s_1, s_2\}$, we have $P_{\pi_b} = \frac{1}{2}I_2$. As a result, the unique stationary distribution $\mu$ of the Markov chain $\{S_k\}$ under $\pi_b$ is given by $\mu = (1/2, 1/2)$. Therefore, the matrix $D \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times|\mathcal{S}||\mathcal{A}|}$ (defined before Theorem 12.4.1) is given by $D = \frac{1}{4}I_4$. We next compute Equation 12.4 in this example. First of all, by definition of the Bellman operator we have for any $\theta \in \mathbb{R}$ that

$$\begin{aligned}
[\mathcal{H}(\Phi\theta)](s_1, a_1) &= \mathcal{R}(s_1, a_1) + \gamma\mathbb{E}[\max_{a'\in\mathcal{A}} \phi(S_{k+1}, a')\theta \mid S_k = s_1, A_k = a_1] \\
&= \mathcal{R}(s_1, a_1) + \gamma \max_{a'\in\mathcal{A}} \phi(s_1, a')\theta \\
&= \begin{cases} 1 + 2\gamma\theta, & \theta \geq 0, \\ 1 + \gamma\theta, & \theta < 0. \end{cases}
\end{aligned}$$

Similarly, we also have

$$[\mathcal{H}(\Phi\theta)](s_1, a_2) = \begin{cases} 2 + 4\gamma\theta, & \theta \geq 0, \\ 2 + 2\gamma\theta, & \theta < 0. \end{cases} \qquad [\mathcal{H}(\Phi\theta)](s_2, a_1) = \begin{cases} 2 + 2\gamma\theta, & \theta \geq 0, \\ 2 + \gamma\theta, & \theta < 0. \end{cases}$$

$$[\mathcal{H}(\Phi\theta)](s_2, a_2) = \begin{cases} 4 + 4\gamma\theta, & \theta \geq 0, \\ 4 + 2\gamma\theta, & \theta < 0. \end{cases}$$

Therefore, Equation 12.4 in the case of Example 12.5.1 is explicitly given by

$$\begin{aligned} \theta &= (\Phi^\top D\Phi)^{-1}\Phi^\top D\mathcal{H}(\Phi\theta) \\ &= \begin{cases} \frac{1}{25}\begin{bmatrix} 1 & 2 & 2 & 4 \end{bmatrix}\begin{bmatrix} 1 + 2\gamma\theta \\ 2 + 4\gamma\theta \\ 2 + 2\gamma\theta \\ 4 + 4\gamma\theta \end{bmatrix}, & \theta \geq 0 \\ \frac{1}{25}\begin{bmatrix} 1 & 2 & 2 & 4 \end{bmatrix}\begin{bmatrix} 1 + \gamma\theta \\ 2 + 2\gamma\theta \\ 2 + \gamma\theta \\ 4 + 2\gamma\theta \end{bmatrix}, & \theta < 0 \end{cases} \\ &= \begin{cases} 1 + \frac{6}{5}\gamma\theta, & \theta \geq 0, \\ 1 + \frac{3}{4}\gamma\theta, & \theta < 0, \end{cases} \\ &= 1 + \frac{9}{10}\gamma\theta + \frac{3}{10}\gamma\theta(\mathbb{I}_{\{\theta \geq 0\}} - \mathbb{I}_{\{\theta < 0\}}). \end{aligned}$$

### 12.6.5 Proof of Lemma 12.5.2

Let $\{\nu(s,a)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$ be any positive weights, and denote the weighted $\ell_p$-norm with weights $\{\nu(s,a)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$ by $\|\cdot\|_{\nu,p}$. For any $x\in\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, we have

$$\begin{aligned}\min_{y\in B_r}\|x-y\|_{\nu,p}&=\min_{y\in B_r}\left(\sum_{s,a}\nu(s,a)|x(s,a)-y(s,a)|^p\right)^{1/p}\\&=\left(\sum_{s,a}\nu(s,a)\min_{-r\leq y(s,a)\leq r}|x(s,a)-y(s,a)|^p\right)^{1/p}\\&=\left(\sum_{s,a}\nu(s,a)|x(s,a)-\lceil x(s,a)\rceil|^p\right)^{1/p}\\&=\|x-\lceil x\rceil\|_{\nu,p}.\end{aligned}$$

Therefore, we have $\lceil x\rceil\in\arg\min_{y\in B_r}\|x-y\|_{\nu,p}$.

## 12.7 Conclusion and Future Work

The work presented in this chapter makes fundamental contributions towards one of the most important open problems in RL: the behavior of $Q$-learning with function approximation. In particular, we design a stable $Q$-learning with linear function approximation using target network and truncation, which achieves the optimal $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity up to a function approximation error. Furthermore, the establishment of our results do not require strong assumptions (e.g. linear MDP, strong negative drift assumption, sufficiently small discount factor $\gamma$, etc.) as in related literature. There are two immediate future directions in this line of work. One is to improve the function approximation error, and the second is to extend the results of this work to using neural network approximation, i.e., the Deep $Q$-Network. The detailed plan is provided in the following.

### 12.7.1 Establishing the Asymptotic Convergence and Improving the Error Due to Function Approximation

Although Theorem 12.4.1 establishes the mean-square error bound of $Q$-learning with linear function approximation, due to the function approximation error, the bound does not imply asymptotic convergence. In light of our discussion in Section 12.5, suppose Algorithm 10 indeed converges (as $K, T \to \infty$ and $\alpha \to 0$) . The corresponding $Q$-function estimate of the output, i.e., $\hat{Q}_T = \lceil \Phi \hat{\theta}_T \rceil$, can only converge to the solution of the *truncated projected Bellman equation*:

$$Q = \lceil \text{Proj}_{\mathcal{W}} \mathcal{H}(Q) \rceil. \tag{12.15}$$

Unlike the projected Bellman equation (cf. Equation 12.5), which may not have a solution in general (see Example 12.5.1), since the truncated projected Bellman operator maps a compact set $B_r$ to itself, Equation 12.15 must have at least one solution according to the Brouwer fixed-point theorem. However, whether the solution to Equation 12.15 is unique or not is unclear. Therefore, it is also unclear if performing fixed-point iteration to solve Equation 12.15, or its stochastic variant (i.e., Algorithm 10) can actually leads to asymptotic convergence. Further investigating the truncated projected Bellman equation to show asymptotic convergence is one of our immediate future directions.

