Finite-Time Convergence Rates of Decentralized Markovian Stochastic Approximation

Pengfei Wang[1,2,3] , **Nenggan Zheng**[2,3,4,5*]
[1]School of Software Technology, Zhejiang University, Ningbo, China
[2]Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou, China
[3]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[4]State Key Lab of Brain-Machine Intelligence, Zhejiang University, Hangzhou, China
[5]CCAI by MOE and Zhejiang Provincial Government (ZJU), Hangzhou, China
pfei@zju.edu.cn, zng@cs.zju.edu.cn

Abstract

Markovian stochastic approximation has recently aroused a great deal of interest in many fields; however, it is not well understood in decentralized settings. Decentralized Markovian stochastic approximation is far more challenging than its single-agent counterpart due to the complex coupling structure between decentralized communication and Markovian noise-corrupted local updates. In this paper, a decentralized local markovian stochastic approximation (DLMSA) algorithm has been proposed and attains a near-optimal convergence rate. Specifically, we first provide a local variant of decentralized Markovian stochastic approximation so that each agent performs multiple local updates and then periodically communicate with its neighbors. Furthermore, we propose DLMSA with compressed communication (C-DLMSA) for further reducing the communication overhead. In this way, each agent only needs to communicate compressed information (e.g., sign compression) with its neighbors. We show that C-DLMSA enjoys the same convergence rate as that of the original DLMSA. Finally, we verify our theoretical results by applying our methods to solve multi-task reinforcement learning problems over multi-agent systems.

1 Introduction

Stochastic approximation (SA) is a class of iterative approaches for solving fixed-point equations in the presence of noise. Since its introduction in [Robbins and Monro, 1951], this type of method has received great interests due to its broad applications in many areas including stochastic optimization [Bottou *et al.*, 2018] and reinforcement learning [Sutton and Barto, 2018]. In stochastic optimization, the stochastic gradient descent (SGD) algorithm is regarded as a SA method to find an optimal solution of a target objective function. In reinforcement learning (RL), Q-learning and TD-learning are popular SA algorithms used to solve the Bellman equations [Bhandari *et al.*, 2018; Srikant and Ying, 2019]. In RL, the SA algorithms naturally involve Markovian noise. Markovian SA, which is characterized by sampling data from Markov processes, has also found applications in many other fields, such as robust optimization [Duchi *et al.*, 2012] and stochastic optimization over ergodic data [Dorfman and Levy, 2022; Alacaoglu and Lyu, 2023], where many existing algorithms can be viewed as different variants of Markovian SA.

Distributed SA has emerged as a powerful class of algorithms, when training data are collected and stored at multiple agents. This distributed training paradigm primarily arises in many real-world applications, including multi-agent reinforcement learning [Zeng *et al.*, 2021b; Sun *et al.*, 2020], decentralized decision making [Lakshmanan and De Farias, 2008], distributed and parallel computing [Kushner and Yin, 1987; Jiang and Xu, 2008], etc. Although distributed training paradigm has been actively applied in various fields [Dean *et al.*, 2012; Mnih *et al.*, 2016], Markovian SA has not been well-studied in decentralized settings. As its name suggests, "decentralized" implies a more challenging setting where all agents must rely on communications to reach a consensus without any coordination from a central server.

In this paper, we consider the Markovian SA problem on multi-agent systems:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\boldsymbol{\xi}_i\sim\mu_i}\left[\mathcal{H}_i(\mathbf{x},\boldsymbol{\xi}_i)\right]=\mathbf{x}, \tag{1}$$

where $\forall i \in \{1,2,\cdots,n\}$, $\mathcal{H}_i(\mathbf{x},\boldsymbol{\xi}_i) : \mathbb{R}^d\times\mathcal{Y}\rightarrow\mathbb{R}^d$ is a general nonlinear operator, $\boldsymbol{\xi}_i\in\mathcal{Y}$ is a random variable, and $\mathcal{H}_i(x)=\mathbb{E}_{\boldsymbol{\xi}_i\sim\mu_i}[\mathcal{H}_i(\mathbf{x},\boldsymbol{\xi}_i)]$ is the expectation operator. We are interested in the case that $\boldsymbol{\xi}_i$ is sampled from a Markov process, whose stationary distribution is μ_i. In multi-agent systems, agents can interact with each other through an undirected connected network $\mathcal{G}=(\mathcal{V},\mathcal{E})$, where $\mathcal{V}=\{1,2,\cdots,n\}$ is the collection of agents and $\mathcal{E}$ is the set of communication links (i,j), $i,j\in\mathcal{V}$ that connect agents.

Existing approaches to solving problem (1) typically use a specific "client/server" model for agent interaction where a central server is responsible for updating the global model. Specifically, [Doan, 2020] proposed a local SA algorithm and derived a convergence rate $\widetilde{\mathcal{O}}(K/T)$, where K is the commu-

[*] Corresponding author

Algorithm	Architecture	Local Updates	Markovian Noise	Compressed Communication	Rate
SA [Chen *et al.*, 2020]	Single	✗	✗	✗	$\mathcal{O}(1/T)$
Markovian SA [Chen *et al.*, 2021]	Single	✗	✔	✗	$\widetilde{\mathcal{O}}(1/T)$
Local SA [Doan, 2020]	Cen.	✔	✔	✗	$\widetilde{\mathcal{O}}(K/T)$
Federated SA [Khodadadian *et al.*, 2022]	Cen.	✔	✔	✗	$\widetilde{\mathcal{O}}(1/T)$
Markovian SA [Wai, 2020]	Dec.	✗	✔	✗	$\widetilde{\mathcal{O}}(\delta^{-1}/T^{1/2})$
DCSA [Zeng *et al.*, 2022]	Dec.	✗	✔	✗	$\widetilde{\mathcal{O}}(\delta^{-2}/T)$
DLMSA (this paper)	Dec.	✔	✔	✗	$\widetilde{\mathcal{O}}(1/T)$
C-DLMSA (this paper)	Dec.	✔	✔	✔	$\widetilde{\mathcal{O}}(1/T)$

Table 1: Convergence rates of different stochastic approximation algorithms for nonlinear stochastic approximation problems. Possible distributed architectures include 1) Single, the single-agent case, 2) Cen., a central server coordinating multiple agents; and 3) Dec., each agent directly communicating with its neighbors, without the need for a central server. δ is the spectral gap of the communication network ($0 < \delta \leq 1$). K is the communication interval ($K \geq 1$). T is the total number of iterations ($T \geq 1$).

nication interval. It is worth noting that the convergence rate depends on the communication frequency. [Khodadadian *et al.*, 2022] proposed a federated SA, and proved that the method achieves a convergence rate of $\widetilde{\mathcal{O}}(1/T)$; however, the convergence rate is derived only for $||\mathbf{x}^{\text{out}}||_c^2$ rather than $||\mathbf{x}^{out}-\mathbf{x}_*||_c^2$, where $\mathbf{x}^{out}$ is the output of the algorithm and $\mathbf{x}_*$ is the optimal solution of the SA problem. Compared to [Doan, 2020; Khodadadian *et al.*, 2022] that restrict their focus on the centralized distributed learning framework, [Wai, 2020] considered Markovian SA in decentralized settings, and showed that the convergence rate of the proposed method is $\widetilde{\mathcal{O}}(\delta^{-1}/T^{1/2})$. [Zeng *et al.*, 2022] proposed a new decentralized SA algorithm DCSA that achieves an $\widetilde{\mathcal{O}}(\delta^{-2}/T)$ convergence rate. We see that the existing convergence rates of decentralized SA depend on the spectral gap δ of the communication network. Convergence rates of existing decentralized SA algorithms are worse than the near-optimal $\mathcal{O}(1/T)$ and $\widetilde{\mathcal{O}}(1/T)$ convergence rates achieved by SA [Chen *et al.*, 2020] and Markovian SA [Chen *et al.*, 2021], respectively.

