UCLA
UCLA Electronic Theses and Dissertations

Title
Exact Diffusion Learning over Networks

Permalink
https://escholarship.org/uc/item/7z3677dm

Author
Yuan, Kun

Publication Date
2019

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA
Los Angeles

Exact Diffusion Learning over Networks

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical and Computer Engineering

by

Kun Yuan

2019

© Copyright by
Kun Yuan
2019

ABSTRACT OF THE DISSERTATION

Exact Diffusion Learning over Networks

by

Kun Yuan
Doctor of Philosophy in Electrical and Computer Engineering
University of California, Los Angeles, 2019
Professor Ali H. Sayed, Chair

In this dissertation, we study optimization, adaptation, and learning problems over connected networks. In these problems, each agent k collects and learns from its own local data and is able to communicate with its local neighbors. While each single node in the network may not be capable of sophisticated behavior on its own, the agents collaborate to solve large-scale and challenging learning problems.

Different approaches have been proposed in the literature to boost the learning capabilities of networked agents. Among these approaches, the class of diffusion strategies has been shown to be particularly well-suited due to their enhanced stability range over other methods and improved performance in adaptive scenarios. However, diffusion implementations suffer from a small inherent bias in the iterates. When a constant step-size is employed to solve deterministic optimization problems, the iterates generated by the diffusion strategy will converge to a small neighborhood around the desired global solution but not to the exact solution itself. This bias is not due to any gradient noise arising from stochastic approximation; it is instead due to the update structure in diffusion implementations. The existence of the bias leads to three questions: (1) What is the origin of this inherent bias? (2) Can it be eliminated? (3) Does the correction of the bias bring benefits to distributed optimization, distributed adaptation, or distributed learning?

This dissertation provides affirmative solutions to these questions. Specifically, we design a new *exact diffusion* approach that eliminates the inherent bias in diffusion. Exact diffusion

has almost the same structure as diffusion, with the addition of a "correction" step between the adaptation and combination steps. Next, this dissertation studies the performance of exact diffusion for the scenarios of distributed optimization, distributed adaptation, and distributed learning, respectively. For distributed optimization, exact diffusion is proven to converge exponentially fast to the *exact* global solution under proper conditions. For distributed adaptation, exact diffusion is proven to have better steady-state mean-square-error than diffusion, and this superiority is analytically shown to be more evident for sparsely-connected networks such as line, cycle, grid, and other topologies. In distributed learning, exact diffusion can be integrated with the amortized variance-reduced gradient method (AVRG) so that it converges exponentially fast to the exact global solution while employing stochastic gradients per iteration. This dissertation also compares exact diffusion with other state-of-the-art methods in literature. Intensive numerical simulations are provided to illustrate the theoretical results derived in the dissertation.

The dissertation of Kun Yuan is approved.

Wotao Yin

Lieven Vandenberghe

Lara Dolecek

Ali H. Sayed, Committee Chair

University of California, Los Angeles

2019

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Professor Ali H. Sayed. Working at the Adaptive Systems Laboratory at UCLA has been one of the most challenging and rewarding experiences in my life. Professor Sayed's high standards in research, careful review of my papers, and numerous valuable suggestions greatly enhanced the quality of my work. Moreover, some of his advice such as "focus on good work" and "Do not simply follow the other people's ideas, figure it out your own way" have become my motto at work.

I am also grateful to Prof. Wotao Yin for his generous help and insightful discussions during my five years stay at UCLA. It is always fun and pleasant to work with him. I would also like to thank Prof. Lara Dolecek and Prof. Lieven Vandenberghe for serving on my doctoral committee, and for their valuable advice and feedback during the thesis defense.

My research would not have been easy without having great and supportive collaborators: Bicheng Ying, Sulaiman A. Alghunaim, Stefan Vlaski, Tianyu Wu, Lucas Cassano, Ernest K. Ryu, Jiageng Liu, Xianghui Mao, Prof. Qing Ling and Prof. Yuantao Gu. I learned a lot from all of them. I also appreciate the time I spent with many good friends at UCLA, EPFL and Microsoft Research. They are Zhimin Peng, Lanchao Liu, Jialin Liu, Yanli Liu, Hanqin Cai, Fei Feng, Yuejiao Sun, Robert Hannah, Hawraa Salami, Chung-Kai Yu, Chengcheng Wang, Jianshu Chen, Edward Nguyen, Eric Tan, Gabrielle Robertson, Roula Nassif, Xiujun Li, Zhirui Zhang, Shuohang Wang and Jinchao Li.

Furthermore, I want to thank my family for their unconditional support. In particular, I wish to express my deepest love to my wife Yang and my son Max. It is them who motivate me to keep working hard and moving forward.

This dissertation is based on work partially supported by the National Science Foundation under grants CCF-1524250 and ECCS-1407712. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

VITA

2007–2011 B.S., Dept. Electrical and Engineering, Xidian University, Xi'an, China.

2011–2014 M.S., Dept. Electrical and Engineering, University of Science and Technology of China (USTC), Hefei, China.

2014–2019 Research Assistant, Dept. Electrical and Computer Engineering, University of California, Los Angeles (UCLA), CA, USA

2018 Intern, Microsoft Research, Redmond, WA, USA

2018 Visiting Student, Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland

PUBLICATIONS

S. A. Alghunaim, K. Yuan, A. H. Sayed, "A linearly convergent proximal gradient algorithm for decentralized optimization", *Proc. Neural Information Processing Systems* (NeurIPS), 2019.

B. Ying, K. Yuan, A. H. Sayed, "Dynamic average diffusion with randomized coordinate updates", *to appear in IEEE Trans. Signal and Information Processing over Networks*, 2019.

S. A. Alghunaim, K. Yuan, A. H. Sayed, "A proximal diffusion strategy for multi-agent optimization with sparse affine constraints", *to appear in IEEE Trans. Automatic Control*,

2019.

K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion for distributed optimization and learning – Part I: Algorithm development," *IEEE Trans. Signal Processing*, vol. 67, no. 3, pp. 708-723, Feb. 2019.

K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion for distributed optimization and learning – Part II: Convergence analysis," *IEEE Trans. Signal Processing*, vol. 67, no. 3, pp. 724-739, Feb. 2019.

B. Ying, K. Yuan, and A. H. Sayed, "Supervised learning under distributed features," *IEEE Trans. Signal Processing*, vol. 67, no. 4, pp. 977-992, Feb. 2019.

B. Ying, K. Yuan, S. Vlaski, and A. H. Sayed, "Stochastic learning under random reshuffling with constant step-sizes," *IEEE Trans. Signal Processing*, vol. 67, no. 2, pp. 474-489, Jan. 2019.

K. Yuan, B. Ying, J. Liu, A. H. Sayed, "Variance-reduced stochastic learning by networked agents under random reshuffling", *IEEE Trans. Signal Processing*, vol. 67, no. 2, pp. 351-366, Jan. 2019.

T. Wu, K. Yuan, Q. Ling, W. Yin, and A. H. Sayed, "Decentralized consensus optimization with asynchrony and delays," *IEEE Trans. Signal and Information Processing over Networks*, vol. 4, no. 2, pp. 293-307, Jun. 2018.

K. Yuan, B. Ying, and A. H. Sayed, "On the influence of momentum acceleration on online learning," *Journal of Machine Learning Research*, vol. 17, no. 192, pp. 1-66, 2016.

CHAPTER 1

Introduction

In this dissertation, we study optimization, adaptation, and learning problems over connected networks. In these problems, each agent k collects and learns from its own local data and is able to communicate with its local neighbors. While each single node in the network may not be capable of sophisticated behavior on its own, it is the interaction among the constituents that leads to a powerful system that is able to solve large-scale and more challenging problems [1, 4].

Different approaches have been proposed in the literature to boost the learning capabilities of networked agents. Among them, the class of diffusion strategies [5–13] has been shown to be particularly well-suited due to their improved stability range over other methods and enhanced performance in adaptive scenarios. In particular, references [1, 4] study diffusion closely and explain how the diffusion strategy (a) performs distributed *optimization* over networks; (b) performs distributed *adaptation* over networks; (c) and performs distributed *learning* over networks. By quantifying the behavior of the algorithm, it is shown that diffusion will improve the averaged performance across the network.

It is known that diffusion implementations suffer from a small inherent bias [14]. When employing a constant step-size to solve a deterministic optimization problem, the iterates generated by the diffusion strategy will converge to a small neighborhood around the desired global solution but not to the exact solution itself. This inherent bias is not due to any gradient noise arising from stochastic approximation; it is instead due to the update structure in diffusion implementations [15, 16]. The existence of the bias in diffusion leads to three questions:

1. What is the origin of the bias?

2. Can we eliminate the bias?

3. Does the correction of the bias bring benefits to distributed optimization, distributed adaptation, or distributed learning?

In the coming chapters, we will present results that allow us to answer the above useful questions in the affirmative. To be specific, we will propose a new method *exact diffusion* that eliminates the inherent bias in Chapter 2. Furthermore, we will show *whether, when and why* exact diffusion can outperform diffusion for optimization, adaptation, and learning scenarios in Chapters 2–5. We will also compare the performance of exact diffusion to other state-of-the-art algorithms in the literature.

In this chapter, we will briefly discuss the problem formulation in distributed optimization, distributed adaptation and online learning, and distributed empirical machine learning, respectively. Next we will review the diffusion strategy in detail and present its various forms when solving problems in each of the above scenarios.

1.1 Problem Formulation

Consider a connected and undirected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of all networked nodes with $|\mathcal{V}| = K$ while $\mathcal{E}$ is the set of all edges. The optimization problem defined over this network is to let each agent operate cooperatively to solve a problem of a form

$$\min_{w \in \mathbb{R}^M} \quad \mathcal{J}^o(w) = \frac{1}{K} \sum_{k=1}^{K} J_k(w), \tag{1.1}$$

where M is the dimension of the variable, $J_k(w) : \mathbb{R}^M \to \mathbb{R}$ is a convex and differentiable cost function at agent k, and $\mathcal{J}^o(w) : \mathbb{R}^M \to \mathbb{R}$ is the global cost function. We let w^o denote the global minimizer of problem (1.1). While each agent can only access the local cost function $J_k(w)$, the target of the network is to let all agents collaborate to seek the global solution w^o. Different from the centralized network topology, e.g., the parameter server [17, 18], where there is a central node connected to all computing agents that is responsible for aggregating and scattering local variables, the network topology considered in the dissertation can take

arbitrary form such as line, cycle, grid, or random geometric graphs. There exists no central node in the considered network topology, and each agent will exchange information with their directly-connected neighbors rather than with a central agent. An illustration of such multi-agent network is shown in Fig. 1.1.

There are many advantages to distributed processing. First, the communication in distributed algorithms is more balanced. With each agent exchanging information with its neighbors, the communication load is evenly distributed over the edges. This is in contrast to centralized algorithms that usually induce great traffic jam on the central node. When the bandwidth around the central server is limited, the performance of centralized algorithms can be significantly degraded. Second, distributed algorithms are more robust to failure of agents. Note that each agent in a distributed strategy plays the same role by conducting the same operations – they update local variables and exchange information with neighbors. When one agent is down, the other agents can still work normally provided the network remains connected. In comparison, centralized strategies are more sensitive to the collapse of the central node which coordinates the computation and communication of all agents. Third, in real-time applications where agents collect data continuously, the repeated exchange of information back and forth between the agents and the fusion center can be costly especially when these exchanges occur over wireless links or require nontrivial routing resources. Finally, in some sensitive applications, agents may be reluctant to share their data with remote centers for various reasons including privacy and secrecy considerations.

Problem (1.1) is quite general and it covers various important scenarios by choosing different forms for $J_k(w)$. We next discuss these scenarios and their applications.

1.1.1 Distributed Optimization

When each local cost function $J_k(w)$ is known and its gradient $\nabla J_k(w)$ can be accessed easily, we regard (1.1) as a distributed optimization problem. This is a deterministic setting and no gradient noise exists to pollute $\nabla J_k(w)$. Distributed Optimization is the foundation to distributed adaptation and learning problems, and its study usually provides strong insights

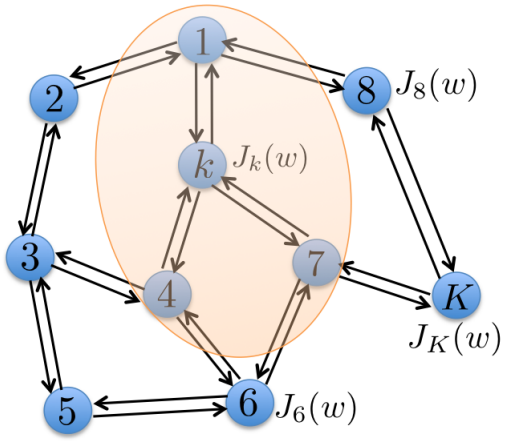

Figure 1.1: An illustration of the network. The network is connected, and each agent holds a local cost function $J_k(w)$. The arrow refers to communication. For example, agent k can send/receive information to/from its immediate neighbors $\{1, 4, 7\}$. The yellow shadow indicates the neighboring set of agent k.

into the latter scenarios. Distributed optimization finds applications in a wide range of areas in signal processing, control and communication including wireless sensor networks [19–23], event detection [24,25], spectrum sensing of cognitive radios [26,27], multi-vehicle and multi-robot control systems [28,29], cyber-physical systems and smart grid implementations [30–33], and many others.

1.1.2 Distributed Adaptation and Online Learning

If $J_k(w)$ is defined as the expectation of some loss function, then problem (1.1) falls into the scenario of distributed adaptation and online learning. To be concrete, distributed adaptation and online learning consider problems of the form

$$\min_{w\in\mathbb{R}^M} \; J(w) = \frac{1}{K}\sum_{k=1}^{K} J_k(w), \quad \text{where} \quad J_k(w) = \mathbb{E}\, Q(w;\boldsymbol{x}_k). \tag{1.2}$$

The random variable $\boldsymbol{x}_k$ represents the streaming data observed by agent k, and $Q(w;\boldsymbol{x}_k)$ is some loss function such as least-squares or the logistic function. Since the distribution of data $\boldsymbol{x}_k$ is generally unknown in advance, we cannot access the cost function $J_k(w)$ and its gradient $\nabla J_k(w)$; instead, we can only access $Q(w;\boldsymbol{x}_{k,i})$ and $\nabla Q(w;\boldsymbol{x}_{k,i})$ where $\boldsymbol{x}_{k,i}$ is the

realization of data $\boldsymbol{x}_k$ at iteration i. Also, throughout the adaptation setting we assume data samples $\boldsymbol{x}_{k,i}$ keep streaming in and the underlying distribution may *drift* with time. Such drifting distribution may cause a shift in the location of the global minimizer w^o, and one has to design strategies that enable agents to respond in real-time to drifts in data.

Problems of the form (1.2) are prevalent in adaptation and online learning contexts. Typical applications can be found in distributed estimation [1,4,14,34–36], dictionary learning [37–39], clustering [40, 41], multi-task learning [42], distributed feature learning [43], multi-target tracking [44, 45], social learning [46, 47], and multi-agent reinforcement learning [48–51].

1.1.3 Distributed Empirical Machine Learning

Many machine learning problems can be modeled as the empirical risk minimization

$$\min_{w\in\mathbb{R}^M} \ \frac{1}{N}\sum_{n=1}^{N} Q(w;x_n) \tag{1.3}$$

where x_n is the n-th data, N is the size of the dataset, and $Q(w;x_n)$ is some loss function as we discussed in Sec. 1.1.2. When the data size N is very large, it is usually intractable or inefficient to solve problem (1.3) with a single machine. To relieve this difficulty, one solution is to divide the N data samples across multiple machines and solve problem (1.3) in a cooperative manner. To this end, we consider K agents that are connected over the graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$. For each agent k, we assign $L=N/K$ data samples to it, which we denote by $\{x_{k,n}\}_{n=1}^{L}$. That is, it holds that $\{x_n\}_{n=1}^{N}=\{\{x_{1,n}\}_{n=1}^{L},\cdots,\{x_{K,n}\}_{n=1}^{L}\}$. One can verify that the empirical risk minimization problem (1.3) is equivalent to

$$\min_{w\in\mathbb{R}^M} \quad \frac{1}{K}\sum_{k=1}^{K} J_k(w) \quad \text{where} \quad J_k(w)=\frac{1}{L}\sum_{n=1}^{L} Q(w;x_{k,n}). \tag{1.4}$$

We regard problem (1.4) as the distributed empirical machine learning problem because it deals with finite, non-streaming, data samples. Since the data samples are fixed, the solution to problem (1.4) is also static.

In large-scale machine learning problems with enormous data to be processed, the number of computing agents K is usually far less than the sample size N. In this case, the size of

the local dataset L can still be very large. Note that each $J_k(w)$ in (1.4) is the average of L local loss functions, and its gradient $\nabla J_k(w) = \frac{1}{L}\sum_{n=1}^{L} \nabla Q(w; x_{k,n})$ is expensive to calculate especially for large L. Therefore, one usually employs the gradient over a single data sample $\nabla Q(w; x_{k,n})$, or the gradient over a batch of data samples $\frac{1}{B}\sum_{b=1}^{B} \nabla Q(w; x_{k,b})$, to approximate the real gradient $\nabla J_k(w)$. This constitutes the major difference between algorithms in distributed optimization and in distributed empirical machine learning.