Suppose we were able to show the asymptotic convergence of Algorithm 10 to the unique solution of the truncated projected Bellman equation presented in Equation 12.15, denoted by $\bar{Q}$. Then, instead of establishing finite-sample bound of the form

$$\mathbb{E}[\|\hat{Q}_T - Q^*\|_\infty] \leq \underbrace{E_1 + E_2 + E_3}_{\text{go to zero as } K,T \to \infty \text{ and } \alpha \to 0} + \underbrace{E_4,}_{\text{Function approximation error}} \tag{12.16}$$

which is in fact what we did in this work, we would seek to establish the finite-sample bound of $\mathbb{E}[\|\hat{Q}_T - \bar{Q}\|_\infty]$, and separately characterize the difference between $Q^*$ and $\bar{Q}$.

This is in the same spirit of the seminal work [92], which studies the TD-learning with linear function approximation algorithm for policy evaluation. There are two advantages of this alternative approach. One is that the sample complexity of $\hat{Q}_T$ converging to $\bar{Q}$ is well-defined once we establish finite-sample convergence of $\mathbb{E}[\|\hat{Q}_T - \bar{Q}\|_\infty]$ to zero, while the sample complexity of convergence bounds of the form given Equation 12.16 is strictly speaking not well-defined because of the additive constant $E_4$, and may lead to erroneous result, as illustrated in [175] Appendix C. Second, this approach would enable us to reduce the function approximation error by removing the $\sup$ operator in $\mathcal{E}_{\text{approx}}$, i.e., from the current $\sup_{Q:\|Q\|_\infty \leq r} \|\lceil \text{Proj}_{\mathcal{W}} \mathcal{H}(Q) \rceil - \mathcal{H}(Q)\|_\infty$ to $\|\lceil \text{Proj}_{\mathcal{W}} \mathcal{H}(\bar{Q}) \rceil - \mathcal{H}(\bar{Q})\|_\infty$.

Although the lack of asymptotic convergence is a major limitation of this work, we want to point out that such limitation is present in almost all related literature on both value-space and policy-space methods whenever function approximation is used. To our knowledge, the only exception is [92] (as well as its follow-up work), where asymptotic convergence was established for TD-learning, and the limit was characterized as the unique solution of the projected Bellman equation. Other literature studying RL with function approximation either do not have asymptotic convergence [153], or have asymptotic convergence without knowing where the limit is [190].

### 12.7.2 The Deep $Q$ Network

The ultimate goal of this line of work is to provide theoretical understanding to the celebrated Deep $Q$-Network. We first present the extension of our Algorithm 10 to the setting where we use arbitrary function approximation (cf. Algorithm 14). Let $\mathcal{F} = \{f_\theta : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R} \mid \theta \in \mathbb{R}^d\}$ be a parametric function class (with parameter $\theta$). For example, $\mathcal{F}$ can be the set of functions representable by a certain neural network, and $\theta$ is the corresponding weight vector.

While the algorithm easily extends, the theoretical results do not. In particular, there are two major challenges.

**Algorithm 14** $Q$-Learning with Arbitrary Function Approximation

```
1: **Input:** Integers $T$, $K$, initialization $\theta_{0,0} = \hat{\theta}_0 = \mathbf{0}$
2: **for** $t = 0, 1, \cdots, T-1$ **do**
3:     **for** $k = 0, 1, \cdots, K-1$ **do**
4:         Sample $A_k \sim \pi_b(\cdot|S_k)$, observe $S_{k+1} \sim P_{A_k}(S_k, \cdot)$
5:         $\theta_{t,k+1} = \theta_{t,k} + \alpha_k \nabla f_{\theta_{t,k}}(S_k, A_k)(\mathcal{R}(S_k, A_k) + \gamma \max_{a' \in \mathcal{A}} \lceil f_{\hat{\theta}_t}(S_{k+1}, a') \rceil - f_{\theta_{t,k}}(S_k, A_k))$
6:     **end for**
7:     $\hat{\theta}_{t+1} = \theta_{t,K}$
8:     $S_0 = S_K$
9: **end for**
10: **Output:** $\hat{\theta}_T$
```

(1) With recent advances in deep learning [207], it is possible to explicitly characterize the function approximation error $\mathcal{E}_{\text{approx}}$ as a function of the hyper-parameters of the chosen neural network, such as the width, the number of layers, and the Hölder continuity parameter, etc.

(2) A more significant challenge is about the convergence of the inner-loop of Algorithm 14. Recall that in the linear function approximation setting, the inner loop (line 5 of Algorithm 10) can be viewed as a one-step Markovian stochastic approximation for solving the linear system of equations $-\Phi^\top D \Phi \theta + \Phi^\top D \mathcal{H}(\lceil \Phi \hat{\theta}_t \rceil) = 0$, or a one-step Markovian stochastic gradient descent for minimizing a quadratic objective $\|\Phi\theta - \mathcal{H}(\lceil \Phi \hat{\theta}_t \rceil)\|_D^2$ in terms of $\theta$. In this case, convergence to the global optimal of the inner-loop iterates is well established in the literature. Now consider using arbitrary function approximation in Algorithm 14. Although the inner-loop (line 5) is still performing a one-step Markovian stochastic gradient descent for minimizing $\|f_\theta - \mathcal{H}(\lceil f_{\hat{\theta}_t} \rceil)\|_D^2$ in terms of $\theta$, since the objective is now in general non-convex, the convergence to global optimal remains as a major theoretical open problem in the deep learning community.

Although the Deep $Q$-Network was previously studied in [206], their results rely on the following two assumptions: (1) the function approximation space is closed under the

Bellman operator, and (2) there exists an oracle that returns the global optimal of non-convex optimization problems. Under these two assumptions, both challenges described earlier are no longer present.