On the other hand, communication overhead is one of the main bottlenecks in distributed learning framework, which motivates the use of advanced algorithmic strategies to alleviate the communication overhead [Liu *et al.*, 2022]. In particular, [Doan, 2020; Khodadadian *et al.*, 2022] increase the number of local updates between the communication rounds to improve the computation-to-communication ratio. Another strategy is to leveraging compressed communication in which each agent sends the compressed information to its neighbors. These two strategies exhibit superior performance in scenarios where each agent has limited communication capabilities. But as far as we are aware, none of the existing work solves the Markovian SA problem by using multiple local updates or compressed communication in decentralized settings. Then, there exists a natural question:

Can we design communication-efficient decentralized Markovian stochastic approximation algorithms with near-optimal $\widetilde{\mathcal{O}}(1/T)$ convergence guarantees?

1.1 Contributions

In this paper, we give an affirmative answer to the above question by deriving convergence rates for DLMSA and C-DLMSA under mild assumptions. The main contributions are briefly described as follows.

Two New Algorithms

We first propose a decentralized local Markovian stochastic approximation algorithm (DLMSA) algorithm. In DLMSA, each agent independently performs multiple local updates corrupted by Markovian noise and periodically communicates with its neighbors over a sparse communication network. DLMSA achieves temporal-spatial communication reduction by simultaneously allowing multiple local updates (i.e., reducing communication frequency) and allowing decentralized communication via a sparse network topology. To further reduce the communication overhead, we further propose DLMSA with compressed communication (C-DLMSA), which covers both unbiased and biased compression operators (e.g., *quantization* or *sparsification*).

Convergence Guarantees

We derive finite-time convergence rates of DLMSA involving a contraction mapping with respect to an arbitrary norm rather than the Euclidean norm. We show that despite Markovian noise, multiple local updates, and network connectivity affecting the higher-order terms, the dominated term $\widetilde{\mathcal{O}}(T^{-1})$[1] in the convergence rate is the same as the centralized baseline with exact communication under i.i.d. samples, differing only by logarithmic factors. Furthermore, we demonstrate that C-DLMSA converges at a same rate to DLMSA, suggesting that C-DLMSA gains communication efficiency through compression, essentially for free. Our results and comparisons with existing work are summarized in Table **??**. The C-DLMSA algorithm significantly outperforms existing work. Our algorithms converge at a near-optimal rate while allowing multiple local updates, Markovian noise, and arbitrary communication compression in decentralized settings.

[1] The $\widetilde{\mathcal{O}}(\cdot)$ notation hides all log-terms and universal constants.

Technical Challenges in Analysis

Our analysis faces multiple challenges and requires several new insights. First, a key non-trivial technical ingredient is to identify error bounds for decentralized communication and Markovian noise-corrupted multiple local updates, and to make them compatible with proofs of the SA algorithm. For C-DLMSA, its analysis is more delicate and challenging because the compression operator may be biased. Second, our analysis is derived in the case where the norm of the contraction mapping of SA is arbitrary (e.g., ℓ_∞-norm) instead of just being the Euclidean norm $||\cdot||_2$. To this end, we introduce a potential function obtained by smoothing the norm-squared function with a generalized Moreau envelope. Third, the contraction mapping of SA or the noise is bounded by a constant in prior literature; instead, we only assume that the contraction mapping is bounded at the 0-point (see Assumption 1), and the second-order moment of the noise scales affinely with the current iterate (see Assumption 3).

Application to Multi-Task Reinforcement Learning

We apply our theoretical results to multi-task reinforcement learning over multi-agent systems. In particular, we present for the first time decentralized federated variants of the Q-learning algorithm and establish its finite-time convergence rates. Performing multiple local updates on agents before exchanging information via sparse communication network is adopted in our algorithms to improve communication efficiency. We prove that when the solution accuracy ϵ is small enough, the sample complexity and communication complexity of our algorithms are $\widetilde{\mathcal{O}}(1/\epsilon)$ and $\widetilde{\mathcal{O}}(1/\sqrt{\epsilon})$, respectively. Experiments on standard multi-task reinforcement learning tasks are provided to illustrate our theoretical results.

1.2 Related Work

Stochastic Approximation with Markovian Noise

Markovian SA relates closely to Markov gradient descent [Benveniste *et al.*, 2012; Doan, 2022] in optimization literature; however, Markovian SA is a more general framework that covers many problems in reinforcement learning that cannot be formulated as optimization problems, e.g., Q-learning [Watkins and Dayan, 1992]. Existing works on Markovian SA mainly focus on the single-agent case. The asymptotic convergence of Markovian SA was established by using the ordinary differential equation (ODE) method [Bertsekas and Tsitsiklis, 1996; Borkar, 2009; Borkar *et al.*, 2021]. The finite-time convergence rates of Markovian SA were studied in [Chen *et al.*, 2021; Chen *et al.*, 2022]. They proved that Markovian SA achieves an $\widetilde{\mathcal{O}}(1/T)$ convergence rate. [Doan, 2020; Khodadadian *et al.*, 2022] proposed distributed local Markovian SA, and derived $\widetilde{\mathcal{O}}(K/T)$ and $\widetilde{\mathcal{O}}(1/T)$ convergence rates, respectively. [Wai, 2020; Zeng *et al.*, 2022] considered distributed Markovian SA in decentralized settings, but the convergence rates of the proposed methods are dependent on the communication network topology. In sharp contrast, our methods achieve a network topology-independent convergence rate, when the total number of iterations is sufficiently large.

Federated Learning with I.I.D. Noise

Federated learning is a distributed learning paradigm that utilizes local computation of agents to train models without sacrificing data privacy; see the recent survey paper [Li *et al.*, 2020b]. In federated learning, the core algorithm FedAvg, also known as "local SGD", is featured by performing more local updates at each local agent and periodical communication via the central server [McMahan *et al.*, 2017]. The convergence analysis of FedAvg is discussed by [Khaled *et al.*, 2019; Li *et al.*, 2019]. The authors in [Sun *et al.*, 2022] studied the decentralized FedAvg, which is implemented on agents that are connected by an undirected communication network. When agents do not trust a central server to protect their privacy, decentralized federated learning is the learning paradigm of choice [Yang *et al.*, 2019]. Except for [Doan, 2020; Khodadadian *et al.*, 2022] mentioned above, we note that existing federated learning work mainly focuses on the case of sampling data from independent and identically distributed unknown distributions.