Distributed machine learning over the centralized network, i.e., a network with a central node that is connected to all nodes, is well-studied to speed up training efficiency when large data samples exist. Many useful algorithms exist such as parallel stochastic gradient descent (SGD) methods [52, 53], distributed second-order methods [54–56], parallel dual coordinate methods [57, 58], and distributed alternating direction method of multipliers (ADMM) [59, 60]. However, the information congestion around the central node limits the speedup of these centralized methods, and this motivates great interest in distributed algorithms. For example, references [61, 62] find distributed algorithms, by eliminating the central node, are empirically shown to converge faster than centralized counterparts in deep learning when the network has limited bandwidth or high latency.

Another attractive emerging application for distributed empirical machine learning is federated learning [63, 64]. Federated learning is a distributed training approach which enables personal devices, e.g., mobile phones, tablets, and the wearables, located at different geographical positions to collaboratively learn a global machine learning model while keeping all the private data on the device. Current research mainly employs the centralized approaches for federated learning, see [63–66]. However, the convergence time for the federated learning process is significantly slow due to the limited communication bandwidth around the central server and the long communication latency between server and clients. As a result, distributed approaches without central server are introduced for federated learning [67–69] to speed up the convergence with more balanced communication load and short communication ranges.

1.2 Diffusion Learning

Research in distributed optimization dates back several decades (see, e.g., [70] and the references therein). In recent years, various centralized optimization methods such as (sub-)gradient descent, proximal gradient descent, (quasi-)Newton method, dual averaging, alternating direction method of multipliers (ADMM), and many other primal-dual methods have been extended to the distributed setting.

Distributed algorithms that are based on gradient-descent methods are effective and easy to implement. There are at least two prominent variants under this class: the consensus strategy [5–13] and the diffusion strategy [1, 4, 34, 36, 71]. There is a subtle but critical difference in the order in which computations are performed under these two strategies. In the consensus implementation, each agent runs a gradient-descent type iteration, albeit one where the starting point for the recursion and the point at which the gradient is approximated are *not* identical. This construction introduces an asymmetry into the update relation, which has some undesirable instability consequences (described, for example, in Secs. 7.2–7.3, Example 8.4, and also in Theorem 9.3 of [1] and Sec. V.B and Example 20 of [4]). The diffusion strategy, in comparison, employs a symmetric update where the starting point for the iteration and the point at which the gradient is approximated coincide. This property results in a wider stability range for diffusion strategies [1, 4].

In this section we review the diffusion learning algorithm and its recursions in various problem settings. In particular, we will reveal that diffusion, similar to consensus, suffers from an inherent limiting bias which may deteriorate its steady-state performance. This motivates us to study approaches that remove bias.

1.2.1 Diffusion for Distributed Optimization

To proceed, we consider solving problem (1.1) over a *connected* network of agents. The standard diffusion strategy [1, 4, 34] is listed in Algorithm 1.1 where the first step (1.5) conducts local gradient descent with constant step-size μ, and the second step (1.6) conducts weighted averaging for received information with $a_{\ell k}$ to scale variables flowing from agent ℓ

Algorithm 1.1 Diffusion strategy for distributed optimization at agent k

Setting: Initialize $w_{k,-1}$ arbitrarily.

Repeat for $i = 0, 1, 2, \cdots$

$$\psi_{k,i} = w_{k,i-1} - \mu \nabla J_k(w_{k,i-1}), \qquad \text{(adaptation)} \tag{1.5}$$

$$w_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \psi_{\ell,i}. \qquad \text{(combination)} \tag{1.6}$$

to k. The weights $\{a_{\ell k}\}_{\ell=1,k=1}^{K}$ are nonnegative and they satisfy

$$a_{\ell k} \begin{cases} \geq 0 & \text{if agents } \ell \text{ and } k \text{ are connected,} \\ = 0 & \text{if agents } \ell \text{ and } k \text{ are not connected,} \end{cases} \qquad a_{\ell k} = a_{k\ell}, \quad \text{and} \quad \sum_{\ell \in \mathcal{N}_k} a_{\ell k} = 1 \tag{1.7}$$

With condition (1.7), it follows that the weight matrix $A = [a_{\ell k}] \in \mathbb{R}^{K \times K}$ is a symmetric and doubly-stochastic matrix, i.e.,

$$A = A^{\mathsf{T}} \quad \text{and} \quad A \mathbb{1}_K = \mathbb{1}_K. \tag{1.8}$$

Moreover, $\mathcal{N}_k$ in (1.6) denotes the set of neighbors of agent k (including agent k itself), and $\nabla J_k(\cdot)$ denotes the gradient vector of $J_k(\cdot)$ relative to w. The combination step (1.6) is illustrated in Fig. 1.2.

Remark 1.1 (Combination matrix) *While we consider the symmetric and doubly stochastic combination matrix satisfying condition* (1.8) *for simplicity in this section, this is not a requirement for diffusion to converge to preferable solutions. In fact, diffusion can employ more relaxed left-stochastic combination matrices as explained in [1, 4]. We will examine the combination matrices employed in diffusion more closely in Chapter 2.* ■

When sufficiently small step-sizes are employed to drive the optimization process, the diffusion strategy is able to converge exponentially fast when $\mathcal{J}^o(w)$ is strongly convex, albeit only to *an approximate solution* [1, 9]. Specifically, it is proved by Theorem 3 in [14]

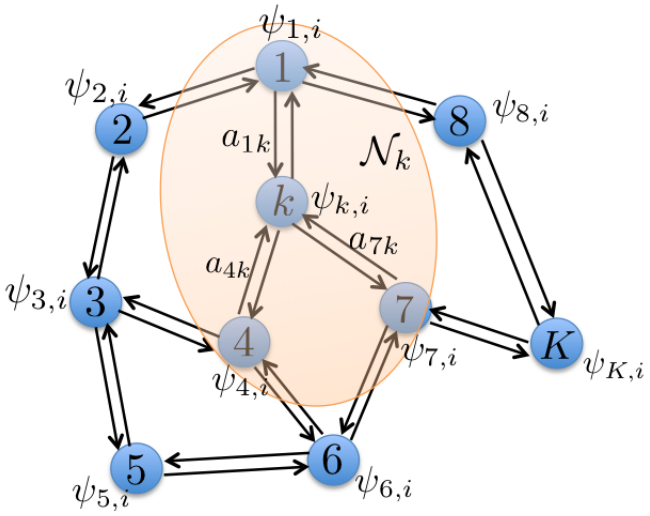

Figure 1.2: An illustration of the combination step (1.6) in diffusion method. Since $\mathcal{N}_k = \{1, 4, 7, k\}$, it holds that $w_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \psi_{\ell,i} = a_{1k}\psi_{1,i} + a_{4k}\psi_{4,i} + a_{7k}\psi_{7,i} + a_{kk}\psi_{k,i}$.

that iterates $w_{k,i}$ generated through the diffusion recursion (1.5)-(1.6) will approach w^o, i.e.,

$$\limsup_{i \to \infty} \|w^o - w_{k,i}\|^2 = O(\mu^2), \quad \forall\, k = 1, \cdots, K, \tag{1.9}$$

where $w_{k,i}$ denotes the local iterate at agent k and iteration i. Result (1.9) implies that the diffusion method will converge to a neighborhood around w^o, and that the square-error bias is small (since μ is usually small) and on the order of $O(\mu^2)$. Note that this limiting bias $O(\mu^2)$ is not due to any gradient noise arising from stochastic approximations; it is instead due to the inherent structure of the diffusion updates. The existence of the limiting bias can be justified by the following simple example.

Example 1.1 (Diffusion has inherent bias) We assume all iterates $\{w_{k,i-1}\}_{k=1}^K$ are staying at the global solution w^o at current iteration $i-1$, and we examine the next iterate $w_{k,i}$. Substituting recursions (1.5) into (1.6), we get

$$\begin{aligned} w_{k,i} &= \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \left(w_{\ell,i-1} - \mu \nabla J_\ell(w_{\ell,i-1}) \right) \\ &= \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \left(w^o - \mu \nabla J_\ell(w^o) \right) \overset{(a)}{=} w^o - \mu \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \nabla J_\ell(w^o) \neq w^0 \end{aligned} \tag{1.10}$$

Algorithm 1.2 Diffusion strategy for distributed adaptation and online learning

Setting: Initialize $w_{k,-1}$ arbitrarily.

Each agent k will **repeat for** $i = 0, 1, 2, \cdots$

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu \nabla Q(\boldsymbol{w}_{k,i-1}; \boldsymbol{x}_{k,i}), \qquad \text{(adaptation)} \tag{1.11}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\psi}_{\ell,i}. \qquad \text{(combination)} \tag{1.12}$$

where equality (a) holds since $\sum_{\ell \in \mathcal{N}_k} a_{\ell k} = 1$ and the last inequality holds since

$$\sum_{\ell \in \mathcal{N}_k} a_{\ell k} \nabla J_\ell(w^o) \neq 0$$

in general. This implies that even if all iterates have converged to w^o at some iteration, they will jump away from w^o at the next iteration. As a result, exact diffusion cannot converge to the exact solution w^o in the steady-state stage. ■

1.2.2 Diffusion for Distributed Adaptation and Online Learning

Now we employ the diffusion strategy to solve the distributed adaptation and online learning problem (1.2). Since $J_k(w)$ is constructed as the expectation of the random loss $Q(w; \boldsymbol{x}_k)$ and the distribution of $\boldsymbol{x}_k$ is unknown in general, the real gradient $\nabla J_k(w)$ cannot be accessed. For this scenario, exact diffusion will employ the stochastic gradient $\nabla Q(w; \boldsymbol{x}_{k,i})$ where $\boldsymbol{x}_{k,i}$ is the realization of $\boldsymbol{x}_k$ at iteration i to approximate the real gradient. The recursion of diffusion for distributed adaptation and online learning is listed in Algorithm 1.2. Comparing (1.11) with (1.5), it is observed that diffusion employs stochastic gradient descent (SGD) in the combination step. For adaptation and online learning, we employ a constant step-size μ to enable continuous adaptation and learning in response to drifts in the location of the global minimizer due to changes in the statistical properties of the data.

Previous studies have shown that diffusion methods (1.11)–(1.12) are able to solve problems of the type (1.2) well for sufficiently small step-sizes. In particular, when each $J_k(w)$

is smooth with Lipschitz continuous gradient, the global cost function $\mathcal{J}^o(w)$ is strongly convex, and the stochastic gradient noise is unbiased with controllable variance, it is proved in, for example, Lemma 5 in [72] or Theorem 9.1 in [1] that

$$\limsup_{i\to\infty} \mathbb{E}\|w^o - \boldsymbol{w}_{k,i}\|^2 = O(\mu\sigma^2 + \mu^2 b^2), \quad \forall\ k = 1, \cdots, K \tag{1.13}$$

for sufficiently small step-sizes, where σ^2 is the magnitude of the gradient noise, and $b^2 = \sum_{k=1}^{K} \|\nabla J_k(w^o)\|^2$ is a bias constant. Result (1.13) has two important implications:

- When there is no gradient noise, i.e., $\sigma^2 = 0$, the adaptive diffusion recursions (1.11)–(1.12) reduce to the deterministic diffusion recursions (1.5)–(1.6). In this scenario, result (1.13) becomes

$$\limsup_{i\to\infty} \mathbb{E}\|w^o - \boldsymbol{w}_{k,i}\|^2 = O(\mu^2 b^2), \quad \forall\ k = 1, \cdots, K \tag{1.14}$$

  which is consistent with the convergence property shown in (1.9). The term $O(\mu^2 b^2)$ is exactly the inherent limiting bias suffered by diffusion.

- When step-size is sufficiently small, the $O(\mu\sigma^2)$ limiting bias will dominate the inherent bias $O(\mu^2 b^2)$. That is,

$$\limsup_{i\to\infty} \mathbb{E}\|w^o - \boldsymbol{w}_{k,i}\|^2 = O(\mu\sigma^2), \quad \forall\ k = 1, \cdots, K. \tag{1.15}$$

  This is a well-known result for the steady-state performance for diffusion. Note that the $O(\mu\sigma^2)$ limiting bias arises from the gradient noise.

1.2.3 Diffusion for Distributed Empirical Machine Learning

In this subsection we extend diffusion to solve the empirical machine learning problem (1.4) in a distributed manner. The diffusion recursion can be easily derived by interpreting (1.4) as a special form of the adaptation problem (1.2) [73]. Recall that each agent k stores data samples $\{x_{k,n}\}_{n=1}^{L}$. We introduce a discrete random variable $\boldsymbol{x}_k$ having these samples as

realizations and a uniform probability mass function (pmf) defined by

$$p(\boldsymbol{x}_k) = \begin{cases} \frac{1}{L}, & \text{if } \boldsymbol{x}_k = x_{k,1}, \\ \vdots & \vdots \\ \frac{1}{L}, & \text{if } \boldsymbol{x}_k = x_{k,L}. \end{cases} \tag{1.16}$$

With the uniform pmf in (1.16), it holds that

$$\frac{1}{L}\sum_{n=1}^{L} Q(w; x_{k,n}) = \mathbb{E}\ Q(w; \boldsymbol{x}_k) \tag{1.17}$$

and hence problem (1.4) can be rewritten as

$$\min_{w\in\mathbb{R}^M} \quad \frac{1}{K}\sum_{k=1}^{K} J_k(w) \quad \text{where} \quad J_k(w) = \mathbb{E}\ Q(w; \boldsymbol{x}_k) = \frac{1}{L}\sum_{n=1}^{L} Q(w; x_{k,n}) \tag{1.18}$$

and the distribution of $\boldsymbol{x}_k$ is defined in (1.16). This implies that the distributed empirical machine learning problem (1.4) is essentially a special form of adaptation and online learning problem (1.18).

We know from Sec. 1.2.2 that diffusion can solve problem (1.18) with recursions

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\nabla Q(\boldsymbol{w}_{k,i-1}; \boldsymbol{x}_{k,i}), \qquad \text{(adaptation)} \tag{1.19}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\psi}_{\ell,i}. \qquad \text{(combination)} \tag{1.20}$$

where the notation $\boldsymbol{x}_{k,i}$ represents the realization of $\boldsymbol{x}_k$ that streams in at iteration i. Since $\boldsymbol{x}_{k,i}$ is selected from $\{x_1, x_2, \cdots, x_N\}$ at iteration i according to the pmf (1.16), we can rewrite $\boldsymbol{x}_{k,i}$ as $x_{\boldsymbol{n}_i}$ and replace (1.19) by

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\nabla Q(\boldsymbol{w}_{k,i-1}; x_{k,\boldsymbol{n}_i}). \tag{1.21}$$

Here, the variable $\boldsymbol{n}_i$ is a uniform discrete random variable indicating the index of the sample that is picked at iteration i. The diffusion strategy for the distributed empirical learning problem (1.4) can therefore be summarized as Algorithm 1.3.

With equivalence between (1.19) and (1.21), we conclude that the diffusion recursions (1.22)–(1.24) are essentially the recursion (1.11)–(1.12) applied to problem (1.18). This

Algorithm 1.3 Diffusion strategy for distributed empirical learning at agent k

Setting: Initialize $w_{k,-1}$ arbitrarily.

Repeat for $i = 0, 1, 2, \cdots$

$$\boldsymbol{n}_i \sim \mathcal{U}[1, L] \qquad \text{(uniformly sample integer from 1 to } L\text{)} \tag{1.22}$$

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu \nabla Q(\boldsymbol{w}_{k,i-1}; x_{k,\boldsymbol{n}_i}), \qquad \text{(adaptation)} \tag{1.23}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\psi}_{\ell,i}. \qquad \text{(combination)} \tag{1.24}$$

interpretation is useful because we can now call upon results from Sec.1.2.2 and apply them to characterize the performance of recursions (1.22)–(1.24). Following this analysis, we can show that the iterates $\boldsymbol{w}_{k,i}$ generated by (1.22)–(1.24) satisfy

$$\limsup_{i \to \infty} \mathbb{E}\|w^o - \boldsymbol{w}_{k,i}\|^2 = O(\mu\sigma^2 + \mu^2 b^2), \quad \forall\ k = 1, \cdots, K \tag{1.25}$$

for sufficiently small step-sizes μ under the same assumptions as in Sec.1.2.2.

1.3 Objectives and Organization

In future chapters, we will answer the three questions listed in the beginning of Sec.1. To be concrete, we will identify the origin of the bias suffered by diffusion and develop a new exact diffusion learning strategy to correct it. We will study its convergence condition and performance for the scenarios of distributed optimization, distributed adaptation and online learning, and distributed empirical learning, respectively. Specifically, we will compare the behavior of diffusion and exact diffusion in each scenario to corroborate the benefits of removing the inherent bias. Furthermore, we will also compare exact diffusion with other well-known distributed methods and show its superiority in stability range, convergence rate, steady-state performance, or memory cost.

The organization of the dissertation is summarized as follows.

- **Chapter 2.** In this chapter, we clarify the origin of the $O(\mu^2)$ inherent bias in the standard diffusion strategy. It turns out that diffusion relies on reformulating the aggregate optimization problem (1.1) as a penalized problem and resorting to a diagonally-weighted incremental construction. Since the achieved penalized problem is just an approximation to problem (1.1), diffusion can only converge to an approximate solution rather than the desired w^o. We next develop the exact diffusion method that directly solves the real problem (1.1) and thus eliminates the limiting bias. We also show in this chapter that the exact diffusion method is applicable to locally-balanced left-stochastic combination matrices which, compared to the conventional doubly-stochastic matrix, are more general and able to endow the algorithm with faster convergence rate, more flexible step-size choices and better privacy-preserving properties. In particular, the simulation shows exact diffusion with a locally-balanced combination matrix converges much faster than the benchmark method EXTRA [74] using the doubly-stochastic matrix.