Once we explicitly characterize the function approximation error $\mathcal{E}_{\text{approx}}$, and show global convergence of the inner-loop, substituting the result into our analysis framework and we would be able to obtain finite-sample guarantees of Deep $Q$-Network, thereby achieving the ultimate goal of this line of research.

# REFERENCES

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

[2] D. Silver *et al.*, “Mastering the game of go with deep neural networks and tree search,” *nature*, vol. 529, no. 7587, p. 484, 2016.

[3] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[4] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A survey of autonomous driving: Common practices and emerging technologies,” *IEEE Access*, vol. 8, pp. 58 443–58 469, 2020.

[5] O. Gottesman *et al.*, “Guidelines for reinforcement learning in healthcare,” *Nature medicine*, vol. 25, no. 1, pp. 16–18, 2019.

[6] J. Degrave *et al.*, “Magnetic control of tokamak plasmas through deep reinforcement learning,” *Nature*, vol. 602, no. 7897, pp. 414–419, 2022.

[7] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[8] L. Baird, “Residual algorithms: Reinforcement learning with function approximation,” in *Machine Learning Proceedings 1995*, Elsevier, 1995, pp. 30–37.

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

[10] H. Robbins and S. Monro, “A stochastic approximation method,” *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

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

[12] R. Srikant and L. Ying, “Finite-Time Error Bounds For Linear Stochastic Approximation and TD Learning,” in *Conference on Learning Theory*, 2019, pp. 2803–2830.

[13] A. Harutyunyan, M. G. Bellemare, T. Stepleton, and R. Munos, “Q($\lambda$) with Off-Policy Corrections,” in *International Conference on Algorithmic Learning Theory*, Springer, 2016, pp. 305–320.

[14] D. Precup, R. S. Sutton, and S. P. Singh, "Eligibility Traces for Off-Policy Policy Evaluation," in *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp. 759–766.

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

[16] Z. Chen, S. T. Maguluri, S. Shakkottai, and K. Shanmugam, "A Lyapunov Theory for Finite-Sample Guarantees of Asynchronous $Q$-Learning and TD-Learning Variants," *Preprint arXiv:2102.01567*, 2021.

[17] C. J. Watkins and P. Dayan, "$Q$-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[18] R. S. Sutton, "Open Theoretical Questions in Reinforcement Learning," in *European Conference on Computational Learning Theory*, Springer, 1999, pp. 11–17.

[19] C. L. Beck and R. Srikant, "Error bounds for constant step-size $Q$-learning," *Systems & control letters*, vol. 61, no. 12, pp. 1203–1208, 2012.

[20] ——, "Improved upper bounds on the expected error in constant step-size $Q$-learning," in *2013 American Control Conference*, IEEE, 2013, pp. 1926–1931.

[21] E. Even-Dar and Y. Mansour, "Learning rates for $Q$-learning," *Journal of Machine Learning Research*, vol. 5, no. Dec, pp. 1–25, 2003.

[22] A. Dvoretzky, "On stochastic approximation," in *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, Berkeley, Calif.: University of California Press, 1956, pp. 39–55.

[23] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *Siam Review*, vol. 60, no. 2, pp. 223–311, 2018.

[24] J. N. Tsitsiklis, "Asynchronous stochastic approximation and $Q$-learning," *Machine learning*, vol. 16, no. 3, pp. 185–202, 1994.

[25] L. Espeholt *et al.*, "IMPALA: Scalable distributed deep-rl with importance weighted actor-learner architectures," in *International Conference on Machine Learning*, 2018, pp. 1407–1416.

[26] H. Küttler *et al.*, "Torchbeast: A PyTorch platform for distributed RL," *Preprint arXiv:1910.03552*, 2019.

[27] P. Mirowski *et al.*, "Learning to navigate in cities without a map," in *Advances in Neural Information Processing Systems*, 2018, pp. 2419–2430.

[28] A. Benveniste, M. Métivier, and P. Priouret, *Adaptive algorithms and stochastic approximations*. Springer Science & Business Media, 2012, vol. 22.

[29] T. Jaakkola, M. I. Jordan, and S. P. Singh, "Convergence of stochastic iterative dynamic programming algorithms," in *Advances in neural information processing systems*, 1994, pp. 703–710.

[30] H. J. Kushner and D. S. Clark, *Stochastic approximation methods for constrained and unconstrained systems*. Springer Science & Business Media, 2012, vol. 26.

[31] V. S. Borkar and S. P. Meyn, "The ODE method for convergence of stochastic approximation and reinforcement learning," *SIAM Journal on Control and Optimization*, vol. 38, no. 2, pp. 447–469, 2000.

[32] L. Ljung, "Analysis of recursive stochastic algorithms," *IEEE transactions on automatic control*, vol. 22, no. 4, pp. 551–575, 1977.

[33] V. S. Borkar, *Stochastic approximation: a dynamical systems viewpoint*. Springer, 2009, vol. 48.

[34] M. Benaim, "A dynamical system approach to stochastic approximations," *SIAM Journal on Control and Optimization*, vol. 34, no. 2, pp. 437–472, 1996.

[35] V. G. Yaji and S. Bhatnagar, "Analysis of stochastic approximation schemes with set-valued maps in the absence of a stability guarantee and their stabilization," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 1100–1115, 2019.

[36] P. Karmakar and S. Bhatnagar, "Stochastic approximation with iterate-dependent markov noise under verifiable conditions in compact state space with the stability of iterates not ensured," *IEEE Transactions on Automatic Control*, 2021.

[37] A. Ramaswamy and S. Bhatnagar, "Stability of stochastic approximations with "controlled markov" noise and temporal difference learning," *IEEE Transactions on Automatic Control*, vol. 64, no. 6, pp. 2614–2620, 2018.

[38] S. Bhatnagar and V. S. Borkar, "A two timescale stochastic approximation scheme for simulation-based parametric optimization," *Probability in the Engineering and Informational Sciences*, vol. 12, no. 4, pp. 519–531, 1998.