Distributed and Multi-Agent RL

Distributed Markovian SA theory has important applications in distributed and multi-agent RL. For example, the non-asymptotic analysis of distributed Q-learning can be seen as a Markovian SA problem [Xu and Gu, 2020]. Recently, there is a large literature on distributed and multi-agent RL. In particular, [Heredia *et al.*, 2020] provided a finite-time analysis of distributed Q-learning in multi-agent systems. [Li *et al.*, 2020a] derived non-asymptotic sample complexity bounds of asynchronous Q-learning. [Wang *et al.*, 2020] provided a non-asymptotic analysis of decentralized TD methods with gradient tracking and linear function approximation. [Zhang *et al.*, 2021] focused on decentralized multi-agent RL policy evaluation with nonlinear function approximation. [Sayin *et al.*, 2021] presented a provably convergent decentralized multi-agent RL learning dynamics for zero-sum discounted Markov games over an infinite horizon. [Chen *et al.*, 2018] proposed a policy gradient method termed lazily aggregated policy gradient to improve communication efficiency via infrequent communication. To our knowledge, all these works do not consider both Markovian noise-corrupted multiple local updates and compressed decentralized communication. In this paper, the proposed C-DLMSA subsumes compressed decentralized federated Q-learning as a special case.

2 Preliminaries

2.1 Single-Agent Case

The SA algorithm is an iterative procedure for finding fixed points of a function when only noisy estimates of the function are observed. Specifically, with contraction mapping $\mathcal{H}:\mathbb{R}^d\times\mathcal{Y}\to\mathbb{R}^d$ and $\boldsymbol{\xi}\in\mathcal{Y}$ being random variable with distribution μ, Markovian SA seeks to solve the equation

$$\mathbb{E}_{\boldsymbol{\xi}\sim\mu}[\mathcal{H}(\mathbf{x},\boldsymbol{\xi})]=\mathbf{x}, \tag{2}$$

by the following update

$$\begin{aligned}&\mathbf{x}_{k+1}=\mathbf{x}_k+\alpha\left(\mathcal{H}(\mathbf{x}_k,\boldsymbol{\xi}_k)-\mathbf{x}_k+\boldsymbol{\omega}_k\right),\\&\text{for } k=0,1,\cdots,T-1,\end{aligned} \tag{3}$$

where $\boldsymbol{\xi}_k$ is a random variable derived from the evolution of a Markov chain, $\boldsymbol{\omega}_k$ is additive noise, α is step size, and T denotes the number of iterations. Indeed, Q-learning can be viewed as a variant of Markovian SA [Tsitsiklis, 1994]. In [Chen *et al.*, 2020], the authors showed that with an appropriate constant step size α, Markovian SA has the following convergent behavior

$$\mathbb{E}[||\mathbf{x}_T - \mathbf{x}_*||^2] \leq C_1(1 - C_0\alpha)^T + C_2\alpha, \tag{4}$$

where C_0, C_1 and C_2 are some problem dependent positive constants and $\mathbf{x}_*$ is a solution of Eq.(2). In Eq.(4), we can reduce the finite-time convergence rate by reducing the step size α. We see that the finite-time convergence rate is dominated by the second term $C_2\alpha$, if α is small enough.

2.2 Some Mild Assumptions

Throughout this work, denote $||\cdot||_c$ as an *arbitrary* norm in $\mathbb{R}^d$. We analyze our algorithms under the following assumptions.

Assumption 1. *There exist $\rho \in (0, 1)$ and two positive constants A_1 and B_1 such that for any agent $i \in \{1, 2, \cdots, n\}$,*
(1) $||\mathcal{H}_i(\mathbf{x}_1) - \mathcal{H}_i(\mathbf{x}_2)||_c \leq \rho||\mathbf{x}_1 - \mathbf{x}_2||_c, \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$;
(2) $||\mathcal{H}_i(\mathbf{x}_1, \boldsymbol{\xi}_i) - \mathcal{H}_i(\mathbf{x}_2, \boldsymbol{\xi}_i)||_c \leq A_1||\mathbf{x}_1 - \mathbf{x}_2||_c, \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$, $\boldsymbol{\xi}_i \in \mathcal{Y}$;
(3) $||\mathcal{H}_i(\mathbf{0}, \boldsymbol{\xi}_i)||_c \leq B_1$, $\forall \boldsymbol{\xi}_i \in \mathcal{Y}$.

The first property above implies that $\mathcal{H}_i(\mathbf{x})$ is a contraction mapping with respect to an arbitrary norm $||\cdot||_c$. By applying Banach fixed-point theorem [Debnath and Mikusinski, 2005], the first property guarantees that Eq.(1) has a unique solution, which we have denoted by $\mathbf{x}_*$. The second property implies that $\mathcal{H}_i(\mathbf{x}, \boldsymbol{\xi}_i)$ is smooth with respect to the input argument $\mathbf{x}$. The last property is weaker than the general boundedness assumption considered in the existing literature; e.g., [Khodadadian *et al.*, 2022; Gao *et al.*, 2022].

Assumption 2. *The weighted connectivity matrix $\mathbf{M} = [\mathbf{M}_{ij}] \in \mathbb{R}^{n\times n}$ satisfies the following:*
(1) Doubly Stochastic: $\sum_{i=1}^n \mathbf{M}_{ij} = \sum_{j=1}^n \mathbf{M}_{ij} = 1$;
(2) Symmetric: $\mathbf{M}_{ij} = \mathbf{M}_{ji}, \forall i, j \in \{1, 2, \cdots, n\}$;
(3) Network Sparsity: $\mathbf{M}_{ij} > 0$ if $(i, j) \in \mathcal{E}$; otherwise $\mathbf{M}_{ij} = 0, \forall i, j \in \{1, 2, \cdots, n\}$.

Assumption 2 is fairly standard in the analysis of decentralized optimization; see e.g., [Xin *et al.*, 2020]. In Assumption 2, $\mathbf{M} = [\mathbf{M}_{ij}] \in \mathbb{R}^{n\times n}$ denotes the weighted connectivity matrix of $\mathcal{G}$. The ordered eigenvalues of $\mathbf{M}$ are denoted by $1 = |\lambda_1(\mathbf{M})| > |\lambda_2(\mathbf{M})| \geq \cdots \geq |\lambda_n(\mathbf{M})|$. We term $\delta = 1 - |\lambda_2(\mathbf{M})|$ as the spectral gap of $\mathbf{M}$. For such a symmetric and doubly stochastic, $\delta \in (0, 1]$ characterizes the connectivity of the network [Nedić *et al.*, 2018].

3 Decentralized Local Markovian Stochastic Approximation

In the section, we propose decentralized local Markovian stochastic approximation (DLMSA) algorithm and its compressed variant (C-DLMSA) to solve Eq.(1).

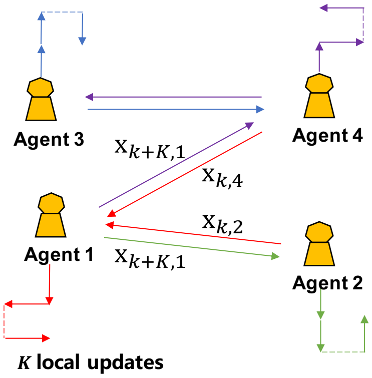

Figure 1: Schematic representation of DLMSA where agents perform K local updates and reach consensus through information exchange over a sparse communication network.