- **Chapter 3.** In this chapter, we examine the convergence and stability properties of exact diffusion in detail and establish its linear convergence rate. We also show that it has a wider stability range than the EXTRA [75] consensus solution even if both algorithms employ the same symmetric and doubly-stochastic combination matrices, meaning that it is stable for a wider range of step-sizes and can, therefore, attain faster convergence rates. Analytical examples and numerical simulations illustrate the theoretical findings.

- **Chapter 4.** While the convergence property of exact diffusion is studied when solving distributed deterministic optimization problems in Chapters 2 and 3, its performance under adaptation and online learning settings remains unclear. It is still unknown whether the bias-correction is necessary over adaptive networks. By studying exact diffusion and examining its steady-state performance under stochastic scenarios, this chapter provides affirmative results. It is proved that the correction step in exact diffusion leads to a better steady-state performance than standard diffusion strategies

under mild conditions. It is also analytically shown the superiority of exact diffusion becomes more evident over sparse or badly-connected network topologies such as line, cycle, grid, and many others. This chapter also explores situations where exact diffusion and diffusion do perform similarly. These conclusions will provide a guideline on how to employ exact diffusion effectively in various applications.

- **Chapter 5.** In this chapter we extend exact diffusion to the empirical learning scenario with finite data samples. The problem considered in this chapter is more general than (1.4) in which the amount of data observed/collected by the individual agents may differ drastically, i.e., it is possible that $L_k \neq L_\ell$ for different agents k and ℓ. To guarantee linear convergence to the exact solution w^o, we integrate exact diffusion with an amortized variance-reduced gradient (AVRG) algorithm developed in [76]. AVRG is a stochastic variance-reduced method. Its memory cost is trivial compared to SAGA and it has a balanced gradient computations in comparison to SVRG. These two key advantages enable AVRG amenable to decentralized implementations. The resulting diffusion-AVRG algorithm is shown to have linear convergence to the exact solution which is opposed to the diffusion strategy that just converges to the neighborhood, see equation (1.25). Diffusion-AVRG is also shown much more memory efficient than the other alternative algorithms such as DSA [77]. In addition, we propose a mini-batch strategy to balance the communication and computation efficiency for diffusion-AVRG. When a proper batch size is selected, it is observed in simulations that diffusion-AVRG is more computationally efficient than exact diffusion or EXTRA while maintaining almost the same communication efficiency.

- **Chapter 6.** This chapter will summarize all derived results in the dissertation and discuss future work on exact diffusion learning.

1.4 Notation

Throughout the dissertation we use $\text{diag}\{x_1, \cdots, x_K\}$ to denote a diagonal matrix consisting of diagonal entries $x_1, \cdots, x_R$, and use $\text{col}\{x_1, \cdots, x_R\}$ to denote a column vector formed by stacking $x_1, \cdots, x_R$. For symmetric matrices X and Y, the notation $X \leq Y$ or $Y \geq X$ denotes $Y - X$ is positive semi-definite. For a vector x, the notation $x \succeq 0$ denotes that each element of x is non-negative, while the notation $x \succ 0$ denotes that each element of x is positive. For a matrix X, we let $\text{range}(X)$ denote its range space, and $\text{null}(X)$ denote its null space. The notation $\mathbb{1}_K = \text{col}\{1, \cdots, 1\} \in \mathbb{R}^K$.

CHAPTER 2

Exact Diffusion for Distributed Optimization: Algorithm Development

2.1 Context and Background

This chapter deals with *deterministic* optimization problems where a collection of K networked agents operate cooperatively to solve an aggregate optimization problem of the form:

$$w^\star = \arg\min_{w\in\mathbb{R}^M} \quad \mathcal{J}^\star(w) = \sum_{k=1}^{K} q_k J_k(w). \tag{2.1}$$

In this formulation, each risk function $J_k(w)$ is convex and differentiable, while the aggregate cost $J(w)$ is strongly-convex. Note that problem (2.1) is more general than the original problem (1.1). The weights $\{q_k\}_{k=1}^K$ are given positive constants to scale each local cost function. When $q_1 = \cdots = q_K = 1/K$, problem (2.1) is equivalent to (1.1). All agents seek to determine the unique global minimizer, $w^\star$, under the constraint that agents can only communicate with their neighbors. This distributed approach is robust to failure of links and/or agents and scalable to the network size. Optimization problems of this type find applications in a wide range of areas, see the discussion in Sec. 1.1.1.

2.1.1 Related Work

Research in distributed optimization dates back several decades (see, e.g., [70] and the references therein). In recent years, various centralized optimization methods such as (sub-)gradient descent, proximal gradient descent, (quasi-)Newton method, dual averaging, alternating direction method of multipliers (ADMM), and many other primal-dual methods have been extended to the distributed setting. In this section, we review several classes of

distributed algorithms that can be used to solve problem (1.1).

2.1.1.1 Distributed Primal Methods

In the primal domain, implementations that are based on gradient-descent methods are effective and easy to implement. There are at least two prominent variants under this class: the consensus strategy [5–13] and the diffusion strategy [1,4,34,36,71]. A brief description of these two primal strategies is given in Appendix 2.A. There is a subtle but critical difference in the order in which computations are performed under these two strategies. In the consensus implementation, each agent runs a gradient-descent type iteration, albeit one where the starting point for the recursion and the point at which the gradient is approximated are *not* identical. This construction introduces an asymmetry into the update relation, which has some undesirable instability consequences (described, for example, in Secs. 7.2–7.3, Example 8.4, and also in Theorem 9.3 of [1] and Sec. V.B and Example 20 of [4]). The diffusion strategy, in comparison, employs a symmetric update where the starting point for the iteration and the point at which the gradient is approximated coincide. This property results in a wider stability range for diffusion strategies [1, 4]. Still, when sufficiently small step-sizes are employed to drive the optimization process, both types of strategies (consensus and diffusion) are able to converge exponentially fast, albeit only to *an approximate solution* [1, 9]. Specifically, it is proved in [1, 9, 14] that both the consensus and diffusion iterates under constant step-size learning converge towards a neighborhood of square-error size $O(\mu^2)$ around the true optimizer, $w^\star$, i.e., $\|w^\star - w_{k,i}\|^2 = O(\mu^2)$ as $i \to \infty$, where μ denotes the step-size and $w_{k,i}$ denotes the local iterate at agent k and iteration i. This limiting $O(\mu^2)$ bias is not due to any gradient noise arising from stochastic approximations; it is instead due to the inherent structure of the consensus and diffusion updates as clarified in the sequel.

Second-order information such as the Hessian matrix can also be introduced to the primal methods, see the distributed Newton method [78, 79], Quasi-Newton method [80] and references therein. While the Hessian matrix helps accelerate the convergence rate, these second-order algorithms still suffer from the $O(\mu^2)$ inherent limiting bias. There is another

type of methods that employ multi-consensus inner loop [81–83] and thus improves the consensus of the variables at each outer iteration. While these two-time scale methods can reduce the limiting bias, the inner consensus loop incurs more communication rounds between agents, and hence slows down the processing of new data received in the outer loop. For this reason, they are not well-suited for the adaptation and online learning problems.

2.1.1.2 Distributed Primal-Dual Methods

Another important class of distributed algorithms are based on the primal dual strategies. A brief analytical derivation of various popular primal-dual methods is given in Sec. 2.B. A well-known family of distributed primal dual methods are those based on alternating direction method of multipliers (ADMM) [74, 84–86] and its variants [87–90]. In particular, work [74] proves that distributed ADMM with constant parameters converges exponentially fast to the *exact* global solution $w^\star$, which is in contrast to the purely primal methods we discussed in Sec. 2.1.1.1 that only converge to an *approximate* solution close to $w^\star$ with constant step-sizes. However, distributed ADMM solutions are computationally more expensive since they necessitate the solution of optimal sub-problems at each iteration. Some useful variations of distributed ADMM [87–89] may alleviate the computational burden, but their recursions are still more difficult to implement than consensus or diffusion due to their primal dual structures.

In more recent work [75, 91], a modified implementation of consensus iterations, referred to as EXTRA, is proposed and shown to converge to the *exact* minimizer $w^\star$ rather than to an $O(\mu^2)-$neighborhood around $w^\star$. The modification has a similar computational burden as traditional consensus and is based on adding a step that combines two prior iterates to remove bias. While EXTRA does not explicitly employ a dual variable, it is essentially a primal dual saddle point algorithm [77]. Motivated by [75], other variations with similar properties were proposed in [92–98]. These variations rely instead on combining inexact gradient evaluations with a gradient tracking technique. The resulting algorithms, compared to EXTRA, have two information combinations per recursion, which doubles the amount of communication

variables compared to EXTRA, and can become a burden when communication resources are limited. Distributed primal-dual second-order methods are also studied in [89, 99] to reduce communication rounds but they suffer from the expensive construction of the Hessian matrix. Due to their easy implementations and fast convergences, EXTRA and tracking methods have been extended to other important scenarios for directed [93, 97, 98, 100, 101] and asynchronous [102] networks. There is also another family of primal-dual methods that are related to EXTRA and utilize the network structure to further accelerate the convergence and reduce the communication rounds [103–105].

When local cost function $J_k(w)$ is not smooth and has the structure $J_k(w) = s_k(w) + r_k(w)$ where $s_k(w)$ is smooth with Lipschitz continuous gradients and $r_k(w)$ is a possibly non-smooth regularization term, one can integrate the proximal gradient descent with the above primal-dual methods, see [88, 91, 106–109]. In particular, [109] proposes a distributed proximal gradient method that endows with exponential convergence to $w^\star$ when each agent shares the same regularization term , i.e., $r_1(w) = \cdots = r_K(w) = r(w)$.

2.1.1.3 Distributed dual methods

A third class of distributed algorithms are purely dual methods, see [110–113]. A short description on dual methods is provided in Sec.2.C. They first derive the unconstrained dual problem of problem (2.1) and then solve it by gradient descent. In particular, the algorithms of [110,111,113] can reach the optimal convergence rate by introducing Nesterov's acceleration to their recursions.

2.1.2 Motivation and Contributions

The current chapter is motivated by the following considerations. The result in [75] shows that the EXTRA technique resolves the bias problem in consensus implementations. However, it is known that traditional diffusion strategies outperform traditional consensus strategies. Would it be possible then to correct the bias in the diffusion implementation and attain an algorithm that is superior to EXTRA (e.g., an implementation that is more stable than

EXTRA)? This is one of the contributions in Chapter 2 and 3. In this chapter, we shall indeed develop a bias-free diffusion strategy that will be shown in chapter 3 to have a wider stability range than EXTRA consensus implementations. Achieving these objectives is challenging for several reasons. First, we need to understand the origin of the bias in diffusion implementations. Compared to the consensus strategy, the source of this bias is different and still not well understood. In seeking an answer to this question, we will initially observe that the diffusion recursion can be framed as an incremental algorithm to solve a penalized version of (2.1) and not (2.1) directly — see expression (2.71) further ahead. In other words, the local diffusion estimate $w_{k,i}$, held by agent k at iteration i, will be shown to approach the solution of a penalized problem rather than $w^\star$, which causes the bias.

We have four main contributions in this chapter and the accompanying chapter 3 relating to: (a) developing a distributed algorithm (which we refer to as exact diffusion) that ensures exact convergence based on the diffusion strategy; (b) showing that exact diffusion has wider stability range and enhanced performance than EXTRA [75]; (c) showing that exact diffusion works for the larger class of locally balanced (rather than only doubly-stochastic) matrices; and (d) showing that neither EXTRA nor exact diffusion can be extended to the general *directed* network by constructing counter examples, which helps illustrate the significance of the proposed locally balanced conditions.

More specifically, we will first show in this chapter how to modify the diffusion strategy such that it solves the real problem (2.1) directly. We shall refer to this variant as *exact diffusion*. Interestingly, the structure of *exact diffusion* will turn out to be very close to the structure of *standard* diffusion. The only difference is that there will be an extra "correction" step added between the usual "adaptation" and "combination" steps of diffusion — see the listing of Algorithm 1 further ahead. It will become clear that this adapt-correct-combine (ACC) structure of the exact diffusion algorithm is more symmetric in comparison to the EXTRA recursions. In addition, the computational cost of the "correction" step is trivial. Therefore, with essentially the same computational efficiency as standard diffusion, the exact diffusion algorithm will be able to converge *exponentially fast* to $w^\star$ without any bias. Secondly, we will show in Chapter 3 that exact diffusion has a wider stability range

than EXTRA. In other words, there will exist a larger range of step-sizes that keeps exact diffusion stable but not the EXTRA algorithm. This is an important observation because larger values for μ help accelerate convergence.

Our third contribution is that we will derive the exact diffusion algorithm, and establish its convergence property for the class of *locally balanced* combination matrices (see Definition 1). This class does not only include symmetric doubly-stochastic matrices as special cases, but it also includes a range of widely-used left-stochastic policies as explained further ahead. First, we recall that left-stochastic matrices are defined as follows. Let $a_{\ell k}$ denote the weight that is used to scale the data that flows from agent ℓ to k. Let $A \triangleq [a_{\ell k}] \in \mathbb{R}^{K\times K}$ denote the matrix that collects all these coefficients. The entries on each column of A are assumed to add up to one so that A is *left-stochastic*, i.e., it holds that

$$A^{\mathsf{T}}\mathbb{1}_K = \mathbb{1}_K, \quad \text{or} \quad \sum_{\ell=1}^{K} a_{\ell k} = 1, \quad \forall\, k = 1, \cdots, K. \tag{2.2}$$

The matrix A will not be required to be symmetric. For example, it may happen that $a_{\ell k} \neq a_{k\ell}$. Using these coefficients, when an agent k combines the iterates $\{\psi_{\ell,i}\}$ it receives from its neighbors, that combination will correspond to:

$$w_{k,i+1} = \sum_{\ell=1}^{K} a_{\ell k}\psi_{\ell,i}, \quad \text{where} \quad \sum_{\ell=1}^{K} a_{\ell k} = 1. \tag{2.3}$$

Obviously, $w_{k,i+1}$ is a convex combination of $\{\psi_{\ell,i}\}$.

It should be emphasized that condition (2.2), which is repeated in (2.3), is different from all previous algorithms studied in [5, 74, 75, 84, 85, 88, 95, 96], which require A to be symmetric and doubly stochastic (i.e., each of its columns and rows should add up to one). For undirected networks, although symmetric doubly-stochastic matrices are commonly used, balanced left-stochastic policies can have significant practical value — they can speed up convergence, a more relaxed selection of the step-size parameter, reach better mean-square-error (MSE) performance over adaptive networks, and enjoy better privacy-preserving properties — see the extended discussions in Sec. 2.2.3.

We also explain in this chapter the significance of the proposed locally balanced conditions. If the combination matrix does not satisfy these conditions, we show that one can

construct counter examples where both exact diffusion and EXTRA diverge for any given step-size (see Sec. 2.5). This implies an interesting conclusion: exact diffusion and EXTRA may not always work for general *directed* networks (see the discussions in Secs. 2.2.4 and 2.5). This seems to be a disadvantage in comparison with DIGing-based methods [92–96] which are designed for directed network. However, for scenarios where the locally balanced condition is satisfied, exact diffusion is shown in simulations to have a wider range of step-sizes and is more communication efficient than DIGing methods [92–96] (recall that in DIGing there are two information combinations per iteration).

In this chapter we derive the exact diffusion algorithm, while in next chapter we establish its convergence properties and prove its stability superiority over the EXTRA algorithm. This article is organized as follows. In Sec. 2.2 we review the standard diffusion algorithm, introduce locally-balanced left-stochastic combination policies, and establish several of their properties. In Sec. 2.3 we identify the source of bias in standard diffusion implementations. In Sec. 2.4 we design the exact diffusion algorithm to correct for the bias. In Sec.2.5 we illustrate the necessity of the locally-balanced condition on the combination policies by showing that divergence can occur if it is not satisfied. Numerical simulations are presented in Sec. 3.3.

2.2 Diffusion and Combination Policies

2.2.1 Standard Diffusion Strategy

To solve problem (2.1) over a *connected* network of agents, we consider the standard diffusion strategy [1, 4, 34]:

$$\psi_{k,i} = w_{k,i-1} - \mu_k \nabla J_k(w_{k,i-1}), \tag{2.4}$$

$$w_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \psi_{\ell,i}, \tag{2.5}$$

where $\{\mu_k\}_{k=1}^K$ are positive step-sizes. Compared to the diffusion method we present i Algorithm 1.1, Recursions (2.4)–(2.5) employ different local step-size μ_k for each agent k. These step-size setting will enable diffusion to converge towards the optimal solution $w^\star$ to

the weighted consensus problem in (2.1). Moreover, in this chapter we will consider using a more relaxed combination policy in diffusion than the symmetric and doubly-stochastic matrix used in Sec. 1.2. Specifically, we assume $\{a_{\ell k}\}_{\ell=1,k=1}^{K}$ are nonnegative combination weights satisfying

$$\sum_{\ell \in \mathcal{N}_k} a_{\ell k} = 1. \tag{2.6}$$

It follows from (2.6) that $A = [a_{\ell k}] \in \mathbb{R}^{K \times K}$ is a left-stochastic matrix, i.e., $A^{\mathsf{T}} \mathbb{1}_K = \mathbb{1}_K$. Note that we do not assume A is symmetric here. The benefits of left-stochastic combination matrix over symmetric and doubly stochastic matrix is discussed in Sec. 2.2.3.