[39] ——, "Multiscale stochastic approximation for parametric optimization of hidden markov models," *Probability in the Engineering and Informational Sciences*, vol. 11, no. 4, pp. 509–522, 1997.

[40] J. Bhandari, D. Russo, and R. Singal, "A Finite Time Analysis of Temporal Difference Learning With Linear Function Approximation," in *Conference On Learning Theory*, 2018, pp. 1691–1692.

[41] G. Dalal, B. Szörényi, G. Thoppe, and S. Mannor, "Finite sample analyses for td$(0)$ with function approximation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[42] G. Thoppe and V. Borkar, "A concentration bound for stochastic approximation via alekseev's formula," *Stochastic Systems*, vol. 9, no. 1, pp. 1–26, 2019.

[43] G. Lan, *First-order and Stochastic Optimization Methods for Machine Learning*. Springer, 2020.

[44] E. Moulines and F. Bach, "Non-asymptotic analysis of stochastic approximation algorithms for machine learning," *Advances in neural information processing systems*, vol. 24, pp. 451–459, 2011.

[45] J. C. Duchi, A. Agarwal, M. Johansson, and M. I. Jordan, "Ergodic mirror descent," *SIAM Journal on Optimization*, vol. 22, no. 4, pp. 1549–1578, 2012.

[46] A. Beck, *First-order methods in optimization*. SIAM, 2017, vol. 25.

[47] S. Banach, "Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales," *Fund. math*, vol. 3, no. 1, pp. 133–181, 1922.

[48] D. A. Levin and Y. Peres, *Markov chains and mixing times*. American Mathematical Soc., 2017, vol. 107.

[49] Z. Chen, S. Zhang, T. T. Doan, J.-P. Clarke, and S. T. Maguluri, "Finite-Sample Analysis of Nonlinear Stochastic Approximation with Applications in Reinforcement Learning," *Preprint arXiv:1905.11425*, 2019.

[50] C. Guzmán and A. Nemirovski, "On lower complexity bounds for large-scale smooth convex optimization," *Journal of Complexity*, vol. 31, no. 1, pp. 1–14, 2015.

[51] A. Beck and M. Teboulle, "Smoothing and first order methods: A unified framework," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 557–580, 2012.

[52] P. Lax, *Linear Algebra*, ser. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts. Wiley, 1997, ISBN: 9780471111115.

[53] R. Vershynin, *High-dimensional probability: An introduction with applications in data science*. Cambridge university press, 2018, vol. 47.

[54] S. Shalev-Shwartz *et al.*, “Online learning and online convex optimization,” *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.

[55] L. Yu, K. Balasubramanian, S. Volgushev, and M. A. Erdogdu, “An Analysis of Constant Step Size SGD in the Non-convex Regime: Asymptotic Normality and Bias,” *Preprint arXiv:2006.07904*, 2020.

[56] R. S. Sutton, “Learning to predict by the methods of temporal differences,” *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.

[57] A. Dieuleveut, A. Durmus, and F. Bach, “Bridging the gap between constant step size stochastic gradient descent and markov chains,” *The Annals of Statistics*, vol. 48, no. 3, pp. 1348–1382, 2020.

[58] P. Bianchi, W. Hachem, and S. Schechtman, “Convergence of constant step stochastic gradient descent for non-smooth non-convex functions,” *Preprint arXiv:2005.08513*, 2020.

[59] A. Durmus, P. Jiménez, É. Moulines, and S. Salem, “On riemannian stochastic approximation schemes with fixed step-size,” in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2021, pp. 1018–1026.

[60] T. Sauer, “Numerical solution of stochastic differential equations in finance,” in *Handbook of computational finance*, Springer, 2012, pp. 529–550.

[61] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[62] W. Hu, C. J. Li, L. Li, and J.-G. Liu, “On the diffusion approximation of nonconvex stochastic gradient descent,” *Annals of Mathematical Sciences and Applications*, vol. 4, no. 1, 2019.

[63] Y. Xie, X. Wu, and R. Ward, “Linear convergence of adaptive stochastic gradient descent,” in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2020, pp. 1475–1485.

[64] P. Mertikopoulos, N. Hallak, A. Kavis, and V. Cevher, “On the almost sure convergence of stochastic gradient descent in non-convex problems,” *Preprint arXiv:2006.11144*, 2020.

[65] O. Shamir and T. Zhang, “Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes,” in *International conference on machine learning*, PMLR, 2013, pp. 71–79.

[66] X. Li and F. Orabona, “On the convergence of stochastic gradient descent with adaptive stepsizes,” in *The 22nd International Conference on Artificial Intelligence and Statistics*, PMLR, 2019, pp. 983–992.

[67] B. Fehrman, B. Gess, and A. Jentzen, “Convergence rates for the stochastic gradient descent method for non-convex objective functions,” *Journal of Machine Learning Research*, vol. 21, 2020.

[68] R. Gower, O. Sebbouh, and N. Loizou, “Sgd for structured nonconvex functions: Learning rates, minibatching and interpolation,” in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2021, pp. 1315–1323.

[69] V. Fabian, “On asymptotic normality in stochastic approximation,” *The Annals of Mathematical Statistics*, pp. 1327–1332, 1968.

[70] K. Hernandez and J. C. Spall, “Generalization of a result of fabian on the asymptotic normality of stochastic approximation,” *Automatica*, vol. 99, pp. 420–424, 2019.

[71] A. M. Devraj, A. Bušić, and S. Meyn, “Zap $Q$-Learning-A User’s Guide,” in *2019 Fifth Indian Control Conference (ICC)*, IEEE, 2019, pp. 10–15.

[72] H. Kesten *et al.*, “Accelerated stochastic approximation,” *Annals of Mathematical Statistics*, vol. 29, no. 1, pp. 41–59, 1958.

[73] Q. Li, C. Tai, and E. Weinan, “Stochastic modified equations and adaptive stochastic gradient algorithms,” in *International Conference on Machine Learning*, PMLR, 2017, pp. 2101–2110.

[74] Y. Feng, L. Li, and J.-G. Liu, “Semigroups of stochastic gradient descent and online principal component analysis: Properties and diffusion approximations,” *Communications in Mathematical Sciences*, vol. 16, no. 3, pp. 777–789, 2018.