Algorithm 1 DLMSA

```
Initialization: $\alpha > 0$, $K \in \mathbb{Z}^+$, $\mathbf{x}_{0,i} = \mathbf{x}_0$ for all agent $i \in \{1, 2, \cdots, n\}$
for $k = 0, 1, \cdots, T-1$ do
    for agent $i \in \{1, 2, \cdots, n\}$ do
        $\mathbf{x}_{k+1,i} = \mathbf{x}_{k,i} + \alpha\left(\mathcal{H}_i(\mathbf{x}_{k,i}, \boldsymbol{\xi}_{k,i}) - \mathbf{x}_{k,i} + \boldsymbol{\omega}_{k,i}\right)$
        if $\mod(k+1, K) = 0$ then
            $\mathbf{x}_{k+1,i} = \sum_{j\in\mathcal{N}_i} \mathbf{M}_{ij}\mathbf{x}_{k+1,j}$
        end if
    end for
end for
```

3.1 DLMSA

The key idea of our proposed DLMSA is to perform multiple iterative updates locally on each agent, followed by decentralized communication over a sparse network topology (see Figure 1). The details of DLMSA are shown in Algorithm 1. Specifically, at each iteration k each agent i maintains a local optimization variable $\mathbf{x}_{k,i}$ and updates this variable as $\mathbf{x}_{k+1,i} = \mathbf{x}_{k,i} + \alpha\left(\mathcal{H}_i(\mathbf{x}_{k,i}, \boldsymbol{\xi}_{k,i}) - \mathbf{x}_{k,i} + \boldsymbol{\omega}_{k,i}\right)$, where $\boldsymbol{\xi}_{k,i}$ is a variable which is Markovian along the time k, $\boldsymbol{\omega}_{k,i}$ is additive noise, and α is step size.

To ensure convergence, at every K iterations, the exchange of information (through a consensus operation) occurs between connected agents (neighbors). Denote $\mathcal{N}_i = \{j \in \mathcal{V} | (i, j) \in \mathcal{E}\} \cup \{i\}$. If $\mod(k+1, K) = 0$, then $\mathbf{x}_{k+1,i} = \sum_{j\in\mathcal{N}_i} \mathbf{M}_{ij}\mathbf{x}_{k+1,j}$. In a practical implementation, each agent only sends its local optimization variable $\mathbf{x}_{k+1,i}$ to its neighbors $\mathcal{N}_i$. Here, the communication period K is greater than 1, so that the number of communication rounds is reduced to T/K.

3.2 C-DLMSA

To further reduce the communication overhead, we propose C-DLMSA (see Algorithm 2). The main pillar in C-DLMSA is an error-compensated mechanism for mitigating compression errors. Specifically, at each iteration k each agent i maintains local variables $(\mathbf{x}_{k,i}, \{\widehat{\mathbf{x}}_{k,j}\}_{j\in\mathcal{N}_i}, \mathbf{y}_{k,i})$, and updates the variable $\mathbf{y}$ as $\mathbf{y}_{k+1,i} = \mathbf{x}_{k,i} + \alpha\left(\mathcal{H}_i(\mathbf{x}_{k,i}, \boldsymbol{\xi}_{k,i}) - \mathbf{x}_{k,i} + \boldsymbol{\omega}_{k,i}\right)$.

Algorithm 2 C-DLMSA

```
Initialization: $\alpha > 0$, $\eta > 0$, $K \in \mathbb{Z}^+$, $\mathbf{x}_{0,i} = \mathbf{x}_0$, $\widehat{\mathbf{x}}_{0,i} = \mathbf{0}$ for all agent $i \in \{1, 2, \cdots, n\}$
for $k = 0, 1, \cdots, T-1$ do
    for agent $i \in \{1, 2, \cdots, n\}$ do
        $\mathbf{y}_{k+1,i} = \mathbf{x}_{k,i} + \alpha \left(\mathcal{H}_i(\mathbf{x}_{k,i}, \boldsymbol{\xi}_{k,i}) - \mathbf{x}_{k,i} + \boldsymbol{\omega}_{k,i}\right)$
        if $\mod(k+1, K) = 0$ then
            $\mathbf{q}_{k,i} = \mathcal{Q}[\mathbf{y}_{k+1,i} - \widehat{\mathbf{x}}_{k,i}]$
            $\widehat{\mathbf{x}}_{k+1,i} = \widehat{\mathbf{x}}_{k,i} + \mathbf{q}_{k,i}$
            $\mathbf{x}_{k+1,i} = \mathbf{y}_{k+1,i} + \eta \sum_{j \in \mathcal{N}_i} \mathbf{M}_{ij}(\widehat{\mathbf{x}}_{k+1,j} - \widehat{\mathbf{x}}_{k+1,i})$
        else
            $\widehat{\mathbf{x}}_{k+1,i} = \widehat{\mathbf{x}}_{k,i}$
            $\mathbf{x}_{k+1,i} = \mathbf{y}_{k+1,i}$
        end if
    end for
end for
```

If $\mod(k+1, K) = 0$, each agent i sends $\mathbf{q}_{k,i} = \mathcal{Q}[\mathbf{y}_{k+1,i} - \widehat{\mathbf{x}}_{k,i}]$ to its neighbors, where $\mathcal{Q}$ is the compressed operator. Upon receiving $\mathbf{q}_{k,j}$ from its neighbors, the agent i updates $\widehat{\mathbf{x}}_{k+1,j} = \widehat{\mathbf{x}}_{k,j} + \mathbf{q}_{k,j}$ and $\mathbf{x}_{k+1,i} = \mathbf{y}_{k+1,i} + \eta \sum_{j \in \mathcal{N}_i} \mathbf{M}_{ij}(\widehat{\mathbf{x}}_{k+1,j} - \widehat{\mathbf{x}}_{k+1,i})$.

If $\mod(k+1, K) \neq 0$, then $\widehat{\mathbf{x}}_{k+1,i} = \widehat{\mathbf{x}}_{k,i}$, $\mathbf{x}_{k+1,i} = \mathbf{y}_{k+1,i}$, and no information exchange occurs.

4 Main Results

Define $\mathcal{F}_k$ as the Sigma-algebra generated by $\{(\mathbf{x}_{k',i}, \boldsymbol{\xi}_{k',i}, \boldsymbol{\omega}_{k',i})\}_{0 \leq k' \leq k-1,\ i \in \{1,2,\cdots,n\}} \cup \{\mathbf{x}_{k,i}\}$. We make the following assumption on the noises $\{\boldsymbol{\omega}_{k,i}\}$.

Assumption 3. *For any agent $i \in \{1, 2, \cdots, n\}$, the random process $\{\boldsymbol{\omega}_{k,i}\}$ satisfies the following:*
(1) $\mathbb{E}\left[\boldsymbol{\omega}_{k,i} | \mathcal{F}_k\right] = 0$ for all $k \geq 0$;
(2) $||\boldsymbol{\omega}_{k,i}||_c \leq A_2 ||\mathbf{x}_{k,i}||_c + B_2$ for all $k \geq 0$, where A_2 and B_2 are positive constants.

Here, we do not assume that $\boldsymbol{\omega}_{k,i}$ are independent among agents $i \in \{1, 2, \cdots, n\}$, which is often assumed when studying the finite-time convergence rates of distributed SA; see for example [Khodadadian *et al.*, 2022].

In this work, we consider the case where the random variable $\{\boldsymbol{\xi}_{k,i}\}$ for each agent i is generated by a Markov process. This Markovian sampling results in correlated and biased data [Kumar *et al.*, 2023]. To this end, we impose the following assumption on the Markov chain $\{\boldsymbol{\xi}_{k,i}\}$.