It is assumed that the graph is strongly-connected in this chapter, which means that at least one diagonal entry of A is non-zero [1] (this is a reasonable assumption since it simply requires that at least one agent in the network has some confidence level in its own data). In this case, the matrix A will be primitive. This implies, in view of the Perron-Frobenius theorem [1, 114], that there exists an eigenvector p satisfying

$$Ap = p, \quad \mathbb{1}_K^{\mathsf{T}} p = 1, \;\; p \succ 0. \tag{2.7}$$

We refer to p as the Perron eigenvector of A. Next, we introduce the vector

$$q \;\triangleq\; \mathrm{col}\{q_1, q_2, \ldots, q_K\} \in \mathbb{R}^K, \tag{2.8}$$

where q_k is the weight associated with $J_k(w)$ in (2.1). Let the constant scalar β be chosen such that

$$q = \beta \, \mathrm{diag}\{\mu_1, \mu_2, \cdots, \mu_K\} p. \tag{2.9}$$

where $\beta > 0$ is some constant, then it was shown by Theorem 3 in [14] that under (2.9), the iterates $w_{k,i}$ generated through the diffusion recursion (1.5)-(1.6) will approach $w^\star$, i.e.,

$$\limsup_{i \to \infty} \|w^\star - w_{k,i}\|^2 = O(\mu_{\max}^2), \quad \forall \; k = 1, \cdots, K, \tag{2.10}$$

where $\mu_{\max} = \max\{\mu_1, \cdots, \mu_K\}$. Result (2.10) implies that the diffusion algorithm will converge to a neighborhood around $w^\star$, and that the square-error bias is on the order of $O(\mu_{\max}^2)$. We discuss a simple example in Sec.2.2 that justifies the existence of the inherent limiting bias.

Remark 2.1 (Scaling) *Condition* (2.9) *is not restrictive and can be satisfied for any left-stochastic matrix A through the choice of the parameter β and the step-sizes. Note that β should satisfy*

$$\beta = \frac{q_k}{p_k}\frac{1}{\mu_k} \tag{2.11}$$

for all k. To make the expression for β independent of k, we parameterize (select) the step-sizes as

$$\mu_k = \left(\frac{q_k}{p_k}\right)\mu_o \tag{2.12}$$

for some small $\mu_o > 0$. Then, $\beta = 1/\mu_o$, which is independent of k, and relation (2.9) *is satisfied.* ■

Remark 2.2 (Perron entries) *Expression* (2.12) *suggests that agent k needs to know the Perron entry p_k in order to run the diffusion strategy* (2.4)–(2.5)*. As we are going to see in the next section, the Perron entries are actually available beforehand and in closed-form for several well-known left-stochastic policies (see, e.g., expressions* (2.17)*,* (2.21)*, and* (2.26) *further ahead). For other left-stochastic policies for which closed-form expressions for the Perron entries may not be available, these can be determined iteratively by means of the power iteration — see, e.g., the explanation leading to future expression* (2.37)*.* ■

2.2.2 Combination Policy

Result (2.10) is a reassuring conclusion: it ensures that the squared-error is small whenever $\mu_{\max}$ is small; moreover, the result holds for *any* left-stochastic matrix. Moving forward, we will focus on an important subclass of left-stochastic matrices, namely, those that satisfy a mild *local balance* condition (we shall refer to these matrices as *balanced* left-stochastic policies) [115]. The balancing condition turns out to have a useful physical interpretation and, in addition, it will be shown to be satisfied by several widely used left-stochastic combination policies. The local balance condition will help endow networks with crucial properties to ensure exact convergence to $w^\star$ without any bias. In this way, we will be able to propose

distributed optimization strategies with exact convergence guarantees for this class of left-stochastic matrices, while EXTRA [75] is limited to (the less practical) doubly-stochastic policies; balanced left-stochastic matrices have many benefits as explained before, which is the main motivation for focusing on them in our treatment.

Definition 1 (Locally balanced Policies) *Let p denote the Perron eigenvector of a primitive left-stochastic matrix A, with entries $\{p_\ell\}$. Let $P = diag(p)$ correspond to the diagonal matrix constructed from p. The matrix A is said to satisfy a local balance condition if it holds that*

$$a_{\ell k}\, p_k = a_{k\ell}\, p_\ell, \quad k, \ell = 1, \cdots, K \tag{2.13}$$

or, equivalently, in matrix form:

$$PA^{\mathsf{T}} = AP. \tag{2.14}$$

■

Relations of the form (2.13) are common in the context of Markov chains. They are used there to model an equilibrium scenario for the probability flux into the Markov states [116, 117], where the $\{a_{\ell k}\}$ represent the transition probabilities from states ℓ to k and the $\{p_\ell\}$ denote the steady-state distribution for the Markov chain.

We provide here an interpretation for (2.13) in the context of multi-agent networks by considering two generic agents, k and ℓ, from an arbitrary network, as shown in Fig. 2.1. The coefficient $a_{\ell k}$ is used by agent k to scale information arriving from agent ℓ. Therefore, this coefficient reflects the amount of confidence that agent k has in the information arriving from agent ℓ. Likewise, for $a_{k\ell}$. Since the combination policy is not necessarily symmetric, it will hold in general that $a_{\ell k} \neq a_{k\ell}$. However, agent k can re-scale the incoming weight $a_{\ell k}$ by p_k, and likewise for agent ℓ, so that the local balance condition (2.13) requires each pair of rescaled weights to match each other. We can interpret $a_{\ell k}$ to represent the (fractional) amount of information flowing from ℓ to k and p_k to represent the price paid by agent k for that information. Expression (2.13) is then requiring the information-cost benefit to be equitable across agents.

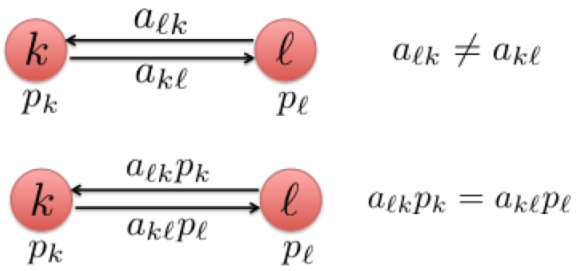

Figure 2.1: Illustration of the local balance condition (2.13).

It is worth noting that the local balancing condition (2.13) is satisfied by several important left-stochastic policies, as illustrated in four examples below. Thus, let $\tau_k = \mu_k/\mu_{\max}$ for agent k. Then condition (2.9) becomes

$$q = \beta\mu_{\max}\,\mathrm{diag}\{\tau_1, \tau_2, \cdots, \tau_K\}p, \tag{2.15}$$

where $\tau_k \in (0, 1]$.

Policy 1 (Hastings rule) The first policy we consider is the Hastings rule. Given $\{q_k\}_{k=1}^N$ and $\{\mu_k\}_{k=1}^N$, we select $a_{\ell k}$ as [1, 118]:

$$a_{\ell k} = \begin{cases} \dfrac{\mu_k/q_k}{\max\{n_k\mu_k/q_k,\, n_\ell\mu_\ell/q_\ell\}}, & \text{if } \ell \in \mathcal{N}_k\backslash\{k\}, \\ 1 - \displaystyle\sum_{m\in\mathcal{N}_k\backslash\{k\}} a_{mk}, & \text{if } \ell = k, \\ 0, & \text{if } \ell \notin \mathcal{N}_k. \end{cases} \tag{2.16}$$

where $n_k \triangleq |\mathcal{N}_k|$ (the number of neighbors of agent k). It can be verified that A is left-stochastic, and that the entries of its Perron eigenvector p are given by

$$p_k \triangleq \frac{q_k/\mu_k}{\sum_{\ell=1}^K q_\ell/\mu_\ell} > 0. \tag{2.17}$$

Let

$$\beta = \sum_{\ell=1}^K q_\ell/\mu_\ell = \frac{1}{\mu_{\max}}\sum_{\ell=1}^K q_\ell/\tau_\ell > 0. \tag{2.18}$$

With (2.16) and (2.17), it is easy to verify that

$$a_{\ell k}p_k = \frac{1}{\beta\max\{n_k\mu_k/q_k,\, n_\ell\mu_\ell/q_\ell\}} = a_{k\ell}p_\ell. \tag{2.19}$$

If $\ell = k$, it is obvious that (2.13) holds. If $\ell \notin \mathcal{N}_k$, then $k \notin \mathcal{N}_\ell$. In this case, $a_{\ell k} p_k = a_{k\ell} p_\ell = 0$.

Furthermore, we can also verify that when $\{q_k\}_{k=1}^N$ and $\{\mu_k\}_{k=1}^N$ are given, $\{a_{\ell k}\}$ are generated through (2.16), and β is chosen as in (2.18), then condition (2.9) is satisfied.

■

Policy 2 (Averaging rule) The second policy we consider is the popular average combination rule where $a_{\ell k}$ is chosen as

$$a_{\ell k} = \begin{cases} 1/n_k, & \text{if } \ell \in \mathcal{N}_k, \\ 0, & \text{otherwise.} \end{cases} \tag{2.20}$$

The entries of the Perron eigenvector p are given by

$$p_k = n_k \left(\sum_{m=1}^{K} n_m \right)^{-1}. \tag{2.21}$$

With (2.20) and (2.21), it clearly holds that

$$a_{\ell k} p_k = \left(\sum_{m=1}^{K} n_m \right)^{-1} = a_{k\ell} p_\ell, \tag{2.22}$$

which implies (2.13).

We can further verify that when μ_k is set as

$$\mu_k = \frac{q_k}{n_k} \mu_o, \quad \forall\, k = 1, 2, \cdots, N \tag{2.23}$$

for some positive constant step-size μ_o and β is set as

$$\beta = \left(\sum_{m=1}^{K} n_m \right) \Big/ \mu_o > 0, \tag{2.24}$$

then condition (2.9) will hold. ■

Policy 3 (Relative-degree rule) The third policy we consider is the relative-degree combination rule [119] where $a_{\ell k}$ is chosen as

$$a_{\ell k} = \begin{cases} n_\ell \left(\sum_{m \in \mathcal{N}_k} n_m \right)^{-1}, & \text{if } \ell \in \mathcal{N}_k, \\ 0, & \text{otherwise,} \end{cases} \tag{2.25}$$

and the entries of the Perron eigenvector p are given by

$$p_k = \frac{n_k \sum_{m\in\mathcal{N}_k} n_m}{\sum_{k=1}^{K} \left(n_k \sum_{m\in\mathcal{N}_k} n_m\right)}. \tag{2.26}$$

With (2.25) and (2.26), it clearly holds that

$$a_{\ell k} p_k = \frac{n_k n_\ell}{\sum_{k=1}^{K} \left(n_k \sum_{m\in\mathcal{N}_k} n_m\right)} = a_{k\ell} p_\ell, \tag{2.27}$$

which implies (2.13).

We can further verify that when μ_k is set as

$$\mu_k = \frac{q_k}{n_k \sum_{m\in\mathcal{N}_k} n_m}\mu_o, \quad \forall\, k = 1, 2, \cdots, K, \tag{2.28}$$

and β is set as

$$\beta = \sum_{k=1}^{K} \left(n_k \sum_{m\in\mathcal{N}_k} n_m\right) \Big/ \mu_o, \tag{2.29}$$

then condition (2.9) will hold. ■

Policy 4 (Doubly stochastic policy) If matrix A is primitive, symmetric, and doubly stochastic, its Perron eigenvector is $p = \frac{1}{K}\mathbb{1}_K$. In this situation, the local balance condition (2.13) holds automatically.

Furthermore, if we assume each agent employs the step-size $\mu_k = q_k K \mu_o$ for some positive constant step-size μ_o, it can be verified that condition (2.9) holds with

$$\beta = 1/\mu_o. \tag{2.30}$$

There are various rules to generate a primitive, symmetric and doubly stochastic matrix. Some common rules are the Laplacian rule, maximum-degree rule, Metropolis rule and other rules that listed in Table 14.1 in [1]. ■

Policy 5 (Other locally-balanced policies) For other left-stochastic-policies for which closed-form expressions for the Perron entries need not be available, the Perron eigenvector p can be learned iteratively to ensure that the step-sizes μ_k end up satisfying (2.12). Before

we explain how this can be done, we remark that since the combination matrix A is left-stochastic in our formulation, the power iteration employed in push-sum implementations cannot be applied since it works for right-stochastic policies. We proceed instead as follows.

Since A is primitive and left-stochastic, it is shown in [1, 120] that

$$\lim_{i\to\infty} A^i = p\mathbb{1}_K^{\mathsf{T}}. \tag{2.31}$$

This relation also implies

$$\lim_{i\to\infty} (A^{\mathsf{T}})^i = \mathbb{1}_K p^{\mathsf{T}}. \tag{2.32}$$

Now let e_k be the k-th column of the identity matrix $I_K \in \mathbb{R}^{N\times K}$. Furthermore, let each agent k keep an auxiliary variable $z_{k,i} \in \mathbb{R}^N$ with each $z_{k,-1}$ initialized to e_k. We also introduce

$$\mathcal{z}_i \triangleq \mathrm{col}\{z_{1,i}, z_{2,i}, \cdots, z_{N,i}\} \in \mathbb{R}^{N^2}, \tag{2.33}$$

$$\mathcal{A} \triangleq A \otimes I_K. \tag{2.34}$$

By iterating $\mathcal{z}_i$ according to

$$\mathcal{z}_{i+1} = \mathcal{A}^{\mathsf{T}} \mathcal{z}_i, \tag{2.35}$$

we have

$$\begin{aligned}
\lim_{i\to\infty} \mathcal{z}_i &= \lim_{i\to\infty} (\mathcal{A}^{\mathsf{T}})^{i+1} \mathcal{z}_{-1} \\
&= \lim_{i\to\infty} [(A^{\mathsf{T}})^{i+1} \otimes I_K] \mathcal{z}_{-1} \overset{(2.32)}{=} (\mathbb{1}_K p^{\mathsf{T}} \otimes I_K) \mathcal{z}_{-1} \\
&= [(\mathbb{1}_K \otimes I_K)(p^{\mathsf{T}} \otimes I_K)] \mathcal{z}_{-1}.
\end{aligned} \tag{2.36}$$

Since $\mathcal{z}_{-1} = \mathrm{col}\{e_1, \cdots .e_K\}$, it can be verified that $(p^{\mathsf{T}} \otimes I_K)\mathcal{z}_{-1} = p$. Substituting into (2.36), we have $\lim_{i\to\infty} z_{k,i} = p$. In summary, it holds that

$$\lim_{i\to\infty} z_{k,i}(k) = p_k \tag{2.37}$$

where $z_{k,i}(k)$ is the k-th entry of the vector $z_{k,i}$. Therefore, if we set

$$\mu_{k,i} = \frac{q_k \mu_o}{z_{k,i}(k)}, \tag{2.38}$$

then it follows that

$$\lim_{i\to\infty} \mu_{k,i} = q_k \mu_o / p_k. \tag{2.39}$$

Finally, to guarantee $z_{k,i}(k) > 0$ for $i = 0, 1, 2, \cdots$, it is enough to assume $a_{kk} > 0$ for each agent $k = 1, 2, \cdots, N$, i.e., each agent has to assign positive weight to itself.

■

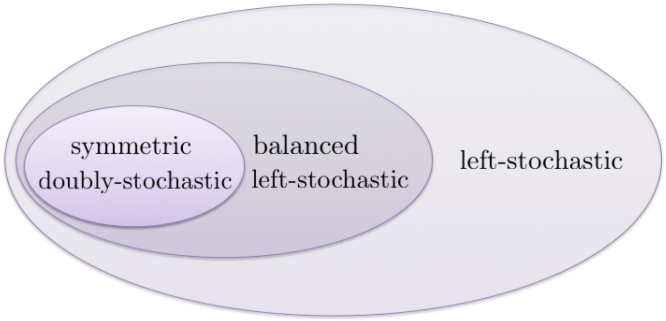

Figure 2.2: Illustration of the relations among the classes of symmetric doubly-stochastic, balanced left-stochastic, and left-stochastic combination matrices.

We illustrate in Fig. 2.2 the relations among the classes of symmetric doubly-stochastic, balanced left-stochastic, and left-stochastic combination matrices. It is seen that every symmetric doubly-stochastic matrix is both left-stochastic and balanced. We indicated earlier that the EXTRA algorithm was derived in [75] with exact convergence properties for symmetric doubly-stochastic matrices. Here, in the sequel, we shall derive an exact diffusion strategy with exact convergence guarantees for the larger class of balanced left-stochastic matrices (which is therefore also applicable to symmetric doubly-stochastic matrices). We will show in Chapter 3 that the exact diffusion implementation has a wider stability range than EXTRA consensus; this is a useful property since larger step-sizes can be used to attain larger convergence rates.

2.2.3 Values of Balanced Left-stochastic Policies

For undirected networks, though it is quite common to employ symmetric and doubly-stochastic combination policies such as in [5, 74, 75, 84, 85, 88, 95, 96], balanced left-stochastic

policies can still be of great significant value. Some of the key benefits of these policies are as follows.

First, balanced left-stochastic policies can speed up convergence. For example, in highly unbalanced networks (e.g., the coauthorship network) where the degrees of neighboring nodes differ drastically, the averaging rule enables faster convergence than doubly-stochastic policies (see the discussions in Sec. 2.6.3). The second scenario where balanced left-stochastic policies help is when the Lipschitz constant associated with each local cost function differs drastically — the Lipschitz constants δ in some nodes are much larger than the other nodes. Note that δ_ℓ can be regarded as an importance measure of node ℓ, and it is helpful for agent k to assign more (less) weights to neighboring node ℓ if δ_ℓ is large (small). One such weighting policy is the Hastings rule

$$a_{\ell k} = \begin{cases} \dfrac{1/\delta_k}{\max\{n_k/\delta_k, n_\ell/\delta_\ell\}}, & \text{if } \ell \in \mathcal{N}_k \backslash \{k\}, \\ 1 - \displaystyle\sum_{m \in \mathcal{N}_k \backslash \{k\}} a_{mk}, & \text{if } \ell = k, \\ 0, & \text{if } \ell \notin \mathcal{N}_k. \end{cases} \tag{2.40}$$

which is balanced left-stochastic. The Hastings rule (2.40) performs similar to importance sampling in the machine learning literature [73, 121, 122] where data samples with larger magnitude are assigned larger sampling probability. We illustrate the benefit of the Hastings rule (2.40) in Sec. 2.6.3.