[75] J. Yang, W. Hu, and C. J. Li, “On the fast convergence of random perturbations of the gradient flow,” *Asymptotic Analysis*, vol. 122, no. 3-4, pp. 371–393, 2021.

[76] J. Sirignano and K. Spiliopoulos, “Stochastic gradient descent in continuous time: A central limit theorem,” *Stochastic Systems*, vol. 10, no. 2, pp. 124–151, 2020.

[77] J. Latz, “Analysis of stochastic gradient descent in continuous time,” *Statistics and Computing*, vol. 31, no. 4, pp. 1–25, 2021.

[78] A. Eryilmaz and R. Srikant, “Asymptotically tight steady-state queue length bounds implied by drift conditions,” *Queueing Systems*, vol. 72, no. 3-4, pp. 311–359, 2012.

[79] D. Gamarnik and A. Zeevi, “Validity of heavy traffic steady-state approximations in generalized Jackson networks,” *The Annals of Applied Probability*, pp. 56–90, 2006.

[80] J. Harrison, “Brownian models of queueing networks with heterogeneous customer populations,” in *Stochastic Differential Systems, Stochastic Control Theory and Applications*, Springer, 1988, pp. 147–186.

[81] ——, “Heavy traffic analysis of a system with parallel servers: Asymptotic optimality of discrete review policies,” *Ann. App. Probab.*, pp. 822–848, 1998.

[82] J. M. Harrison and M. J. López, “Heavy traffic resource pooling in parallel-server systems,” *Queueing Systems*, pp. 339–368, 1999.

[83] A. L. Stolyar *et al.*, “Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic,” *The Annals of Applied Probability*, vol. 14, no. 1, pp. 1–53, 2004.

[84] R. J. Williams, “Diffusion approximations for open multiclass queueing networks: Sufficient conditions involving state space collapse,” *Queueing Systems Theory and Applications*, pp. 27–88, 1998.

[85] S. T. Maguluri and R. Srikant, “Heavy traffic queue length behavior in a switch under the maxweight algorithm,” *Stochastic Systems*, vol. 6, no. 1, pp. 211–250, 2016.

[86] S. T. Maguluri, S. K. Burle, and R. Srikant, “Optimal heavy-traffic queue length scaling in an incompletely saturated switch,” *Queueing Systems*, vol. 88, no. 3-4, pp. 279–309, 2018.

[87] D. Hurtado-Lange and S. T. Maguluri, “Transform methods for heavy-traffic analysis,” *Stochastic Systems*, vol. 10, no. 4, pp. 275–309, 2020.

[88] D. Hurtado-Lange, S. M. Varma, and S. T. Maguluri, “Logarithmic heavy traffic error bounds in generalized switch and load balancing systems,” *Preprint arXiv:2003.07821*, 2020.

[89] S. Mou and S. T. Maguluri, “Heavy traffic queue length behaviour in a switch under markovian arrivals,” *Preprint arXiv:2006.06150*, 2020.

[90] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice hall Upper Saddle River, NJ, 2002, vol. 3.

[91] W. M. Haddad and V. Chellaboina, *Nonlinear dynamical systems and control: a Lyapunov-based approach*. Princeton University Press, 2011.

[92] J. N. Tsitsiklis and B. Van Roy, “An analysis of temporal-difference learning with function approximation,” *IEEE transactions on automatic control*, vol. 42, no. 5, pp. 674–690, 1997.

[93] H. Flanders, “Differentiation under the integral sign,” *The American Mathematical Monthly*, vol. 80, no. 6, pp. 615–627, 1973.

[94] A. Muleshkov and T. Nguyen, “Easy proof of the Jacobian for the n-dimensional polar coordinates,” *Pi Mu Epsilon Journal*, vol. 14, pp. 269–273, 2016.

[95] M. Wenlong, F. Nicolas, W. Martin J., and B. Peter L., “Improved bounds for discretization of langevin diffusions: Near-optimal rates without convexity,” *https://arxiv.org/pdf/1907* 2019.

[96] R. Durrett, *Probability: theory and examples*. Cambridge university press, 2019, vol. 49.

[97] A. W. Van der Vaart, *Asymptotic statistics*. Cambridge university press, 2000, vol. 3.

[98] P. Diaconis and D. Freedman, “Iterated random functions,” *SIAM review*, vol. 41, no. 1, pp. 45–76, 1999.

[99] C. J. C. H. Watkins, “Learning from delayed rewards,” 1989.

[100] S. P. Singh and R. S. Sutton, “Reinforcement learning with replacing eligibility traces,” *Machine learning*, vol. 22, no. 1, pp. 123–158, 1996.

[101] M. J. Kearns and S. P. Singh, “Bias-Variance Error Bounds for Temporal Difference Updates,” in *COLT*, Citeseer, 2000, pp. 142–147.

[102] P. Dayan and T. J. Sejnowski, “TD($\lambda$) converges with probability 1,” *Machine Learning*, vol. 14, no. 3, pp. 295–301, 1994.

[103] Y. Zhao, D. Zeng, M. A. Socinski, and M. R. Kosorok, “Reinforcement learning strategies for clinical trials in nonsmall cell lung cancer,” *Biometrics*, vol. 67, no. 4, pp. 1422–1433, 2011.

[104] O. Gottesman *et al.*, “Interpretable off-policy evaluation in reinforcement learning by highlighting influential transitions,” in *International Conference on Machine Learning*, PMLR, 2020, pp. 3658–3667.

[105] C. Dann, L. Li, W. Wei, and E. Brunskill, “Policy certificates: Towards accountable reinforcement learning,” in *International Conference on Machine Learning*, PMLR, 2019, pp. 1507–1516.

[106] T. Mandel, Y.-E. Liu, S. Levine, E. Brunskill, and Z. Popovic, “Offline policy evaluation across representations with applications to educational games.,” in *AAMAS*, 2014, pp. 1077–1084.

[107] S. Gu, E. Holly, T. Lillicrap, and S. Levine, “Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates,” in *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2017, pp. 3389–3396.