Assumption 4. *For any agent $i \in \{1, 2, \cdots, n\}$, the Markov chain $\{\boldsymbol{\xi}_{k,i}\}$ has a unique stationary distribution $\mu_i \in \Delta^{|\mathcal{Y}|}$, and there exist constants $\zeta > 0$ and $\sigma \in (0, 1)$ such that $\max_{\boldsymbol{\xi}_i \in \mathcal{Y}} ||P^k(\boldsymbol{\xi}_i, \cdot) - \mu_i(\cdot)||_{\mathrm{TV}} \leq \zeta \sigma^k$ for all $k \geq 0$, where $\Delta^{|\mathcal{Y}|}$ is the probability simplex on $\mathbb{R}^{|\mathcal{Y}|}$, and $|| \cdot ||_{\mathrm{TV}}$ stands for the total variation distance [Levin and Peres, 2017].*

We denote by $\tau_\alpha = \min\{k \geq 0 : \max_{\boldsymbol{\xi}_i \in \mathcal{Y}} ||P^k(\boldsymbol{\xi}_i, \cdot) - \mu_i(\cdot)||_{\mathrm{TV}} \leq \alpha, \forall i \in \{1, 2, \cdots, n\}\}$. The mixing time τ_α represents the time it takes for the distribution $\{\boldsymbol{\xi}_{k,i}\}$ to get close to the stationary distribution μ_i. Assumption 4 implies that $\tau_\alpha \leq \left(\log(1/\alpha) + \log(\zeta/\sigma)\right) / \log(1/\sigma)$, and it follows that $\lim_{\alpha \to 0} \alpha \tau_\alpha^2 = 0$. When the Markov chain $\{\boldsymbol{\xi}_{k,i}\}$ is irreducible and aperiodic, Assumption 4 holds [Levin and Peres, 2017].

4.1 Finite-Time Convergence Rate of DLMSA

We provide a finite-time convergence rate for Algorithm 1 in the following theorem.

Theorem 1. *Consider the updates of Algorithm 1. Suppose that Assumptions 1, 2, 3, and 4 are satisfied, and $\alpha \leq \min\left\{\frac{1}{4A\tau_\alpha}, \frac{\delta}{4}\frac{1}{\sqrt{A}K}, \frac{\phi}{8}\frac{1}{\digamma A^2 \kappa \nu \tau_\alpha^2}, \frac{1}{32}\sqrt{\frac{\phi\delta^2}{\digamma A^2 \kappa K^2 (\tau_\alpha + 1)\nu}}\right\}$ for some positive constants $\digamma$, A, κ, ϕ, and ν, which are specified precisely in Appendix A, and are independent of α, δ, τ_α, n, T, and K. For any $T > 2\tau_\alpha$, we have*

$$\begin{aligned} &||\mathbf{x}^{out} - \mathbf{x}_*||_c^2 \\ &\leq \mathcal{O}\left(\frac{1}{\alpha}\left(1 - \frac{1}{4}\phi\alpha\right)^{T - 2\tau_\alpha + 1} + \alpha\tau_\alpha^2 + \frac{\alpha^2 K^2}{\delta^2}\right), \end{aligned} \quad (5)$$

where $\mathbf{x}^{out} = \frac{1}{W_T}\sum_{k=2\tau_\alpha}^{T} \frac{w_k}{n} \sum_{i=1}^{n} \mathbf{x}_{k,i}$ for weights $w_k = \left(1 - \frac{1}{4}\phi\alpha\right)^{-k}$, $W_T = \sum_{k=2\tau_\alpha}^{T} w_k$, and $0 < \phi < 1$.

Remark 1 (Finite-time convergence analysis): Theorem 1 provides the first finite-time analysis of decentralized federated stochastic approximation under Markovian sampling. Specifically, with respect to an arbitrary norm $|| \cdot ||_c$, Theorem 1 establishes the convergence of $\mathbf{x}^{out}$ to a ball around the optimal solution $\mathbf{x}_*$ with a radius on the order of $\alpha \log^2(\frac{1}{\alpha})$. The first term in Eq.(5) converges geometrically to zero as T grows, and the convergence rate in Eq.(5) is dominated by the second term, being proportional to $\alpha \log^2(\frac{1}{\alpha})$ for small enough α. In Theorem 1, the network topology and multiple local updates mildly affect the convergence rate, with δ and K only affecting the highest-order term of α in Eq.(5). Selecting the step size $\alpha = \mathcal{O}(\log T / T)$ for sufficiently large T, we see that $\mathbf{x}^{out}$ converges exactly to the optimal solution with convergence rate $\widetilde{\mathcal{O}}(1/T)$. When $\mathcal{H}_i(\mathbf{x}, \boldsymbol{\xi}_i)$ is the gradient of some function, we recover the convergence rate of the decentralized federated averaging algorithm in [Sun *et al.*, 2022] for solving strongly convex objective functions under i.i.d. samples, up to logarithmic factors.

Remark 2 (Convergence dependence on δ and K): The highest-order term of α in Eq.(5) depends quadratically on the inverse of the spectral gap δ, which shows the impact of the communication network on the convergence of the algorithm. We see that a smaller δ implies that the communication network is sparse and the convergence rate slows down. The highest-order term of α in Eq.(5) depends quadratically on the communication interval K. We see that a larger K indicates less communication between agents, so the highest-order term of α in Eq.(5) becomes larger.

Remark 3 (Comparison with the rate of decentralized stochastic approximation): In Theorem 1, the network topology δ only affects the highest-order term of α in Eq.(5). However, the dominated order of bounds in [Zeng *et al.*, 2022] is affected by the network topology. Furthermore, the order of convergence rate in Theorem 1 is independent of the number of agents, i.e., n, which is in contrast to the convergence rate in [Zeng *et al.*, 2022] that gets worse as n increases.

Remark 4 (Comparison with the rate of federated stochastic approximation): In Theorem 1, we prove that the output $\mathbf{x}^{out}$ of DLMSA using a decreasing step size converges to the optimal solution $\mathbf{x}_*$ of the SA problem; however, [Khodadadian *et al.*, 2022] only proves that $||\mathbf{x}^{out}||_c^2$ converges to 0. Furthermore, the consensus errors induced by the decentralized network structure in this paper make the analysis more challenging compared to the centralized SA algorithm proposed in [Khodadadian *et al.*, 2022].

4.2 Finite-Time Convergence Rate of C-DLMSA

To analyze the convergence rate of C-DLMSA, we impose the following assumption on the compressed operator $\mathcal{Q}$.

Assumption 5. *The compressed operator* $\mathcal{Q} : \mathbb{R}^d \to \mathbb{R}^d$ *satisfies* $\mathbb{E}_Q[||\mathcal{Q}[\mathbf{x}] - \mathbf{x}||_c^2] \leq (1-\omega)||\mathbf{x}||_c^2$ *for a parameter* $0 < \omega \leq 1$ *and* $\forall \mathbf{x} \in \mathbb{R}^d$*. Here* $\mathbb{E}_Q[\cdot]$ *denotes the expectation on the internal randomness of the compressed operator* $\mathcal{Q}$*.*

We provide a finite-time convergence rate for Algorithm 2 in the following theorem.