Second, balanced left-stochastic policies enable more flexible step-size choices — each agent k can choose a different local step-size μ_k. For example, suppose each agent k sets a proper local step-size μ_k, the exact convergence can be guaranteed if the combination policy is generated according to the Hastings rule (2.16), see the explanation in Policy 1. In contrast, EXTRA with a doubly-stochastic matrix has to enforce that all agents choose the same step-size μ. Note that such flexible step-size choices have many benefits. It avoids the communication costs to coordinate step-sizes. Moreover, each agent can choose step-sizes purely according to its own local cost functions. If the condition number of $J_k(w)$ is small (or large), agent k can set a relatively large (or small) step-size, which can speed up the

converge of the algorithm.

Third, balanced left-stochastic policies can have better privacy-preserving properties than doubly-stochastic policies. For example, the averaging rule (2.20) can be constructed from the agent's own degree, and no neighbors' degree is required. In contrast, the doubly-stochastic matrices generated by the maximum-degree rule or Metropolis rule [1] will require agents to share their *degrees* with neighbors.

Fourth, it is shown in Chapters 12 and 15 of [1] that the Hastings rule and the relative-degree rule (see (2.25)) achieve better mean-square-error (MSE) performance over adaptive networks than doubly-stochastic policies.

2.2.4 Necessity of Locally Balanced Condition

One may wonder whether exact convergence can be guaranteed for general left-stochastic matrices that are not necessarily balanced (i.e., whether the convergence property can be extended beyond the middle elliptical area in Fig. 2.2). It turns out that one can provide examples of combination matrices that are left-stochastic (but not necessarily balanced) for which exact convergence occurs and others for which exact convergence does not occur (see, e.g., the examples in Section 2.5 and Figs. 2.9 and 2.10). In other words, exact convergence is not always guaranteed beyond the balanced class. This conclusion is another useful contribution of this work; it shows that there is a boundary inside the set of left-stochastic matrices within which convergence can be always guaranteed (namely, the set of balanced matrices).

It is worth noting that the recent works [100, 123] extend the EXTRA method to the case of directed networks by employing a push-sum technique. These extensions do not require the local balancing condition but they establish convergence only if the step-size parameter falls within an interval $(c_{\text{lower}}, c_{\text{upper}})$ where c_{lower} and c_{upper} are two positive constants. However, it is not proved in these works whether this interval is feasible, i.e., whether $c_{\text{upper}} > c_{\text{lower}}$. In fact, we will construct examples in Section 2.5 for which both exact diffusion and push-sum EXTRA will diverge for any step-size μ. In other words, both exact diffusion and EXTRA

methods need not work well for directed networks. This is a disadvantage in comparison with DIGing-based methods [92–96].

In summary, when locally-balanced policies is employed, exact diffusion is more communication efficient and also more stable than other techniques including DIGing methods (because the communicated variables required in each iteration of DIGing is twice as much as that in exact diffusion) and EXTRA. However, just like EXTRA, the exact diffusion strategy is applicable to *undirected* (rather than *directed*) graphs.

2.2.5 Useful Properties

We now establish several useful properties for primitive left-stochastic matrices that satisfy the local balance condition (2.13). These properties will be used in the sequel.

Lemma 2.1 (Properties of $AP - P + I_K$) *When A satisfies the local balance condition* (2.13)*, it holds that the matrix $AP - P + I_K$ is primitive, symmetric, and doubly stochastic.*

Proof: With condition (2.13), the symmetry of $AP - P + I_K$ is obvious. To check the primitiveness of $AP - P + I_K$, we need to verify two facts, namely, that: (a) at least one diagonal entry in $AP - P + I_K$ is positive, and (b) there exists at least one path with nonzero weights between any two agents. It is easy to verify condition (a) because A is already primitive and $P < I_K$. For condition (b), since A is connected and all diagonal entries of P are positive, then if there exists a path with nonzero coefficients linking agents k and ℓ under A, the same path will continue to exist under AP. Moreover, since all diagonal entries of $-P + I_K$ are positive, then the same path will also exist under $AP - P + I_K$. Finally, $AP - P + I_K$ is doubly stochastic because

$$\mathbb{1}_K^{\mathsf{T}} (AP - P + I_K) = p^{\mathsf{T}} - p^{\mathsf{T}} + \mathbb{1}_K^{\mathsf{T}} = \mathbb{1}_K^{\mathsf{T}}, \tag{2.41}$$

$$(AP - P + I_K)\, \mathbb{1}_K = p - p + \mathbb{1}_K = \mathbb{1}_K. \tag{2.42}$$

■

Lemma 2.2 (Nullspace of $P - AP$) *When A satisfies the local balance condition* (2.13)*, it holds that $P - AP$ is symmetric and positive semi-definite. Moreover, it holds that*

$$\text{null}(P - AP) = \text{span}\{\mathbb{1}_K\}, \tag{2.43}$$

where null$(\cdot)$ *denotes the null space of its matrix argument.*

Proof: Let λ_k denote the k-th largest eigenvalue of $AP - P + I_K$. Recall from Lemma 2.1 that $AP - P + I_K$ is primitive and doubly stochastic. Therefore, according to Lemma F.4 from [1] it holds that

$$1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_K > -1, \tag{2.44}$$

It follows that the eigenvalues of $AP - P$ are non-positive so that $P - AP \geq 0$.

Note further from (2.44) that the matrix $AP - P + I_K$ has a single eigenvalue at one with multiplicity one. Moreover, from (2.42) we know that the vector $\mathbb{1}_K$ is a right-eigenvector associated with this eigenvalue at one. Based on these two facts, we have

$$(AP - P + I_K)\, x = x \Longleftrightarrow x = c\mathbb{1}_K \tag{2.45}$$

for any constant c. Relation (2.45) is equivalent to

$$(AP - P)\, x = 0 \Longleftrightarrow x = c\mathbb{1}_K, \tag{2.46}$$

which confirms (2.43). ■

Corollary 2.1 (Nullspace of $\mathcal{P} - \mathcal{A}\mathcal{P}$) *Let $\mathcal{P} \triangleq P \otimes I_M$ and $\mathcal{A} \triangleq A \otimes I_M$. When A satisfies the local balance condition* (2.13)*, it holds that*

$$\begin{aligned} \text{null}(\mathcal{P} - \mathcal{A}\mathcal{P}) &= \text{null}\Big((P - AP) \otimes I_M\Big) \\ &= \text{span}\{\mathbb{1}_K \otimes I_M\}. \end{aligned} \tag{2.47}$$

Moreover, for any block vector $x = \text{col}\{x_1, x_2, \cdots, x_K\} \in \mathbb{R}^{MN}$ in the nullspace of $\mathcal{P} - \mathcal{A}\mathcal{P}$ with entries $x_k \in \mathbb{R}^M$, it holds that

$$(\mathcal{P} - \mathcal{A}\mathcal{P})x = 0 \Longleftrightarrow x_1 = x_2 = \cdots = x_K. \tag{2.48}$$

Proof: Since $P - AP + I_K$ has a single eigenvalue at 1 with multiplicity one, we conclude that $(P - AP + I_K) \otimes I_M$ will have an eigenvalue at 1 with multiplicity M. Next we denote the columns of the identity matrix by $I_M = [e_1, e_2, \cdots, e_K]$ where $e_k \in \mathbb{R}^M$. We can verify that $\mathbb{1}_K \otimes e_k$ is a right-eigenvector associated with the eigenvalue 1 because

$$\begin{aligned} &[(P - AP + I_K) \otimes I_M][\mathbb{1}_K \otimes e_k] \\ &\quad = [(P - AP + I_K)\mathbb{1}_K] \otimes e_k = \mathbb{1}_K \otimes e_k. \end{aligned} \tag{2.49}$$

Now since any two vectors in the set $\{\mathbb{1}_K \otimes e_k\}_{k=1}^M$ are mutually independent, we conclude that

$$\begin{aligned} (\mathcal{P} - \mathcal{A}\mathcal{P})x = 0 &\Longleftrightarrow (\mathcal{P} - \mathcal{A}\mathcal{P} + I_{MN})x = x \\ &\Longleftrightarrow x \in \text{span}\{[\mathbb{1}_K \otimes e_1, \cdots, \mathbb{1}_K \otimes e_M]\} \\ &\Longleftrightarrow x \in \text{span}\{\mathbb{1}_K \otimes I_M\}. \end{aligned} \tag{2.50}$$

These equalities establish (2.47). From (2.47) we can also conclude (2.48) because

$$\begin{aligned} &x \in \text{span}\{\mathbb{1}_K \otimes I_M\} \\ &\quad \Rightarrow x = (\mathbb{1}_K \otimes I_M) \cdot x = \text{col}\{x, x, \cdots, x\} \end{aligned} \tag{2.51}$$

from some $x \in \mathbb{R}^M$. The direction "$\Leftarrow$" of (2.48) is obvious. ■

Lemma 2.3 (Real eigenvalues) *When A satisfies the local balance condition* (2.13)*, it holds that A is diagonalizable with real eigenvalues in the interval* $(-1, 1]$*, i.e.,*

$$A = Y \Lambda Y^{-1}, \tag{2.52}$$

where $\Lambda = diag\{\lambda_1(A), \cdots, \lambda_K(A)\} \in \mathbb{R}^{K \times K}$*, and*

$$1 = \lambda_1(A) > \lambda_2(A) \geq \lambda_3(A) \geq \cdots \geq \lambda_K(A) > -1. \tag{2.53}$$

Proof: According to the local balance condition (2.14), PA^{T} is symmetric. Using the fact that $P > 0$ is diagonal, it holds that

$$P^{-\frac{1}{2}} A P^{\frac{1}{2}} = P^{-\frac{1}{2}} (AP) P^{-\frac{1}{2}}, \tag{2.54}$$

which shows that the matrix on the left-hand side is symmetric. Therefore, $P^{-\frac{1}{2}}AP^{\frac{1}{2}}$ can be decomposed as

$$P^{-\frac{1}{2}}AP^{\frac{1}{2}} = Y_1 \Lambda Y_1^{\mathsf{T}}, \tag{2.55}$$

where Y_1 is an orthogonal matrix and Λ is a real diagonal matrix. From (2.55), we further have that

$$A = P^{\frac{1}{2}} Y_1 \Lambda Y_1^{\mathsf{T}} P^{-\frac{1}{2}}. \tag{2.56}$$

If we let $Y = P^{\frac{1}{2}} Y_1$, we reach the decomposition (2.52). Moreover, since A is a primitive left-stochastic matrix, according to Lemma F.4 in [1], the eigenvalues of A satisfy (2.53). ■

For ease of reference, we collect in Table 2.1 the properties established in Lemmas 2.1 through 2.3 for balanced primitive left-stochastic matrices A.

Properties of balanced primitive left-stochastic matrices A
A is diagonalizable with real eigenvalues in $(-1, 1]$;
A has a single eigenvalue at 1;
$AP - P + I_K$ is symmetric, primitive, doubly-stochastic;
$P - AP$ is positive semi-definite;
$\text{null}(P - AP) = \text{span}(\mathbb{1}_K)$;
$\text{null}(\mathcal{P} - \mathcal{A}\mathcal{P}) = \text{span}\{\mathbb{1}_K \otimes I_M\}$.

Table 2.1: Properties of balanced primitive left-stochastic matrices A

2.3 Penalized Formulation of Diffusion

In this section, we employ the properties derived in the previous section to reformulate the unconstrained optimization problem (2.1) into the equivalent constrained problem (2.69),

which will be solved using a penalized formulation. This derivation will help clarify the origin of the $O(\mu_{\max}^2)$ bias from (2.10) in the standard diffusion implementation.

2.3.1 Constrained Problem Formulation

To begin with, note that the unconstrained problem (2.1) is equivalent to the following constrained problem:

$$\begin{aligned} \min_{\{w_k\}} \quad & \sum_{k=1}^{K} q_k J_k(w_k), \\ \text{s.t.} \quad & w_1 = w_2 = \cdots = w_K. \end{aligned} \tag{2.57}$$

Now we introduce the block vector $\mathcal{w} \triangleq \text{col}\{w_1, \cdots, w_K\} \in \mathbb{R}^{KM}$ and

$$\mathcal{J}^\star(\mathcal{w}) \triangleq \sum_{k=1}^{K} q_k J_k(w_k), \tag{2.58}$$

With (2.48) and (2.58), problem (2.57) is equivalent to

$$\min_{\mathcal{W} \in \mathbb{R}^{NM}} \quad \mathcal{J}^\star(\mathcal{w}), \quad \text{s.t.} \quad \frac{1}{2}\left(\mathcal{P} - \mathcal{AP}\right)\mathcal{w} = 0. \tag{2.59}$$

From Lemma 2.2, we know that $P - AP$ is symmetric and positive semi-definite. Therefore, we can decompose

$$\frac{P - AP}{2} = U \Sigma U^{\mathsf{T}}, \tag{2.60}$$

where $\Sigma \in \mathbb{R}^{K \times K}$ is a non-negative diagonal matrix and $U \in \mathbb{R}^{K \times K}$ is an orthogonal matrix. If we introduce the symmetric square-root matrix

$$V \triangleq U \Sigma^{1/2} U^{\mathsf{T}} \in \mathbb{R}^{K \times K}, \tag{2.61}$$

then it holds that

$$\frac{P - AP}{2} = V^2. \tag{2.62}$$

Let $\mathcal{V} \triangleq V \otimes I_M$ so that

$$\frac{\mathcal{P} - \mathcal{AP}}{2} = \mathcal{V}^2. \tag{2.63}$$

Lemma 2.4 (Nullspace of V) *With V defined as in (2.61), it holds that*

$$\text{null}(V) = \text{null}(P - AP) = \text{span}\{\mathbb{1}_K\}. \tag{2.64}$$

Proof: To prove $\text{null}(V) = \text{null}(P - AP)$, it is enough to prove

$$(P - AP)x = 0 \Longleftrightarrow Vx = 0. \tag{2.65}$$

Indeed, notice that

$$\begin{aligned}(P - AP)x = 0 \Rightarrow V^2 x = 0 \Rightarrow x^{\mathsf{T}} V^{\mathsf{T}} V x = 0 \\ \Rightarrow \|Vx\|^2 = 0 \Rightarrow Vx = 0.\end{aligned} \tag{2.66}$$

The reverse direction "$\Leftarrow$" in (2.65) is obvious. ■

Remark 2.3 (Nullspace of $\mathcal{V}$) *Similar to the arguments in (2.47) and (2.48), we have*

$$\text{null}(\mathcal{V}) = \text{null}(\mathcal{P} - \mathcal{A}\mathcal{P}) = \text{span}\{\mathbb{1}_K \otimes I_M\}, \tag{2.67}$$

and, hence,

$$\mathcal{V}x = 0 \Longleftrightarrow (\mathcal{P} - \mathcal{A}\mathcal{P})x = 0 \Longleftrightarrow x_1 = \cdots = x_K. \tag{2.68}$$

■

With (2.68), problem (2.59) is equivalent to

$$\min_{\mathcal{W} \in \mathbb{R}^{NM}} \quad \mathcal{J}^\star(\mathcal{W}), \quad \text{s.t.} \quad \mathcal{V}\mathcal{W} = 0. \tag{2.69}$$

In this way, we have transformed the original problem (2.1) to the equivalent constrained problem (2.69).

2.3.2 Penalized Formulation

There are many techniques to solve constrained problems of the form (2.69). One useful and popular technique is to add a penalty term to the cost function and to consider instead a penalized problem of the form:

$$\min_{\mathcal{W} \in \mathbb{R}^{NM}} \quad \mathcal{J}^\star(\mathcal{W}) + \frac{1}{\alpha} \|\mathcal{V}\mathcal{W}\|^2, \tag{2.70}$$

where $\alpha > 0$ is a penalty parameter. Problem (2.70) is not equivalent to (2.69) but is a useful approximation. The smaller the value of α is, the closer the solutions of problems (2.69) and (2.70) become to each other [124–126]. We now verify that the diffusion strategy (2.4)–(2.5) follows from applying an incremental technique to solving the approximate penalized problem (2.70), not the real problem (2.69). It will then become clear that the diffusion estimate $w_{k,i}$ cannot converge to the exact solution $w^\star$ of problem (2.1) (or (2.69)).