[108] Y. Liu *et al.*, “Representation Balancing MDPs for Off-policy Policy Evaluation,” *Advances in Neural Information Processing Systems*, vol. 31, pp. 2644–2653, 2018.

[109] S. Levine, A. Kumar, G. Tucker, and J. Fu, “Offline reinforcement learning: Tutorial, review, and perspectives on open problems,” *Preprint arXiv:2005.01643*, 2020.

[110] P. W. Glynn and D. L. Iglehart, “Importance sampling for stochastic simulations,” *Management science*, vol. 35, no. 11, pp. 1367–1392, 1989.

[111] P. Cichosz and J. J. Mulawka, “Fast and efficient reinforcement learning with truncated temporal differences,” in *Machine Learning Proceedings 1995*, Elsevier, 1995, pp. 99–107.

[112] H. Van Seijen, A. R. Mahmood, P. M. Pilarski, M. C. Machado, and R. S. Sutton, “True online temporal-difference learning,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 5057–5096, 2016.

[113] S. P. Singh, T. Jaakkola, and M. I. Jordan, “Learning without state-estimation in partially observable Markovian decision processes,” in *Machine Learning Proceedings 1994*, Elsevier, 1994, pp. 284–292.

[114] P. Dayan, “The convergence of TD($\lambda$) for general $\lambda$,” *Machine learning*, vol. 8, no. 3-4, pp. 341–362, 1992.

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

[116] G. Thoppe and V. Borkar, “A concentration bound for stochastic approximation via Alekseev’s formula,” *Stochastic Systems*, vol. 9, no. 1, pp. 1–26, 2019.

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

[118] Q. Cai, Z. Yang, J. D. Lee, and Z. Wang, "Neural temporal-difference learning converges to global optima," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 11 315–11 326.

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

[120] H. R. Maei, "Gradient temporal-difference learning algorithms," 2011.

[121] R. S. Sutton *et al.*, "Fast gradient-descent methods for temporal-difference learning with linear function approximation," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 993–1000.

[122] R. S. Sutton, A. R. Mahmood, and M. White, "An emphatic approach to the problem of off-policy temporal-difference learning," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2603–2631, 2016.

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

[124] L. Tai and M. Liu, "A robot exploration strategy based on $Q$-learning network," in *2016 ieee international conference on real-time computing and robotics (rcar)*, IEEE, 2016, pp. 57–62.

[125] P. Mohamed Shakeel, S. Baskar, V. Sarma Dhulipala, S. Mishra, and M. M. Jaber, "Maintaining security and privacy in health care system using learning based deep-$Q$-networks," *Journal of medical systems*, vol. 42, no. 10, pp. 1–10, 2018.

[126] G. A. Rummery and M. Niranjan, *On-line $Q$-learning using connectionist systems*. Citeseer, 1994, vol. 37.

[127] D. Ernst, P. Geurts, and L. Wehenkel, "Tree-based batch mode reinforcement learning," *Journal of Machine Learning Research*, vol. 6, pp. 503–556, 2005.

[128] A. M. Devraj and S. Meyn, "Zap $Q$-learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 2235–2244.

[129] C. Szepesvári *et al.*, "The asymptotic convergence-rate of $Q$-learning," in *NIPS*, Citeseer, vol. 10, 1997, pp. 1064–1070.

[130] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan, "Is $Q$-learning provably efficient?" In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 4868–4878.

[131] G. Li, C. Cai, Y. Chen, Y. Gu, Y. Wei, and Y. Chi, "Tightening the dependence on horizon in the sample complexity of q-learning," in *International Conference on Machine Learning*, PMLR, 2021, pp. 6296–6306.

[132] M. J. Wainwright, "Stochastic approximation with cone-contractive operators: Sharp $\ell_\infty$-bounds for $Q$-learning," *Preprint arXiv:1905.06265*, 2019.

[133] G. Qu and A. Wierman, "Finite-Time Analysis of Asynchronous Stochastic Approximation and $Q$-Learning," in *Conference on Learning Theory*, PMLR, 2020, pp. 3185–3205.

[134] G. Li, Y. Wei, Y. Chi, Y. Gu, and Y. Chen, "Sample Complexity of Asynchronous $Q$-Learning: Sharper Analysis and Variance Reduction," in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 7031–7043.

[135] J. N. Tsitsiklis and B. Van Roy, "Average cost temporal-difference learning," *Automatica*, vol. 35, no. 11, pp. 1799–1808, 1999.

[136] D. P. Bertsekas and H. Yu, "Projected equation methods for approximate solution of large linear systems," *Journal of Computational and Applied Mathematics*, vol. 227, no. 1, pp. 27–50, 2009.

[137] S. Khodadadian, T. T. Doan, S. T. Maguluri, and J. Romberg, "Finite Sample Analysis of Two-Time-Scale Natural Actor-Critic Algorithm," *Preprint arXiv:2101.10506*, 2021.

[138] Z. Chen, S. T. Maguluri, S. Shakkottai, and K. Shanmugam, "Finite-Sample Analysis of Off-Policy TD-Learning via Generalized Bellman Operators," *Preprint arXiv:2106.12729*, 2021.

[139] A. Lazaric, M. Ghavamzadeh, and R. Munos, "Finite-sample analysis of least-squares policy iteration," *Journal of Machine Learning Research*, vol. 13, pp. 3041–3074, 2012.

[140] R. S. Sutton, C. Szepesvári, and H. R. Maei, "A convergent $\mathcal{O}(n)$ algorithm for off-policy temporal-difference learning with linear function approximation," *Advances in neural information processing systems*, vol. 21, no. 21, pp. 1609–1616, 2008.

[141] Z. Chen, S. Khodadadian, and S. T. Maguluri, "Finite-sample analysis of off-policy natural actor-critic with linear function approximation," *Preprint arXiv:2105.12540*, 2021.

[142] S. Ma, Y. Zhou, and S. Zou, "Variance-reduced off-policy TDC learning: Non-asymptotic convergence analysis," *Preprint arXiv:2010.13272*, 2020.