Theorem 2. *Consider the updates of Algorithm 2. Suppose that Assumptions 1, 2, 3, 4, and 5 are satisfied,* $\eta = \frac{\delta\omega^3}{64\beta^2+12\beta^2\omega^2+2\delta^2\omega^2}$ *with* $\beta = \max_i\{1 - \lambda_i(\mathbf{M})\}$*, and* $\alpha \leq \min\left\{\frac{1}{4A\tau_\alpha}, \frac{\delta}{4}\frac{1}{\sqrt{AK}}, \frac{\phi}{8}\frac{1}{\digamma A^2\kappa\nu\tau_\alpha^2}, \frac{1}{8}\sqrt{\frac{\eta\delta}{AK^2\mathcal{C}(\eta,\delta,\beta,\omega)}}, \frac{1}{64}\sqrt{\frac{\phi\eta\delta}{\digamma A^2\kappa K^2(\tau_\alpha+1)\nu\mathcal{C}(\eta,\delta,\beta,\omega)}}\right\}$ *for some positive constants* $\digamma$*,* A*,* κ*,* ϕ*, and* ν*, which are specified precisely in Appendix B, and are independent of* α*,* δ*,* τ_α*,* n*,* T*, and* K*, where* $\mathcal{C}(\eta,\delta,\beta,\omega)$ *is a constant defined in Appendix B, being dependent on* η*,* δ*,* β*, and* ω*. For any* $T > 2\tau_\alpha$*, we have*

$$||\mathbf{x}^{out} - \mathbf{x}_*||_c^2 \leq \mathcal{O}\left(\frac{1}{\alpha}\left(1 - \frac{1}{4}\phi\alpha\right)^{T-2\tau_\alpha+1} + \alpha\tau_\alpha^2 + \frac{\alpha^2K^2}{\delta^4\omega^6}\right), \quad (6)$$

where $\mathbf{x}^{out} = \frac{1}{W_T}\sum_{k=2\tau_\alpha}^{T}\frac{w_k}{n}\sum_{i=1}^{n}\mathbf{x}_{k,i}$ *for weights* $w_k = \left(1 - \frac{1}{4}\phi\alpha\right)^{-k}$, $W_T = \sum_{k=2\tau_\alpha}^{T} w_k$*, and* $0 < \phi < 1$*.*

Remark 5 (Finite-time convergence analysis): Theorem 2 provides a finite-time analysis of C-DLMSA under Markovian sampling. Similarly, the first term in Eq.(6) converges geometrically to zero as T grows, and the convergence rate in Eq.(6) is dominated by the second term, being proportional to $\alpha\log^2(\frac{1}{\alpha})$ for small enough α. The network topology, multiple local updates, and compressed communication mildly affect the convergence rate, with δ, K, and ω only affecting the highest-order term of α in Eq.(6).

Remark 6 (Convergence dependence on ω): The highest-order term of α in Eq.(6) depends on the inverse of the compression parameter ω, which shows the impact of compressed communication on the convergence of the algorithm. We see that a smaller ω close to 0 means that the compressing communication loses more information, so the highest-order term of α in Eq.(6) becomes larger.

5 Application to Multi-Task Reinforcement Learning

In decentralized federated Q-learning, agents collaboratively estimate the optimal Q-function. Specifically, for each $k \geq 0$, each agent i maintains a local Q-function estimate $Q_{k,i}(s,a)$. Here, we also consider an MDP consisting of a finite set of states $\mathcal{S}$ and a finite set of actions $\mathcal{A}$. Given trajectories $\{(S_{k,i}, A_{k,i}, R_{k,i}, S_{k+1,i})\}$ collected using a suitable behavior policy π_b, the iterate $Q_{k,i}(s,a)$ is updated as

$$\begin{aligned} Q_{k+1,i}(S_{k,i}, A_{k,i}) &= Q_{k,i}(S_{k,i}, A_{k,i}) \\ &\quad + \alpha\Gamma[Q_{k,i}, S_{k,i}, A_{k,i}, R_{k,i}, S_{k+1,i}]; \\ Q_{k+1,i}(s,a) &= Q_{k,i}(s,a), \quad \text{otherwise}, \end{aligned} \quad (7)$$

where we denote by $\Gamma[Q_{k,i}, S_{k,i}, A_{k,i}, R_{k,i}, S_{k+1,i}] = R_{k,i} + \gamma\max_{a'} Q_{k,i}(S_{k+1,i}, a') - Q_{k,i}(S_{k,i}, A_{k,i})$. If $\mod (k+1, K) = 0$, then $Q_{k+1,i} = \sum_{j\in\mathcal{N}_i}\mathbf{M}_{i,j}Q_{k+1,j}$. Similarly, we can use the communication compression in Algorithm 2 to get compressed decentralized federated Q-learning. Specifically, the update (7) in decentralized federated Q-learning can be rewritten as

$$Q_{k+1,i} = Q_{k,i} + \alpha(\mathcal{H}_i(Q_{k,i}, \boldsymbol{\xi}_{k,i}) - Q_{k,i} + \boldsymbol{\omega}_{k,i}), \quad (8)$$

which is in the same form of C-DLMSA by defining $\boldsymbol{\xi}_{k,i} = (S_{k,i}, A_{k,i}, R_{k,i}, S_{k+1,i})$, $\boldsymbol{\omega}_{k,i} = 0$, and the nonlinear operator $\mathcal{H}_i(Q_{k,i}, \boldsymbol{\xi}_{k,i}) : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \times \mathcal{Y} \to \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$:

$$\begin{aligned} &\mathcal{H}_i(Q_{k,i}, \boldsymbol{\xi}_{k,i})(s,a) = \mathbb{1}_{\{(s,a)=(S_{k,i},A_{k,i})\}} \\ &\quad \times \Gamma[Q_{k,i}, s, a, R_{k,i}, S_{k+1,i}] + Q_{k,i}(s,a), \ \forall(s,a). \end{aligned} \quad (9)$$

The properties of the operator $\mathcal{H}_i(Q_{k,i}, \boldsymbol{\xi}_{k,i})$ and the Markov chain $\{\boldsymbol{\xi}_{k,i}\}$ are established in the following proposition, which guarantee that Assumptions 1, 3, and 4 are satisfied.

Proposition 1. *Assume the behavior policy* π_b *satisfies* $\pi_b(a|s) > 0$ *for all* (s,a)*, and the Markov chains* $\{S_{k,i}\}$ *induced by* π_b *are irreducible and aperiodic, then Assumptions 1, 3, and 4 are satisfied with respect to the* $||\cdot||_\infty$ *norm.*

Corollary 1. *Consider* Q^{out} *generated by (compressed) decentralized federated Q-learning. Assume* π_b *satisfies the conditions in Proposition 1. In order to make* $||Q^{out} - Q_*||_\infty^2 \leq \epsilon$*, where* $\epsilon > 0$ *is a given accuracy and* Q_* *is the Q-function associated with an optimal policy, the sample and communication complexities are* $\widetilde{\mathcal{O}}(1/\epsilon)$ *and* $\widetilde{\mathcal{O}}(1/\sqrt{\epsilon})$*, respectively.*

5.1 Experiments

In this section, we aim to support our theoretical results by applying (compressed) decentralized federated Q-learning to solve the multi-task GridWorld problem [Zeng *et al.*, 2021a].