Since (2.63) holds, problem (2.70) is equivalent to

$$\min_{w\in\mathbb{R}^{NM}} \quad \mathcal{J}^\star(\mathcal{w}) + \frac{1}{2\alpha}\mathcal{w}^{\mathsf{T}}(\mathcal{P}-\mathcal{A}\mathcal{P})\mathcal{w}. \tag{2.71}$$

This is an unconstrained problem, which we can solve using, for example, a diagonally-weighted incremental algorithm, namely,

$$\begin{cases} \psi_i = \mathcal{w}_{i-1} - \alpha\mathcal{P}^{-1}\nabla\mathcal{J}^\star(\mathcal{w}_{i-1}), \\ \mathcal{w}_i = \psi_i - \alpha\mathcal{P}^{-1}\Big(\dfrac{1}{\alpha}(\mathcal{P}-\mathcal{A}\mathcal{P})\psi_i\Big), \end{cases} \tag{2.72}$$

The above recursion can be simplified as follows. Assume we select

$$\alpha \;\triangleq\; \beta^{-1}, \tag{2.73}$$

where β is the same constant used in relation (2.9). Recall from (2.18), (2.24), (2.29) and (2.30) that $\beta = O(1/\mu_{\max})$ and hence $\alpha = O(\mu_{\max})$. Moreover, from the definition of $\mathcal{J}^\star(\mathcal{w})$ in (2.58), we have

$$\nabla\mathcal{J}^\star(\mathcal{w}) = \begin{bmatrix} q_1\nabla J_1(w_1) \\ \vdots \\ q_K\nabla J_K(w_K) \end{bmatrix} \tag{2.74}$$

Using (2.9), namely,

$$q_k = \beta\mu_k p_k, \tag{2.75}$$

we find that

$$\alpha\mathcal{P}^{-1}\nabla\mathcal{J}^\star(\mathcal{w}_{i-1}) = \begin{bmatrix} \mu_1\nabla J_1(w_{1,i-1}) \\ \vdots \\ \mu_K\nabla J_K(w_{K,i-1}) \end{bmatrix}. \tag{2.76}$$

We further introduce the aggregate cost (which is similar to (2.58) but without the weighting coefficients):

$$\mathcal{J}^o(\mathcal{w}) \triangleq \sum_{k=1}^{K} J_k(w_k), \tag{2.77}$$

and note that

$$\nabla\mathcal{J}^o(\mathcal{w}) = \begin{bmatrix} \nabla J_1(w_1) \\ \vdots \\ \nabla J_K(w_K) \end{bmatrix}. \tag{2.78}$$

Let $\mathcal{M} \triangleq \text{diag}\{\mu_1, \mu_2, \cdots, \mu_K\} \otimes I_M$. Using (2.76) and (2.78), the first recursion in (2.72) can be rewritten as

$$\psi_i = \mathcal{w}_{i-1} - \mathcal{M}\nabla\mathcal{J}^o(\mathcal{w}_{i-1}). \tag{2.79}$$

For the second recursion of (2.72), it can be rewritten as

$$\mathcal{w}_i = \mathcal{A}^{\mathsf{T}}\psi_i \tag{2.80}$$

because $\mathcal{A}\mathcal{P} = \mathcal{P}\mathcal{A}^{\mathsf{T}}$. Relations (2.79)–(2.80) are equivalent to (2.4)–(2.5). Specifically, if we collect all iterates from across all agents into block vectors $\{\mathcal{w}_i, \psi_i\}$, then (2.4)–(2.5) would lead to (2.79)–(2.80). From this derivation, we conclude that the diffusion algorithm (2.4)–(2.5) can be interpreted as performing the diagonally-weighted incremental construction (2.72) to solve the approximate penalized problem (2.71). Since this construction is not solving the real problem (2.1), there exists a bias between its fixed point and the real solution $w^\star$. As shown in (2.10), the size of this bias is related to $\mu_{\max}$. When $\mu_{\max}$ is small, the bias is also small. This same conclusion can be seen by noting that a small $\mu_{\max}$ corresponds to a large penalty factor $1/\alpha$ under which the solutions to problems (2.1) and (2.69) approach each other.

2.4 Development of Exact Diffusion

We now explain how to adjust the diffusion strategy (2.4)–(2.5) to ensure exact convergence to $w^\star$. Instead of solving the approximate penalized problem (2.71), we apply the primal-dual

saddle point method to solve the original problem (2.69) directly. We continue to assume that the combination policy A is primitive and satisfies the local balancing condition (2.13).

To solve (2.69) with saddle point algorithm, we first introduce the augmented Lagrangian function:

$$\begin{aligned}\mathcal{L}_a(\mathcal{W},\mathcal{Y}) &= \mathcal{J}^\star(\mathcal{W}) + \frac{1}{\alpha}\mathcal{Y}^{\mathsf{T}}\mathcal{V}\mathcal{W} + \frac{1}{2\alpha}\left\|\mathcal{V}\mathcal{W}\right\|^2 \\ &\overset{(2.63)}{=} \mathcal{J}^\star(\mathcal{W}) + \frac{1}{\alpha}\mathcal{Y}^{\mathsf{T}}\mathcal{V}\mathcal{W} + \frac{1}{4\alpha}\mathcal{W}^{\mathsf{T}}(\mathcal{P}-\mathcal{P}\mathcal{A}^{\mathsf{T}})\mathcal{W},\end{aligned} \tag{2.81}$$

where $\mathcal{Y} = \mathrm{col}\{y_1,\cdots,y_K\} \in \mathbb{R}^{NM}$ is the dual variable. The standard primal-dual saddle point algorithm has recursions

$$\begin{cases} \mathcal{W}_i = \mathcal{W}_{i-1} - \alpha\nabla_{\mathcal{W}}\mathcal{L}_a(\mathcal{W}_{i-1},\mathcal{Y}_{i-1}), \\ \mathcal{Y}_i = \mathcal{Y}_{i-1} + \alpha\left(\dfrac{1}{\alpha}\mathcal{V}\mathcal{W}_i\right) = \mathcal{Y}_{i-1} + \mathcal{V}\mathcal{W}_i. \end{cases} \tag{2.82}$$

The first recursion in (2.82) is the primal descent while the second recursion is the dual ascent. Now, instead of performing the descent step directly as shown in the first recursion in (2.82), we perform it in an incremental manner. Thus, let

$$\mathcal{D}(\mathcal{W}) \triangleq \frac{1}{4\alpha}\mathcal{W}^{\mathsf{T}}(\mathcal{P}-\mathcal{P}\mathcal{A}^{\mathsf{T}})\mathcal{W},\ \mathcal{C}(\mathcal{W},\mathcal{Y}) \triangleq \frac{1}{\alpha}\mathcal{Y}^{\mathsf{T}}\mathcal{V}\mathcal{W}, \tag{2.83}$$

so that

$$\mathcal{L}_a(\mathcal{W},\mathcal{Y}_{i-1}) = \mathcal{J}^\star(\mathcal{W}) + \mathcal{D}(\mathcal{W}) + \mathcal{C}(\mathcal{W},\mathcal{Y}_{i-1}). \tag{2.84}$$

The diagonally incremental recursion that corresponds to the first step in (2.82) is then:

$$\begin{cases} \theta_i = \mathcal{W}_{i-1} - \alpha\mathcal{P}^{-1}\nabla\mathcal{J}^\star(\mathcal{W}_{i-1}), \\ \phi_i = \theta_i - \alpha\mathcal{P}^{-1}\nabla\mathcal{D}(\theta_i) = \dfrac{I_{MN}+\mathcal{A}^{\mathsf{T}}}{2}\,\theta_i = \overline{\mathcal{A}}^{\mathsf{T}}\theta_i, \\ \mathcal{W}_i = \phi_i - \alpha\mathcal{P}^{-1}\nabla_{\mathcal{W}}\mathcal{C}(\phi_i,\mathcal{Y}_{i-1}) = \phi_i - \mathcal{P}^{-1}\mathcal{V}\mathcal{Y}_{i-1}, \end{cases} \tag{2.85}$$

where in the second recursion of (2.85) we introduced

$$\overline{\mathcal{A}} \triangleq (I_{MK}+\mathcal{A})/2. \tag{2.86}$$

We know from (2.53) that the eigenvalues of $\overline{A}$ are positive and lie within the interval $(0,1]$. In (2.85), if we substitute the first and second recursions into the third one, and also recall (2.76) that $\alpha\mathcal{P}^{-1}\nabla\mathcal{J}^{\star}(\mathcal{W}_{i-1}) = \mathcal{M}\nabla\mathcal{J}^{o}(\mathcal{W}_{i-1})$, then we get

$$\mathcal{W}_i = \overline{\mathcal{A}}^{\mathsf{T}}\Big(\mathcal{W}_{i-1} - \mathcal{M}\nabla\mathcal{J}^{o}(\mathcal{W}_{i-1})\Big) - \mathcal{P}^{-1}\mathcal{V}\mathcal{Y}_{i-1}. \tag{2.87}$$

Replacing the first recursion in (2.82) with (2.87), the previous primal-dual saddle point recursion (2.82) becomes

$$\boxed{\begin{cases} \mathcal{W}_i = \overline{\mathcal{A}}^{\mathsf{T}}\Big(\mathcal{W}_{i-1} - \mathcal{M}\nabla\mathcal{J}^{o}(\mathcal{W}_{i-1})\Big) - \mathcal{P}^{-1}\mathcal{V}\mathcal{Y}_{i-1} \\ \mathcal{Y}_i = \mathcal{Y}_{i-1} + \mathcal{V}\mathcal{W}_i \end{cases}} \tag{2.88}$$

Recursion (2.88) is the primal-dual form of the exact diffusion recursion we are seeking. For the initialization, we set $y_{-1} = 0$ and w_{-1} to be any value, and hence for $i = 0$ we have

$$\begin{cases} \mathcal{W}_0 = \overline{\mathcal{A}}^{\mathsf{T}}\Big(\mathcal{W}_{-1} - \mathcal{M}\nabla\mathcal{J}^{o}(\mathcal{W}_{-1})\Big), \\ \mathcal{Y}_0 = \mathcal{V}\mathcal{W}_0. \end{cases} \tag{2.89}$$

We can rewrite (2.88) in a simpler form by eliminating the dual variable $\mathcal{Y}$ from the first recursion. For $i = 1, 2, \cdots$, from (2.88) we have

$$\mathcal{W}_i - \mathcal{W}_{i-1} = \overline{\mathcal{A}}^{\mathsf{T}}\Big(\mathcal{W}_{i-1} - \mathcal{W}_{i-2} - \mathcal{M}\big(\nabla\mathcal{J}^{o}(\mathcal{W}_{i-1}) - \nabla\mathcal{J}^{o}(\mathcal{W}_{i-2})\big)\Big) - \mathcal{P}^{-1}\mathcal{V}(\mathcal{Y}_{i-1} - \mathcal{Y}_{i-2}). \tag{2.90}$$

From the second step in (2.88) we have

$$\mathcal{P}^{-1}\mathcal{V}(\mathcal{Y}_{i-1} - \mathcal{Y}_{i-2}) = \mathcal{P}^{-1}\mathcal{V}^2\mathcal{W}_{i-1} \overset{(2.63)}{=} \mathcal{P}^{-1}\left(\frac{\mathcal{P} - \mathcal{P}\mathcal{A}^{\mathsf{T}}}{2}\right)\mathcal{W}_{i-1} = \left(\frac{I_{MK} - \mathcal{A}^{\mathsf{T}}}{2}\right)\mathcal{W}_{i-1}. \tag{2.91}$$

Substituting (2.91) into (2.90), we arrive at

$$\boxed{\mathcal{W}_i = \overline{\mathcal{A}}^{\mathsf{T}}\Big(2\mathcal{W}_{i-1} - \mathcal{W}_{i-2} - \mathcal{M}\big(\nabla\mathcal{J}^{o}(\mathcal{W}_{i-1}) - \nabla\mathcal{J}^{o}(\mathcal{W}_{i-2})\big)\Big)} \tag{2.92}$$

Recursion (2.92) is the primal version of the exact diffusion.

We can rewrite (2.92) in a distributed form that resembles (2.4)–(2.5) more closely, as listed below in Algorithm 1, where we denote the entries of $\overline{A}$ by $\overline{a}_{\ell k}$. It is observed in

Algorithm 2.1 Exact diffusion strategy for agent k

Setting: Let $\overline{A} = (I_K + A)/2$, and $w_{k,-1}$ arbitrary. Set $\psi_{k,-1} = w_{k,-1}$. Let $\mu_k = q_k\mu_o/p_k$

Repeat for $i = 0, 1, 2, \cdots$

$$\psi_{k,i} = w_{k,i-1} - \mu_k \nabla J_k(w_{k,i-1}), \qquad \text{(adaptation)} \tag{2.93}$$

$$\phi_{k,i} = \psi_{k,i} + w_{k,i-1} - \psi_{k,i-1}, \qquad \text{(correction)} \tag{2.94}$$

$$w_{k,i} = \sum_{\ell \in \mathcal{N}_k} \overline{a}_{\ell k} \phi_{\ell,i}. \qquad \text{(combination)} \tag{2.95}$$

Algorithm 1 that the exact diffusion strategy resembles (2.4)–(2.5) to great extent, with the addition of a "correction" step between the adaptation and combination step. In the correction step, the intermediate estimate $\psi_{k,i}$ is "corrected" by removing from it the difference between $w_{k,i-1}$ and $\psi_{k,i-1}$ from the previous iteration. Moreover, it is also observed that the exact and standard diffusion strategies have essentially the same computational complexity, apart from $2M$ (M is the dimension of $w_{k,i}$) additional additions per agent in the correction step of the exact implementation. Also, there is one combination step in each iteration, which reduces the communication cost by about one half in comparison to recent DIGing-based works [92–96].

One can directly run Algorithm 1 when the Perron entries $\{p_k\}$ are known beforehand, as explained in Section II-B. When this is not the case, we can blend iteration (2.35) into the algorithm and modify it as follows.

2.5 Significance of Balanced Policies

The stability and convergence properties of the exact diffusion strategy (2.93)–(2.95) will be examined in detail in Chapter 3. There we will show that exact diffusion is guaranteed to converge for all balanced left-stochastic matrices for sufficiently small step-sizes. The local balancing property turns out to be critical in the sense that convergence may or may not

Algorithm 2.2 Exact diffusion strategy when p is unknown

Setting: Let $\overline{A} = (I_K + A)/2$, and $w_{k,-1}$ arbitrary. Set $\psi_{k,-1} = w_{k,-1}$ and $z_{k,-1} = e_k$

Repeat for $i = 0, 1, 2, \cdots$

$$z_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} z_{\ell,i-1}, \qquad \text{(power iteration)} \tag{2.96}$$

$$\psi_{k,i} = w_{k,i-1} - \frac{q_k \mu_o}{z_{k,i}(k)} \nabla J_k(w_{k,i-1}), \qquad \text{(adaptation)} \tag{2.97}$$

$$\phi_{k,i} = \psi_{k,i} + w_{k,i-1} - \psi_{k,i-1}, \qquad \text{(correction)} \tag{2.98}$$

$$w_{k,i} = \sum_{\ell \in \mathcal{N}_k} \overline{a}_{\ell k} \phi_{\ell,i}. \qquad \text{(combination)} \tag{2.99}$$

occur if we move beyond the set of balanced policies. We can illustrate these possibilities here by means of examples. The two examples discussed in the sequel highlight the importance of having balanced combination policies for exact convergence.

Thus, consider the primal recursion of the exact diffusion algorithm (2.92), where $\overline{\mathcal{A}}$ is a general left-stochastic matrix. We subtract $\mathcal{w}^\star$ from both sides of (2.92), to get the error recursion

$$\widetilde{\mathcal{w}}_i = \overline{\mathcal{A}}^{\mathsf{T}} \left(2\widetilde{\mathcal{w}}_{i-1} - \widetilde{\mathcal{w}}_{i-2} + \mathcal{M}\big(\nabla \mathcal{J}^o(\mathcal{w}_{i-1}) - \nabla \mathcal{J}^o(\mathcal{w}_{i-2})\big)\right), \tag{2.100}$$

where $\widetilde{\mathcal{w}}_i = \mathcal{w}^\star - \mathcal{w}_i$. When $\nabla J_k(w)$ is twice-differentiable, we can appeal to the mean-value theorem from Lemma D.1 in [1], which allows us to express each difference

$$\nabla J_k(w_{k,i-1}) - \nabla J_k(w^\star) = -\left(\int_0^1 \nabla^2 J_k\big(w^\star - r\widetilde{w}_{k,i-1}\big) dr\right) \widetilde{w}_{k,i-1}. \tag{2.101}$$

If we let

$$H_{k,i-1} \triangleq \int_0^1 \nabla^2 J_k\big(w^\star - r\widetilde{w}_{k,i-1}\big) dr \in \mathbb{R}^{M \times M}, \tag{2.102}$$

and introduce the block diagonal matrix:

$$\mathcal{H}_{i-1} \triangleq \mathrm{diag}\{H_{1,i-1}, H_{2,i-1}, \cdots, H_{N,i-1}\}, \tag{2.103}$$

then we can rewrite

$$\nabla \mathcal{J}^o(\mathcal{w}_{i-1}) - \nabla \mathcal{J}^o(\mathcal{w}^\star) = -\mathcal{H}_{i-1}\widetilde{\mathcal{w}}_{i-1}. \tag{2.104}$$

Notice that

$$\begin{aligned}
&\nabla \mathcal{J}^o(\mathcal{w}_{i-1}) - \nabla \mathcal{J}^o(\mathcal{w}_{i-2}) \\
&\quad = \nabla \mathcal{J}^o(\mathcal{w}_{i-1}) - \nabla \mathcal{J}^o(\mathcal{w}^\star) + \nabla \mathcal{J}^o(\mathcal{w}^\star) - \nabla \mathcal{J}^o(\mathcal{w}_{i-2}) \\
&\overset{(2.104)}{=} \mathcal{H}_{i-2}\widetilde{\mathcal{w}}_{i-2} - \mathcal{H}_{i-1}\widetilde{\mathcal{w}}_{i-1}.
\end{aligned} \tag{2.105}$$