[143] Y. Wang, S. Zou, and Y. Zhou, "Finite-Sample Analysis for Two Time-scale Non-linear TDC with General Smooth Function Approximation," *Preprint arXiv:2104.02836*, 2021.

[144] J. Peters and S. Schaal, "Natural actor-critic," *Neurocomputing*, vol. 71, no. 7-9, pp. 1180–1190, 2008.

[145] T. Degris, M. White, and R. Sutton, "Off-Policy Actor-Critic," in *International Conference on Machine Learning*, 2012.

[146] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in neural information processing systems*, Citeseer, 2000, pp. 1008–1014.

[147] L. Wang, Q. Cai, Z. Yang, and Z. Wang, "Neural Policy Gradient Methods: Global Optimality and Rates of Convergence," in *International Conference on Learning Representations*, 2019.

[148] S. Qiu, Z. Yang, J. Ye, and Z. Wang, "On the finite-time convergence of actor-critic algorithm," in *Optimization Foundations for Reinforcement Learning Workshop at Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[149] H. Kumar, A. Koppel, and A. Ribeiro, "On the Sample Complexity of Actor-Critic Method for Reinforcement Learning with Function Approximation," *Preprint arXiv:1910.08412*, 2019.

[150] H. R. Maei, "Convergent actor-critic algorithms under off-policy training and function approximation," *Preprint arXiv:1802.07842*, 2018.

[151] S. Zhang, B. Liu, H. Yao, and S. Whiteson, "Provably convergent two-timescale off-policy actor-critic with function approximation," in *International Conference on Machine Learning*, PMLR, 2020, pp. 11 204–11 213.

[152] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee, "Natural actor–critic algorithms," *Automatica*, vol. 45, no. 11, pp. 2471–2482, 2009.

[153] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, "On the theory of policy gradient methods: Optimality, approximation, and distribution shift," *Journal of Machine Learning Research*, vol. 22, no. 98, pp. 1–76, 2021.

[154] T. Xu, Z. Yang, Z. Wang, and Y. Liang, "Doubly robust off-policy actor-critic: Convergence and optimality," *Preprint arXiv:2102.11866*, 2021.

[155] D. P. Bertsekas, "Approximate policy iteration: A survey and some new methods," *Journal of Control Theory and Applications*, vol. 9, no. 3, pp. 310–335, 2011.

[156] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 5, pp. 834–846, 1983.

[157] V. S. Borkar and V. R. Konda, "The actor-critic algorithm as multi-time-scale stochastic approximation," *Sadhana*, vol. 22, no. 4, pp. 525–543, 1997.

[158] T. Morimura, E. Uchibe, J. Yoshimoto, and K. Doya, "A generalized natural actor-critic algorithm," in *Advances in neural information processing systems*, 2009, pp. 1312–1320.

[159] P. S. Thomas, W. Dabney, S. Mahadevan, and S. Giguere, "Projected natural actor-critic," in *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, 2013, pp. 2337–2345.

[160] R. J. Williams and L. C. Baird, "A mathematical analysis of actor-critic architectures for learning optimal controls through incremental dynamic programming," in *Proceedings of the Sixth Yale Workshop on Adaptive and Learning Systems*, Citeseer, 1990, pp. 96–101.

[161] E. Even-Dar, S. M. Kakade, and Y. Mansour, "Online markov decision processes," *Mathematics of Operations Research*, vol. 34, no. 3, pp. 726–736, 2009.

[162] L. Shani, Y. Efroni, and S. Mannor, "Adaptive Trust Region Policy Optimization: Global Convergence and Faster Rates for Regularized MDPs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 5668–5675.

[163] G. Lan, "Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes," *Preprint arXiv:2102.00135*, 2021.

[164] K. Zhang, A. Koppel, H. Zhu, and T. Başar, "Convergence and iteration complexity of policy gradient method for infinite-horizon reinforcement learning," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, 2019, pp. 7415–7422.

[165] B. Liu, Q. Cai, Z. Yang, and Z. Wang, "Neural proximal/trust region policy optimization attains globally optimal policy," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[166] T. Xu, Z. Wang, and Y. Liang, "Non-asymptotic Convergence Analysis of Two Time-scale (Natural) Actor-Critic Algorithms," *Preprint arXiv:2005.03557*, 2020.

[167] ——, "Improving sample complexity bounds for (natural) actor-critic algorithms," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[168] Y. Liu, K. Zhang, T. Basar, and W. Yin, “An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[169] Y. Wu, W. Zhang, P. Xu, and Q. Gu, “A Finite Time Analysis of Two Time-Scale Actor Critic Methods,” *Preprint arXiv:2005.01350*, 2020.

[170] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, “Deterministic policy gradient algorithms,” in *International conference on machine learning*, PMLR, 2014, pp. 387–395.

[171] T. P. Lillicrap *et al.*, “Continuous control with deep reinforcement learning.,” in *ICLR (Poster)*, 2016.

[172] Z. Wang *et al.*, “Sample efficient actor-critic with experience replay,” *Preprint arXiv:1611.01224*, 2016.

[173] S. Fujimoto, H. Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” in *International Conference on Machine Learning*, PMLR, 2018, pp. 1587–1596.

[174] E. Imani, E. Graves, and M. White, “An off-policy policy gradient theorem using emphatic weightings,” *Preprint arXiv:1811.09013*, 2018.

[175] S. Khodadadian, Z. Chen, and S. T. Maguluri, “Finite-Sample Analysis of Off-Policy Natural Actor-Critic Algorithm,” in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 18–24 Jul 2021, pp. 5420–5431.

[176] Y. Liu, A. Swaminathan, A. Agarwal, and E. Brunskill, “Off-policy policy gradient with stationary distribution correction,” in *Uncertainty in Artificial Intelligence*, PMLR, 2020, pp. 1180–1190.

[177] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, “On the theory of policy gradient methods: Optimality, approximation, and distribution shift,” *Journal of Machine Learning Research*, vol. 22, no. 98, pp. 1–76, 2021.