We consider the GridWorld problem with n individual environments of size 10×10 over n different agents. We assign each agent to an environment where the agent is placed in a grid of cells, each cell having one of three labels $\{goal, obstacle, empty\}$. The agent chooses one action of 4 actions $\{up, down, left, right\}$ to move to the next cell. The

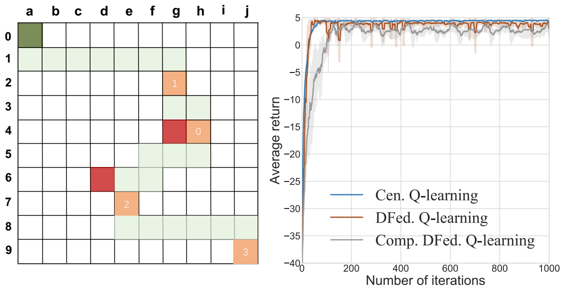

Figure 2: 4Agents@2Obstacles: Left: evaluation of learned policy. Right: average return for different algorithms. The Dark green cell represents the starting position of the agents. Red cells represent obstacles. Yellow cells represent goals and white numbers represent the indices of agents.

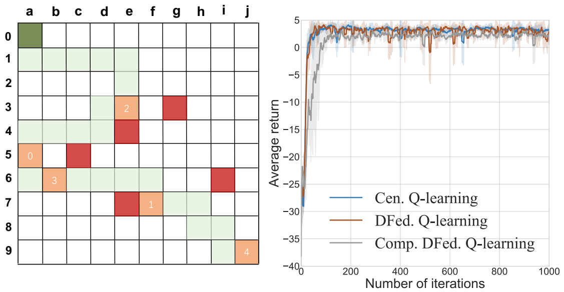

Figure 3: 5Agents@5Obstacles: Left: evaluation of learned policy. Right: average return for different algorithms.

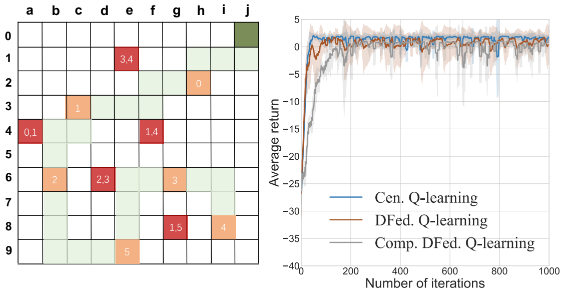

Figure 4: 6Agents@5Obstacles: Left: evaluation of learned policy. Right: average return for different algorithms. The number i in red cell represents the obstacle of the i-th agent.

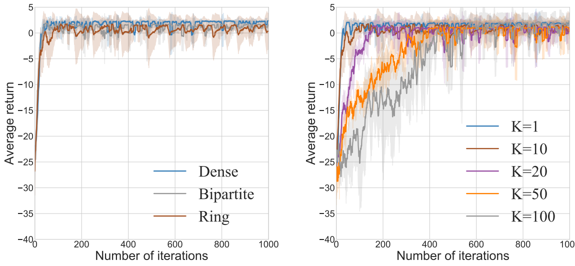

Figure 5: Left: different choices of the network topology. Right: different choices of the communication interval.

reward is $-0.1\times$ (distance between agent and goal) ± 10, depending on whether the goal is reached or the agent is trapped in an obstacle. By training agents in different environments to obtain a unified policy, it is expected to have better generalization ability. We connect the agents in a ring graph and use centralized Q-learning (Cen. Q-learning), decentralized federated Q-learning (DFed. Q-learning), and compressed decentralized federated Q-learning (Comp. DFed. Q-learning) to train them for 1000 episodes. In centralized Q-learning, consensus operations with exact information are performed at each iteration. We use the same best-tuned learning rate $\alpha = 0.5$, communication interval $K = 10$, and discounter factor $\gamma = 0.99$ in all experiments. For Comp. DFed. Q-learning, we use $top_{1\%}$ [Stich, 2018] as the compression operator. Experimental results are the average over 10 random seeds.

Figure 2 considers experiments with 4 agents, while Figures 3 and 4 consider experiments with 5 and 6 agents, respectively. After 1000 episodes of training with DFed. Q-learning, we observe that agents agree on a policy whose performance is tested across all environments. We combine all the results into one grid, as shown on the left side of Figures 2-4. The light green path is the route that the agents visit in these environments. We see that a unified policy finds all targets in all environments. This verifies the effectiveness of DFed. Q-learning. On the right side of Figures 2-4, we compare Fed. Q-learning and Comp. DFed. Q-learning with Cen. Q-learning. We show the results for the average return versus the number of iterations. We see that despite consensus errors, local update errors, and compression errors, the Comp. DFed. Q-learning achieves comparable performance to Cen. Q-learning. Figure 5 shows different choices of network topology δ and communication interval K. The results show that the effect of network topology on the convergence rate of DFed. Q-learning is limited, and we observe similar results for DFed. Q-learning when the communication interval K increases from 1, 10 to 20.

6 Conclusion

In this paper, we proposed decentralized local Markovian stochastic approximation DLMSA algorithm and its compressed variant C-DLMSA. The C-DLMSA algorithm can achieve temporal-spatial communication reduction by allowing multiple local updates, decentralized communication through sparse network topology, and arbitrary communication compression. We established finite-time convergence rates for DLMSA and C-DLMSA, and showed that the algorithms converge at a near-optimal rate $\widetilde{\mathcal{O}}(1/T)$. Finally, we applied our algorithms to multi-task reinforcement learning. Future directions of this work include studying DLMSA and C-DLMSA under asynchronous communication as well as time-varying and/or directed communication graphs.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (T2293723), Zhejiang Provincial Natural Science Foundation (Z24F020008), the Key R&D Program of Zhejiang Province (2022C01022, 2022C01119, 2021C03003), and the Fundamental Research Funds for the Central Universities (No. 226-2022-00051).

References

[Alacaoglu and Lyu, 2023] Ahmet Alacaoglu and Hanbaek Lyu. Convergence of first-order methods for constrained nonconvex optimization with dependent data. In *International Conference on Machine Learning*, volume 202, pages 458–489. PMLR, 2023.

[Benveniste *et al.*, 2012] Albert Benveniste, Michel Métivier, and Pierre Priouret. *Adaptive algorithms and stochastic approximations*, volume 22. Springer Science & Business Media, 2012.

[Bertsekas and Tsitsiklis, 1996] Dimitri Bertsekas and John N Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

[Bhandari *et al.*, 2018] Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. In *Conference on learning theory*, pages 1691–1692. PMLR, 2018.

[Borkar *et al.*, 2021] Vivek Borkar, Shuhang Chen, Adithya Devraj, Ioannis Kontoyiannis, and Sean Meyn. The ode method for asymptotic statistics in stochastic approximation and reinforcement learning. *arXiv preprint arXiv:2110.14427*, 2021.

[Borkar, 2009] Vivek S Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

[Bottou *et al.*, 2018] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018.

[Chen *et al.*, 2018] Tianyi Chen, Kaiqing Zhang, Georgios B Giannakis, and Tamer Basar. Communication-efficient distributed reinforcement learning. *arXiv preprint arXiv:1812.03239*, 16, 2018.

[Chen *et al.*, 2020] Zaiwei Chen, Siva Theja Maguluri, Sanjay Shakkottai, and Karthikeyan Shanmugam. Finite-sample analysis of contractive stochastic approximation using smooth convex envelopes. *Advances in Neural Information Processing Systems*, 33:8223–8234, 2020.

[Chen *et al.*, 2021] Zaiwei Chen, Siva Theja Maguluri, Sanjay Shakkottai, and Karthikeyan Shanmugam. A lyapunov theory for finite-sample guarantees of asynchronous q-learning and td-learning variants. *arXiv preprint arXiv:2102.01567*, 2021.