Combining (2.100), (2.105) and the fact $\widetilde{\mathcal{w}}_{i-1} = \widetilde{\mathcal{w}}_{i-1}$, we have

$$\begin{bmatrix} \widetilde{\mathcal{w}}_i \\ \widetilde{\mathcal{w}}_{i-1} \end{bmatrix} = (\mathcal{F} - \mathcal{G}_{i-1}) \begin{bmatrix} \widetilde{\mathcal{w}}_{i-1} \\ \widetilde{\mathcal{w}}_{i-2} \end{bmatrix}, \tag{2.106}$$

where

$$\mathcal{F} \triangleq \begin{bmatrix} 2\overline{\mathcal{A}}^{\mathsf{T}} & -\overline{\mathcal{A}}^{\mathsf{T}} \\ \mathcal{I}_{MN} & 0 \end{bmatrix} \in \mathbb{R}^{2MN \times 2MN}, \tag{2.107}$$

$$\mathcal{G}_{i-1} \triangleq \begin{bmatrix} \overline{\mathcal{A}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1} & -\overline{\mathcal{A}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-2} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{2MN \times 2MN}. \tag{2.108}$$

In the next two examples, we consider the simple case where the dimension $M = 1$, $q_k = 1$ for $k \in \{1, \cdots, K\}$, and the step-size $\mathcal{M} = \mu P^{-1}$, where

$$P = \mathrm{diag}\{p_1, \cdots, p_K\} \in \mathbb{R}^{K \times K}. \tag{2.109}$$

In this situation, the matrix $\mathcal{F} - \mathcal{G}_{i-1}$ reduces to

$$\mathcal{F} - \mathcal{G}_{i-1} = \begin{bmatrix} \overline{A}^{\mathsf{T}}(2I_K - \mu P^{-1}H_{i-1}) & -\overline{A}^{\mathsf{T}}(I_K - \mu P^{-1}H_{i-2}) \\ I_K & 0 \end{bmatrix}. \tag{2.110}$$

Moreover, we also assume H_i is iteration independent, i.e.,

$$H_i = H, \quad \forall\ i = 1, 2, \cdots \tag{2.111}$$

This assumption holds for quadratic costs $J_k(w)$. Under the above conditions, we have

$$(\mathcal{F}-\mathcal{G}_{i-1})\begin{bmatrix}\mathbb{1}_K\\ \mathbb{1}_K\end{bmatrix}=\begin{bmatrix}\overline{A}^{\mathsf{T}}\mathbb{1}_K\\ \mathbb{1}_K\end{bmatrix}=\begin{bmatrix}\mathbb{1}_K\\ \mathbb{1}_K\end{bmatrix}, \tag{2.112}$$

which implies that $\lambda_1=1$ is one eigenvalue of $\mathcal{F}-\mathcal{G}_{i-1}$ no matter what the step-size μ is. However, since $\mathcal{W}_0$ is initialized as $\mathcal{V}\mathcal{Y}_0$ and, hence, lies in range$(\mathcal{V})$, the eigenvalue $\lambda_1=1$ will not influence the convergence of recursion (2.106) (the detailed explanation is spelled out in Sec.3.1 and 3.2 in Chapter 3). Let $\{\lambda_k\}_{k=2}^{2K}$ denote the remaining eigenvalues of $\mathcal{F}-\mathcal{G}_{i-1}$, and introduce

$$\rho(\mathcal{F}-\mathcal{G}_{i-1})\ \triangleq\ \max\{|\lambda_2|,|\lambda_3|,\cdots,|\lambda_{2K}|\}. \tag{2.113}$$

It is $\rho(\mathcal{F}-\mathcal{G}_{i-1})$ that determines the convergence of recursion (2.106): the exact diffusion recursion (2.106) will diverge if $\rho(\mathcal{F}-\mathcal{G}_{i-1})>1$, and will converge if $\rho(\mathcal{F}-\mathcal{G}_{i-1})<1$.

Example 1 (Diverging case). Consider the following left-stochastic matrix A:

$$A=\begin{bmatrix}0 & 0 & 0 & 1\\ 0 & 0.5 & 0.5 & 0\\ 1 & 0 & 0.5 & 0\\ 0 & 0.5 & 0 & 0\end{bmatrix}. \tag{2.114}$$

It can be verified that A is primitive, left-stochastic but not balanced. For such A, its Perron eigenvector p can be calculated in advance, and hence P is also known. Also, H_{i-1} is assumed to satisfy

$$P^{-1}H_{i-1}=\text{diag}\{20,1,1,1\}\in\mathbb{R}^{4\times 4} \tag{2.115}$$

Substituting the above A and PH_{i-1} into $\mathcal{F}-\mathcal{G}_{i-1}$ shown in (2.110), it can be verified that

$$\rho(\mathcal{F}-\mathcal{G}_{i-1})>1 \tag{2.116}$$

for *any* step-size $\mu>0$. The proof is given in Appendix 2.D by appealing to the Jury test for stability. In the top plot in Fig. 2.9, we show the spectral radius $\rho(\mathcal{F}-\mathcal{G}_{i-1})$ for step-sizes $\mu\in[1e^{-6},3]$. It is observed that $\rho(\mathcal{F}-\mathcal{G}_{i-1})>1$.

By following similar arguments, we can find a counter example such that EXTRA will also diverge for any step-size $\mu > 0$, even if we assume the Perron eigenvector p is known in advance. For example, if

$$A = \begin{bmatrix} 0.36 & 0.99 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0.6 & 0 \\ 0 & 0 & 0.02 & 0 & 0.95 \\ 0 & 0 & 0.98 & 0.4 & 0 \\ 0.64 & 0 & 0 & 0 & 0.05 \end{bmatrix} \in \mathbb{R}^{5\times 5} \tag{2.117}$$

and

$$P^{-1}H_{i-1} = \mathrm{diag}\{20, 1, 1, 1, 1\} \in \mathbb{R}^{5\times 5}, \tag{2.118}$$

one can verify that EXTRA will diverge for any $\mu > 0$ by following the arguments in Appendix 2.D. As a result, the push-sum based algorithms [100,123] that extend EXTRA to non-symmetric networks cannot always converge. This example indicates that the stability range $(c_{\text{lower}}, c_{\text{upper}})$ provided in [100,123] may not always be feasible. ■

Example 2 (Converging case). Consider the following left-stochastic matrix A:

$$A = \begin{bmatrix} 0.3 & 0.6 & 0.2 & 0 & 0 \\ 0.2 & 0.2 & 0 & 0.3 & 0 \\ 0.1 & 0.1 & 0.5 & 0.3 & 0.2 \\ 0 & 0.1 & 0.3 & 0.4 & 0.1 \\ 0.4 & 0 & 0 & 0 & 0.7 \end{bmatrix}. \tag{2.119}$$

It can be verified that A is primitive and not balanced. Also, H_{i-1} is assumed to satisfy

$$P^{-1}H_{i-1} = \mathrm{diag}\{10, 10, 10, 10, 10\} \in \mathbb{R}^{5\times 5}. \tag{2.120}$$

Substituting the above A and $P^{-1}H_{i-1}$ into (2.110), it can be verified that $\rho(\mathcal{F}) = 0.9923$. Therefore, when μ is sufficiently small, $\mathcal{F}$ will dominate in $\mathcal{F}-\mathcal{G}_{i-1}$ and $\rho(\mathcal{F}-\mathcal{G}_{i-1}) < 1$. The simulations in Fig. 2.10 confirm this fact. In particular, it is observed that $\rho(\mathcal{F}-\mathcal{G}_{i-1}) < 1$ when $\mu < 0.2$. As a result, the exact diffusion will converge when $\mu < 0.2$ under this setting.

■

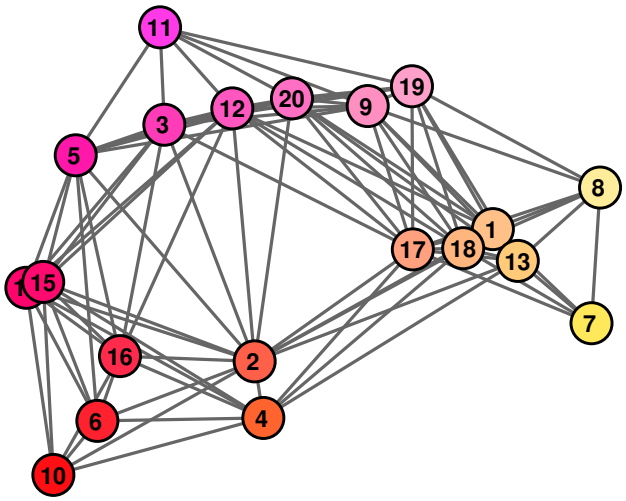

Figure 2.3: Network topology used in the simulations.

2.6 Numerical Experiments

In this section we illustrate the performance of the proposed exact diffusion algorithm. In all figures, the y-axis indicates the relative error, i.e., $\|\mathcal{w}_i - \mathcal{w}^\star\|^2/\|\mathcal{w}_0 - \mathcal{w}^\star\|^2$, where $\mathcal{w}_i = \text{col}\{w_{1,i}, \cdots, w_{N,i}\} \in \mathbb{R}^{NM}$ and $\mathcal{w}^\star = \text{col}\{w^\star, \cdots, w^\star\} \in \mathbb{R}^{NM}$.

2.6.1 Distributed Least-squares

In this experiment, we focus on solving the least-squares problem over the network shown in 2.3:

$$w^o = \arg\min_{w\in\mathbb{R}^M} \quad \frac{1}{2K}\sum_{k=1}^{K}\|U_k w - d_k\|^2. \tag{2.121}$$

where the network size $N = 20$ and the dimension $M = 30$. Each entry in both $U_k \in \mathbb{R}^{50\times 30}$ and $d_k \in \mathbb{R}^{50}$ is generated from the standard Gaussian distribution $\mathcal{N}(0, 1)$.

We compare the convergence behavior of standard diffusion and the exact diffusion algorithm in the simulation. The left-stochastic matrix A is generated through the averaging rule (see (2.20)), and each agent k employs step-size $\mu_k = \mu_o/n_k$ (see (2.23)) where μ_o is a small constant step-size. The convergence of both algorithms is shown in Fig. 2.4, where we set

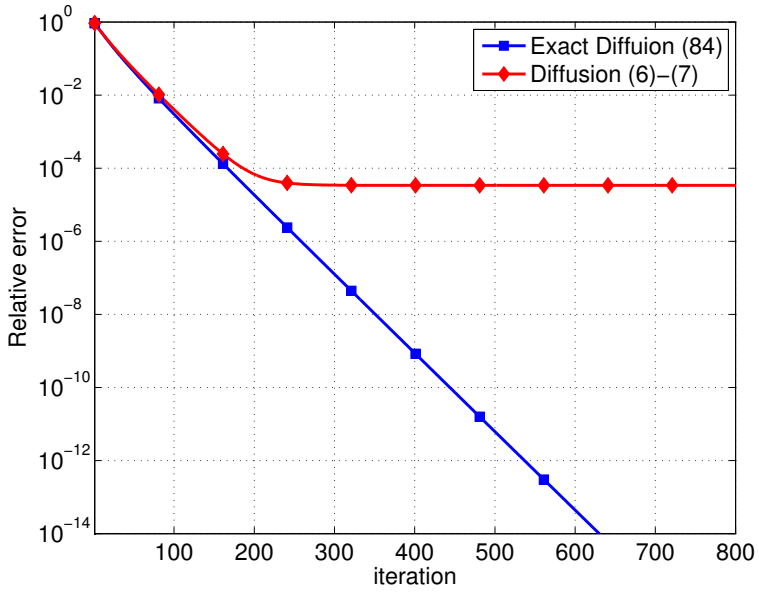

Figure 2.4: Convergence comparison between standard diffusion and exact diffusion for the distributed least-squares (2.121).

$\mu_o = 0.01$. It is observed that the standard diffusion algorithm converges to a neighborhood of w^o on the order $O(\mu_o^2)$, while the exact diffusion converges exponentially fast to the exact solution w^o. This figure confirms that exact diffusion corrects the bias in standard diffusion.

2.6.2 Distributed Logistic Regression

We next consider a pattern classification scenario. Each agent k holds local data samples $\{h_{k,j}, \gamma_{k,j}\}_{j=1}^{L}$, where $h_{k,j} \in \mathbb{R}^M$ is a feature vector and $\gamma_{k,j} \in \{-1, +1\}$ is the corresponding label. Moreover, the value L is the number of local samples at each agent. All agents will cooperatively solve the regularized logistic regression problem over the network in Fig. 2.3:

$$w^o = \arg\min_{w \in \mathbb{R}^M} \sum_{k=1}^{K} \Big[\frac{1}{L} \sum_{\ell=1}^{L} \ln\big(1+\exp(-\gamma_{k,\ell} h_{k,\ell}^{\mathsf{T}} w)\big) + \frac{\rho}{2}\|w\|^2 \Big]. \tag{2.122}$$

In the experiments, we set $N = 20$, $M = 30$, and $L = 50$. For local data samples $\{h_{k,j}, \gamma_{k,j}\}_{j=1}^{L}$ at agent k, each $h_{k,j}$ is generated from the standard normal distribution $\mathcal{N}(0; 10 I_M)$. To generate $\gamma_{k,j}$, we first generate an auxiliary random vector $w_0 \in \mathbb{R}^M$ with

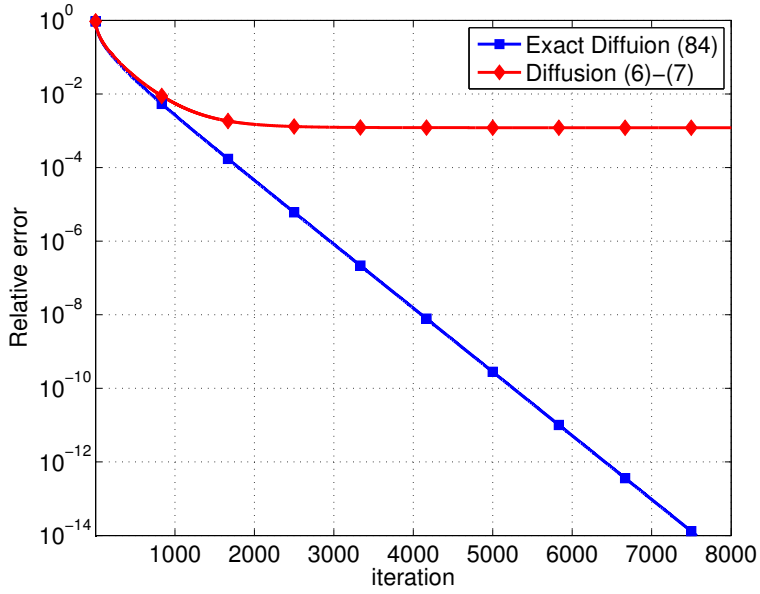

Figure 2.5: Convergence comparison between standard diffusion and exact diffusion for distributed logistic regression (2.122).

each entry following $\mathcal{N}(0,1)$. Next, we generate $\gamma_{k,j}$ from a uniform distribution $\mathcal{U}(0,1)$. If $\gamma_{k,j} \leq 1/[1+\exp(-(h_{k,j})^{\mathsf{T}} w_0)]$ then $\gamma_{k,j}$ is set as $+1$; otherwise $\gamma_{k,j}$ is set as -1. We set $\rho = 0.1$.

We still compare the convergence behavior of the standard diffusion and exact diffusion. The left-stochastic matrix A is generated through the averaging rule, and each agent k employs step-size $\mu_k = \mu_o/n_k$. The convergence of both algorithms is shown in Fig. 2.5. The step-size $\mu_o = 0.05$. It is also observed that the exact diffusion corrects the bias in standard diffusion.

2.6.3 Benefits of Balanced Left-stochastic Policies

In this subsection we illustrate one of the benefits of balanced left-stochastic combination matrices — they can speed up the convergence.

In the first experiment, we consider a network with a highly unbalanced topology as shown in Fig. 2.6. Nodes 1 and 2 are "celebrities" with many neighbors, while the other

48 nodes just have two neighbors each. Such a network topology is quite common in social networks.

Interestingly, both the maximum degree rule and the Metropolis rule will generate the same doubly-stochastic combination matrix for this network. Let L be the Laplacian matrix associated with that network, then the generated doubly-stochastic combination matrix is

$$A = I - L/49. \tag{2.123}$$

This combination matrix A merges information just slightly better than the identity matrix I because the term $L/49$ is quite small, which is not efficient. In contrast, the normal agent k (where $3 \leq k \leq 50$) will assign $1/3$ to incoming information from agents 1 and 2 if the averaging rule is used, which combines information more efficiently and hence leads to faster convergence. In Fig. 2.7, we compare exact diffusion and EXTRA methods over the distributed least-square problem (2.121). The experimental setting is the same as in Sec. 3.3.1 except for the combination rules. Exact diffusion employs the left-stochastic matrix generated by the averaging rule while EXTRA employs a doubly-stochastic combination matrix (recall that EXTRA [75] has convergence guarantees only for doubly-stochastic matrices). The step-sizes are carefully chosen such that each algorithm reaches its fastest convergence. As expected, it is observed that exact diffusion with the averaging rule is almost three times faster than EXTRA with doubly-stochastic combination matrices.

In the second experiment, we consider the distributed least-square problem (2.121) and assume the Lipschitz constants associated with each local cost function differs drastically. In this experiment, we set $N = 20$, and the network topology is the same as in Fig. 2.3. Among all nodes, we assume for 4 random nodes that the local data U_k and d_k are generated from $\mathcal{N}(0, 100)$ while in the remaining nodes they are generated from $\mathcal{N}(0, 1)$. Under such setting, each local Lipschitz constant is quite different. We again compare the convergence between exact diffusion and EXTRA where the combination rule for exact diffusion is generated according to the Hastings rule (2.40) while EXTRA employs the Metropolis combination matrix, which is doubly stochastic. Fig. 2.8 depicts the convergence for each algorithm. Again, the step-sizes are carefully chosen such that each algorithm reaches its

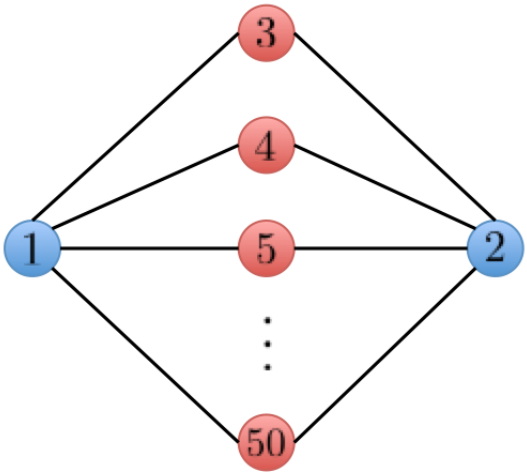

Figure 2.6: A highly unbalanced network topology.

fastest convergence. As expected, it is observed that exact diffusion with Hastings rule is almost four times faster than EXTRA with the doubly-stochastic matrix.