[178] S. Cayci, N. He, and R. Srikant, “Linear convergence of entropy-regularized natural policy gradient with linear function approximation,” *Preprint arXiv:2106.04096*, 2021.

[179] S. Khodadadian, P. R. Jhunjhunwala, S. M. Varma, and S. T. Maguluri, “On the linear convergence of natural policy gradient algorithm,” *Preprint arXiv:2105.01424*, 2021.

[180] F. S. Melo, S. P. Meyn, and M. I. Ribeiro, "An analysis of reinforcement learning with function approximation," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 664–671.

[181] D. P. De Farias and B. Van Roy, "On the existence of fixed points for approximate value iteration and temporal-difference learning," *Journal of Optimization theory and Applications*, vol. 105, no. 3, pp. 589–608, 2000.

[182] W. Rudin *et al.*, *Principles of mathematical analysis*. McGraw-hill New York, 1964, vol. 3.

[183] D. Lee and N. He, "A unified switching system perspective and convergence analysis of $Q$-learning algorithms," in *34th Conference on Neural Information Processing Systems, NeurIPS 2020*, Conference on Neural Information Processing Systems, 2020.

[184] N. Jiang, A. Kulesza, S. Singh, and R. Lewis, "The dependence of effective planning horizon on model accuracy," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, Citeseer, 2015, pp. 1181–1189.

[185] Z. Gao, Q. Ma, T. Başar, and J. R. Birge, "Finite-Sample Analysis of Decentralized $Q$-Learning for Stochastic Games," *Preprint arXiv:2112.07859*, 2021.

[186] S. Zou, T. Xu, and Y. Liang, "Finite-sample analysis for SARSA with linear function approximation," in *Advances in Neural Information Processing Systems*, 2019, pp. 8668–8678.

[187] P. Xu and Q. Gu, "A finite-time analysis of $Q$-learning with neural network function approximation," in *International Conference on Machine Learning*, PMLR, 2020, pp. 10 555–10 565.

[188] D. Carvalho, F. S. Melo, and P. Santos, "A new convergent variant of $Q$-learning with linear function approximation," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[189] S. Zhang, H. Yao, and S. Whiteson, "Breaking the Deadly Triad with a Target Network," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 18–24 Jul 2021, pp. 12 621–12 631.

[190] H. R. Maei, C. Szepesvári, S. Bhatnagar, and R. S. Sutton, "Toward off-policy learning control with function approximation," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 719–726.

[191] Y. Wang and S. Zou, “Finite-sample Analysis of Greedy-GQ with Linear Function Approximation under Markovian Noise,” in *Conference on Uncertainty in Artificial Intelligence*, PMLR, 2020, pp. 11–20.

[192] S. Ma, Z. Chen, Y. Zhou, and S. Zou, “Greedy-GQ with Variance Reduction: Finite-time Analysis and Improved Complexity,” in *International Conference on Learning Representations*, 2021.

[193] T. Xu and Y. Liang, “Sample complexity bounds for two timescale value-based reinforcement learning algorithms,” in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2021, pp. 811–819.

[194] C. Szepesvári and R. Munos, “Finite time bounds for sampling based fitted value iteration,” in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 880–887.

[195] R. Munos and C. Szepesvári, “Finite-Time Bounds for Fitted Value Iteration,” *Journal of Machine Learning Research*, vol. 9, no. 5, 2008.

[196] T. Xie and N. Jiang, “Batch value-function approximation with only realizability,” *Preprint arXiv:2008.04990*, 2020.

[197] L. Györfi, M. Kohler, A. Krzyzak, H. Walk, *et al.*, *A distribution-free theory of nonparametric regression*. Springer, 2002, vol. 1.

[198] S. S. Du, Y. Luo, R. Wang, and H. Zhang, “Provably efficient $Q$-learning with function approximation via distribution shift error checking oracle,” in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 8060–8070.

[199] L. Yang and M. Wang, “Sample-optimal parametric $Q$-learning using linearly additive features,” in *International Conference on Machine Learning*, PMLR, 2019, pp. 6995–7004.

[200] ——, “Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound,” in *International Conference on Machine Learning*, PMLR, 2020, pp. 10 746–10 756.

[201] C. Jin, Z. Yang, Z. Wang, and M. I. Jordan, “Provably efficient reinforcement learning with linear function approximation,” in *Conference on Learning Theory*, PMLR, 2020, pp. 2137–2143.

[202] D. Zhou, J. He, and Q. Gu, “Provably efficient reinforcement learning for discounted MDPs with feature mapping,” in *International Conference on Machine Learning*, PMLR, 2021, pp. 12 793–12 802.

[203] J. He, D. Zhou, and Q. Gu, “Uniform-PAC Bounds for Reinforcement Learning with Linear Function Approximation,” *Advances in Neural Information Processing Systems, 34, 2021*, 2021.

[204] G. Li, Y. Chen, Y. Chi, Y. Gu, and Y. Wei, “Sample-Efficient Reinforcement Learning Is Feasible for Linearly Realizable MDPs with Limited Revisiting,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[205] S. S. Du, J. D. Lee, G. Mahajan, and R. Wang, “Agnostic $Q$-learning with Function Approximation in Deterministic Systems: Near-Optimal Bounds on Approximation Error and Sample Complexity,” *Advances in Neural Information Processing Systems*, vol. 2020, 2020.

[206] J. Fan, Z. Wang, Y. Xie, and Z. Yang, “A theoretical analysis of deep $Q$-learning,” in *Learning for Dynamics and Control*, PMLR, 2020, pp. 486–489.

[207] D. A. Roberts, S. Yaida, and B. Hanin, “The Principles of Deep Learning Theory,” *Preprint arXiv:2106.10165*, 2021.

# VITA

Zaiwei Chen received his B.S. degree in Chu Kochen Honors College, Zhejiang University, majoring in Electrical Engineering. Afterwards, he joined the Machine Learning Ph.D. program in the School of Industrial and Systems Engineering at Georgia Institute of Technology. On the way to his Ph.D., Zaiwei obtained two M.S. degrees, one in Operations Research, and the other in Mathematics.