[Chen *et al.*, 2022] Zaiwei Chen, Sheng Zhang, Thinh T Doan, John-Paul Clarke, and Siva Theja Maguluri. Finite-sample analysis of nonlinear stochastic approximation with applications in reinforcement learning. *Automatica*, 146:110623, 2022.

[Dean *et al.*, 2012] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, et al. Large scale distributed deep networks. *Advances in neural information processing systems*, 25, 2012.

[Debnath and Mikusinski, 2005] Lokenath Debnath and Piotr Mikusinski. *Introduction to Hilbert spaces with applications*. Academic press, 2005.

[Doan, 2020] Thinh T Doan. Local stochastic approximation: A unified view of federated learning and distributed multi-task reinforcement learning algorithms. *arXiv preprint arXiv:2006.13460*, 2020.

[Doan, 2022] Thinh T Doan. Finite-time analysis of markov gradient descent. *IEEE Transactions on Automatic Control*, 68(4):2140–2153, 2022.

[Dorfman and Levy, 2022] Ron Dorfman and Kfir Yehuda Levy. Adapting to mixing time in stochastic optimization with markovian data. In *International Conference on Machine Learning*, pages 5429–5446. PMLR, 2022.

[Duchi *et al.*, 2012] John C Duchi, Alekh Agarwal, Mikael Johansson, and Michael I Jordan. Ergodic mirror descent. *SIAM Journal on Optimization*, 22(4):1549–1578, 2012.

[Gao *et al.*, 2022] Hongchang Gao, Junyi Li, and Heng Huang. On the convergence of local stochastic compositional gradient descent with momentum. In *International Conference on Machine Learning*, pages 7017–7035. PMLR, 2022.

[Heredia *et al.*, 2020] Paulo Heredia, Hasan Ghadialy, and Shaoshuai Mou. Finite-sample analysis of distributed q-learning for multi-agent networks. In *2020 American Control Conference*, pages 3511–3516. IEEE, 2020.

[Jiang and Xu, 2008] Houyuan Jiang and Huifu Xu. Stochastic approximation approaches to the stochastic variational inequality problem. *IEEE Transactions on Automatic Control*, 53(6):1462–1475, 2008.

[Khaled *et al.*, 2019] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. First analysis of local gd on heterogeneous data. *arXiv preprint arXiv:1909.04715*, 2019.

[Khodadadian *et al.*, 2022] Sajad Khodadadian, Pranay Sharma, Gauri Joshi, and Siva Theja Maguluri. Federated reinforcement learning: Linear speedup under markovian sampling. In *International Conference on Machine Learning*, pages 10997–11057. PMLR, 2022.

[Kumar *et al.*, 2023] Harshat Kumar, Alec Koppel, and Alejandro Ribeiro. On the sample complexity of actor-critic method for reinforcement learning with function approximation. *Machine Learning*, pages 1–35, 2023.

[Kushner and Yin, 1987] Harold Joseph Kushner and George Yin. Stochastic approximation algorithms for parallel and distributed processing. *Stochastics: An International Journal of Probability and Stochastic Processes*, 22(3-4):219–250, 1987.

[Lakshmanan and De Farias, 2008] Hariharan Lakshmanan and Daniela Pucci De Farias. Decentralized resource allocation in dynamic networks of agents. *SIAM Journal on Optimization*, 19(2):911–940, 2008.

[Levin and Peres, 2017] David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

[Li *et al.*, 2019] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2019.

[Li *et al.*, 2020a] Gen Li, Yuting Wei, Yuejie Chi, Yuantao Gu, and Yuxin Chen. Sample complexity of asynchronous q-learning: Sharper analysis and variance reduction. *Advances in neural information processing systems*, 33:7031–7043, 2020.

[Li *et al.*, 2020b] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

[Liu *et al.*, 2022] Wei Liu, Li Chen, and Wenyi Zhang. Decentralized federated learning: Balancing communication and computing costs. *IEEE Transactions on Signal and Information Processing over Networks*, 8:131–143, 2022.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[Mnih *et al.*, 2016] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[Nedić *et al.*, 2018] Angelia Nedić, Alex Olshevsky, and Michael G Rabbat. Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106(5):953–976, 2018.

[Robbins and Monro, 1951] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[Sayin *et al.*, 2021] Muhammed Sayin, Kaiqing Zhang, David Leslie, Tamer Basar, and Asuman Ozdaglar. Decentralized q-learning in zero-sum markov games. *Advances in Neural Information Processing Systems*, 34:18320–18334, 2021.

[Srikant and Ying, 2019] Rayadurgam Srikant and Lei Ying. Finite-time error bounds for linear stochastic approximation and td learning. In *Conference on Learning Theory*, pages 2803–2830. PMLR, 2019.

[Stich, 2018] Sebastian U Stich. Local sgd converges fast and communicates little. In *International Conference on Learning Representations*, 2018.

[Sun *et al.*, 2020] Jun Sun, Gang Wang, Georgios B Giannakis, Qinmin Yang, and Zaiyue Yang. Finite-time analysis of decentralized temporal-difference learning with linear function approximation. In *International Conference on Artificial Intelligence and Statistics*, pages 4485–4495. PMLR, 2020.

[Sun *et al.*, 2022] Tao Sun, Dongsheng Li, and Bao Wang. Decentralized federated averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Tsitsiklis, 1994] John N Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine learning*, 16:185–202, 1994.

[Wai, 2020] Hoi-To Wai. On the convergence of consensus algorithms with markovian noise and gradient bias. In *2020 59th IEEE Conference on Decision and Control*, pages 4897–4902. IEEE, 2020.

[Wang *et al.*, 2020] Gang Wang, Songtao Lu, Georgios Giannakis, Gerald Tesauro, and Jian Sun. Decentralized td tracking with linear function approximation and its finite-time analysis. *Advances in Neural Information Processing Systems*, 33:13762–13772, 2020.

[Watkins and Dayan, 1992] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.

[Xin *et al.*, 2020] Ran Xin, Soummya Kar, and Usman A Khan. Decentralized stochastic optimization and machine learning: A unified variance-reduction framework for robust performance and fast convergence. *IEEE Signal Processing Magazine*, 37(3):102–113, 2020.

[Xu and Gu, 2020] Pan Xu and Quanquan Gu. A finite-time analysis of q-learning with neural network function approximation. In *International Conference on Machine Learning*, pages 10555–10565. PMLR, 2020.

[Yang *et al.*, 2019] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[Zeng *et al.*, 2021a] Sihan Zeng, Malik Aqeel Anwar, Thinh T Doan, Arijit Raychowdhury, and Justin Romberg. A decentralized policy gradient approach to multi-task reinforcement learning. In *Uncertainty in Artificial Intelligence*, pages 1002–1012. PMLR, 2021.

[Zeng *et al.*, 2021b] Sihan Zeng, Thinh T Doan, and Justin Romberg. Finite-time analysis of decentralized stochastic approximation with applications in multi-agent and multi-task learning. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 2641–2646. IEEE, 2021.

[Zeng *et al.*, 2022] Sihan Zeng, Thinh T Doan, and Justin Romberg. Finite-time convergence rates of decentralized stochastic approximation with applications in multi-agent and multi-task learning. *IEEE Transactions on Automatic Control*, 2022.

[Zhang *et al.*, 2021] Xin Zhang, Zhuqing Liu, Jia Liu, Zhengyuan Zhu, and Songtao Lu. Taming communication and sample complexities in decentralized policy evaluation for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:18825–18838, 2021.