2.6.4 Exact Diffusion for General Left-Stochastic A

In this subsection we test exact diffusion for the general left-stochastic A shown in Section 2.5. In Fig. 2.9 we test the setting of Example 1 in which A is in the form of (2.114) and H is (2.115). We introduce $\rho = \rho(\mathcal{F} - \mathcal{G}_{i-1})$. In the top plot, we illustrate how ρ varies with step-size μ. In this plot, the step-size varies over $[10^{-6}, 3]$, and the interval between two consecutive μ is 10^{-6}. It is observed that $\rho > 1$ for any $\mu \in [10^{-6}, 3]$, which confirms with our conclusion that exact diffusion will diverge for any step-size μ under the setting in Example 1. In the bottom plot of Fig. 2.9 we illustrate the standard diffusion converges to a neighborhood of $w^\star$ on the order of $O(\mu^2)$ for $\mu = 0.01$, while the exact diffusion diverges.

In Fig. 2.10 we test the setting of Example 2 in which A is in the form of (2.119) and H is of (2.120). In the top plot, we illustrate how ρ varies with μ. It is observed that $\rho < 1$ when $\mu < 0.2$, which implies that the exact diffusion recursion (2.106) will converge when $\mu < 0.2$. In the bottom figure, with $\mu = 0.001$ it is observed that exact diffusion will converge exactly to $w^\star$. Figures. 2.9 and 2.10 confirm that general left-stochastic A cannot always guarantee

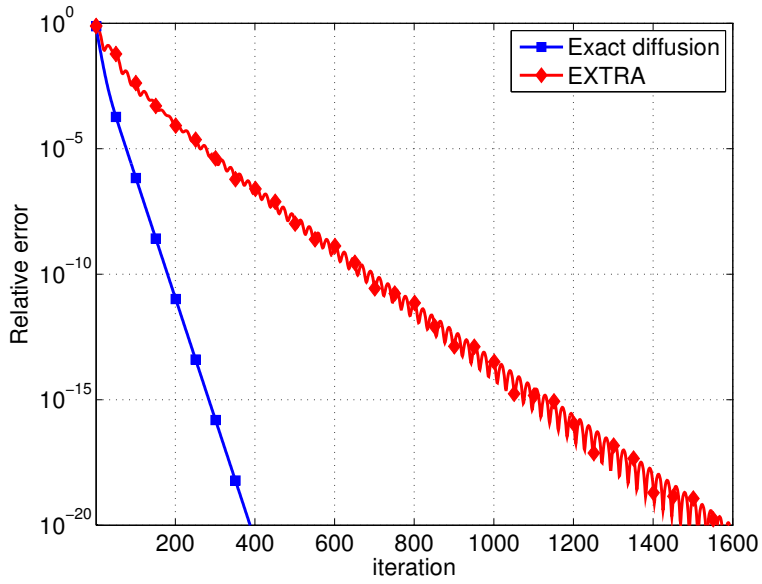

Figure 2.7: Convergence comparison between exact diffusion and EXTRA for highly unbalanced network. Exact diffusion is with the averaging rule while EXTRA is with the doubly stochastic rule.

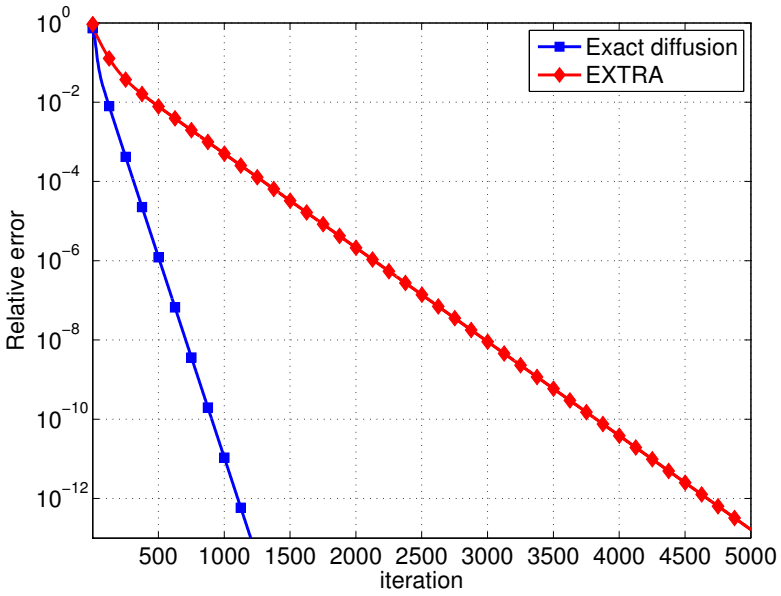

Figure 2.8: Convergence comparison between exact diffusion and EXTRA for the scenario in which local Lipschitz constants differ drastically. Exact diffusion is with the Hastings rule (2.40) while EXTRA is with the doubly stochastic rule.

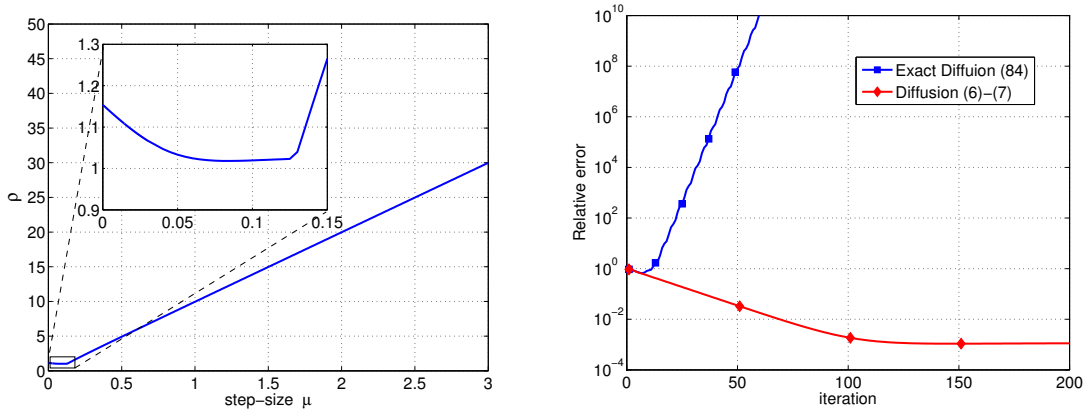

Figure 2.9: Exact diffusion under the setting of Example 1 in Section 2.5. Top: $\rho > 1$ no matter what value μ is. Bottom: Convergence comparison between diffusion and exact diffusion when $\mu = 0.01$.

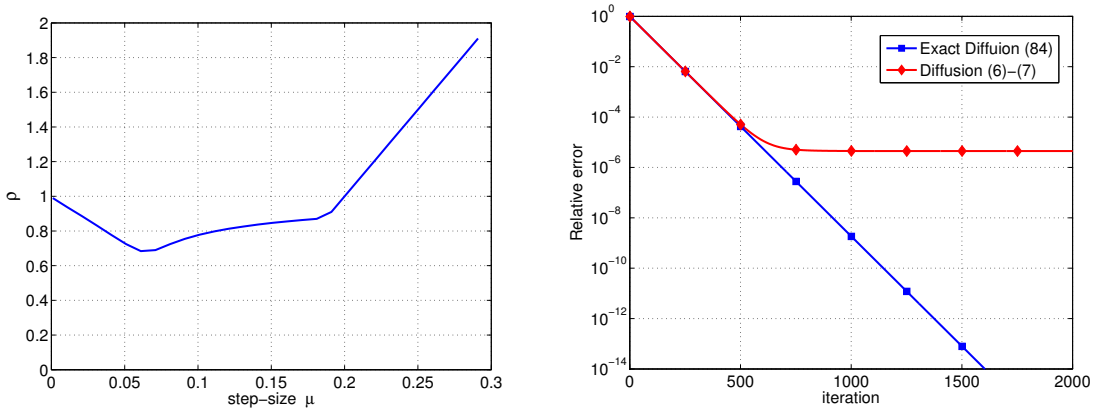

Figure 2.10: Exact diffusion under the setting of Example 2 in Section 2.5. Top: $\rho < 1$ when $\mu < 0.2$. Bottom: Convergence comparison between standard diffusion and exact diffusion when $\mu = 0.001$.

convergence to $w^\star$.

2.7 Concluding Remarks

This chapter developed a diffusion optimization strategy with guaranteed exact convergence for a broad class of combination policies. The strategy is applicable to the locally-balanced left-stochastic combination matrices which are able to endow the algorithm with faster convergence rate, more flexible step-size choices and better privacy-preserving properties compared to doubly-stochastic combination matrices. Chapter 3 establishes analytically, and

by means of examples and simulations, the superior convergence and stability properties of exact diffusion implementations.

2.A Formulation of Primal Methods

In this section we formulate two prominent primal methods that are based on gradient descent: consensus [5–13] and diffusion [1,4,34,36,71]. For simplicity, we assume the network is strongly connected, undirected, and the associated combination matrix A is symmetric and doubly-stochastic (see equation (1.8)). For such combination matrix A, it holds that

$$1 = \lambda_1(A) > \lambda_2(A) \geq \cdots \geq \lambda_K(A) > -1 \tag{2.124}$$

and hence the matrix $I - A$ is positive semi-definite.

We introduce the eigenvalue decomposition $(I - A)/2 = U\Sigma U$ where Σ is a diagonal matrix with nonnegative diagonal entries. We also define $V = U\Sigma^{1/2}U$ and note that $V^T = V$, $V^2 = (I - A)/2$, and more importantly,

$$Vw = 0 \Longleftrightarrow w(1) = w(2) = \cdots = w(K) \tag{2.125}$$

which is established in Lemma 2.4. In the above expression, the notation $w(k)$ refers to the k-th element in w. If we define

$$\mathcal{w} \triangleq \mathrm{col}\{w_1, \cdots, w_K\} \in \mathbb{R}^{KM}, \tag{2.126}$$

$$\mathcal{A} \triangleq A \otimes I_M \in \mathbb{R}^{KM}, \tag{2.127}$$

$$\mathcal{V} \triangleq V \otimes I_M \in \mathbb{R}^{KM}, \tag{2.128}$$

$$\mathcal{J}^o(\mathcal{w}) \triangleq \frac{1}{K}\sum_{k=1}^{K} J_k(w_k), \tag{2.129}$$

it then follows that $\mathcal{V}^2 = (I_{KM} - \mathcal{A})/2$ and

$$\mathcal{V}\mathcal{w} = 0 \Longleftrightarrow w_1 = w_2 = \cdots = w_K. \tag{2.130}$$

Using (2.129) and (2.130), we find that problem (1.1) can be rewritten as the constrained problem

$$\begin{aligned} \min_{\mathcal{W}} \quad & \mathcal{J}^o(\mathcal{W}), \\ \text{s.t.} \quad & \mathcal{V}\mathcal{W} = 0. \end{aligned} \tag{2.131}$$

One common approach to solve such problems is the penalty method [127, Sec. 4.1, 4.3], [124][pp. 277 and Ch.6], [128, Ch.9], [126]. We penalize the constraints and transform (2.131) to the unconstrained problem

$$\min_{\mathcal{W}} \quad \mathcal{J}^o(\mathcal{W}) + \frac{1}{\mu}\|\mathcal{V}\mathcal{W}\|^2 \tag{2.132}$$

where $\mu > 0$ is a constant coefficient. Using $\mathcal{V}^2 = (I_{KM} - \mathcal{A})/2$, problem (2.132) is equivalent to

$$\min_{\mathcal{W}} \quad \mathcal{J}^o(\mathcal{W}) + \frac{1}{2\mu}\mathcal{W}^{\mathsf{T}}(I - \mathcal{A})\mathcal{W} \tag{2.133}$$

2.A.1 Consensus Strategy

If we solve problem (2.133) with gradient descent, we get

$$\begin{aligned} \mathcal{W}_{i+1} &= \mathcal{W}_i - \mu\Big(\nabla\mathcal{J}^o(\mathcal{W}_i) + \frac{1}{\mu}(I - \mathcal{A})\mathcal{W}_i\Big) \\ &= \mathcal{A}\mathcal{W}_i - \mu\nabla\mathcal{J}^o(\mathcal{W}_i) \end{aligned} \tag{2.134}$$

which is exactly the consensus approach for solving the distributed optimization problem (1.1).

2.A.2 Diffusion Strategy

If we solve problem (2.133) with incremental gradient descent, we get

$$\begin{cases} \boldsymbol{\phi}_i = \mathcal{W}_i - \mu\nabla\mathcal{J}^o(\mathcal{W}_i), \\ \mathcal{W}_{i+1} = \boldsymbol{\phi}_i - \mu\cdot\frac{1}{\mu}(I - \mathcal{A})\boldsymbol{\phi}_i = \mathcal{A}\boldsymbol{\phi}_i. \end{cases} \tag{2.135}$$

By substituting the first equation into the second one, we have

$$\mathcal{w}_{i+1} = \mathcal{A}\Big(\mathcal{w}_i - \mu\nabla\mathcal{J}^o(\mathcal{w}_i)\Big) \tag{2.136}$$

A more generalized version of diffusion is derived in Sec. 2.3.

2.A.3 Other Primal Methods

There are still approaches that solve the unconstrained problem (2.133) using other primal methods such as Newton or quasi-Newton methods. Similar to the consensus or diffusion strategies, the resulting distributed Newton [78,79,129] and Quasi-Newton methods [80] only have primal variables in their recursions. Since all of these methods focus on solving the penalized problem (2.133) rather than the real problem (2.131), they cannot converge to the exact global solution unless a decaying step-size μ is used.

2.B Formulation of Primal-Dual Methods

Different from the above-mentioned primal methods, primal-dual methods aim at solving the constrained problem (2.131) directly.

2.B.1 EXTRA Method

Several of primal-dual methods are based on the augmented Lagrangian technique [130–132], [133, Sec.17.4]. Different from the Lagrangian method, which focuses on the standard Lagrangian function:

$$L(\mathcal{w},\mathcal{y}) = \mathcal{J}^o(\mathcal{w}) + \frac{1}{\mu}\mathcal{y}^{\mathsf{T}}(\mathcal{V}\mathcal{w}) \tag{2.137}$$

where $\mathcal{y}$ is the dual variable (also known as the Lagrangian multiplier), the augmented Lagrangian method introduces an extra quadratic term to the standard Lagrangian function:

$$\begin{aligned} L_a(\mathcal{w},\mathcal{y}) &= \mathcal{J}^o(\mathcal{w}) + \frac{1}{\mu}\mathcal{y}^{\mathsf{T}}(\mathcal{V}\mathcal{w}) + \frac{1}{\mu}\|\mathcal{V}\mathcal{w}\|^2 \\ &= \mathcal{J}^o(\mathcal{w}) + \frac{1}{\mu}\mathcal{y}^{\mathsf{T}}(\mathcal{V}\mathcal{w}) + \frac{1}{2\mu}\mathcal{w}^{\mathsf{T}}(I-\mathcal{A})\mathcal{w}. \end{aligned} \tag{2.138}$$

The introduction of the quadratic term $\|\mathcal{V}w\|^2$ will impose strong convexity to $L(w,y)$ (in terms of w) and hence will ensure a wider stability range and faster convergence.

The primal-descent and dual-ascent approach to determining the saddle-point of the above augmented Lagrangian function is

$$\begin{cases} w_{i+1} = w_i - \mu\nabla\mathcal{J}^o(w_i) - \mathcal{V}y_i - (I-\mathcal{A})w_i = \mathcal{A}w_i - \mu\nabla\mathcal{J}^o(w_i) - \mathcal{V}y_i, \\ y_{i+1} = y_i + \mathcal{V}w_{i+1} \end{cases} \tag{2.139}$$

From the first recursion, we have

$$w_{i+1} - w_i = \mathcal{A}(w_i - w_{i-1}) - \mu(\nabla\mathcal{J}^o(w_i) - \nabla\mathcal{J}^o(w_{i-1})) - \mathcal{V}(y_i - y_{i-1}) \tag{2.140}$$

Substituting the second recursion in (2.139) into the above recursion, we reach

$$w_{i+1} = \bar{\mathcal{A}}(2w_i - w_{i-1}) - \mu(\nabla\mathcal{J}^o(w_i) - \nabla\mathcal{J}^o(w_{i-1})) \tag{2.141}$$

where $\bar{\mathcal{A}} = (I+A)/2$. Recursion (2.141) is the EXTRA algorithm proposed in [74]. Different from consensus or diffusion, EXTRA will converge to the exact global solution with constant step-size μ. To see it, we observe the fixed point (w^∞, y^∞) of recursion (2.139) satisfies the following condition

$$\begin{cases} \mu\nabla\mathcal{J}^o(w^\infty) + \mathcal{V}y^\infty = 0, \\ \mathcal{V}w^\infty = 0. \end{cases} \tag{2.142}$$

which is essentially the optimality condition of problem (1.1) [75, Proposition 2.1].

2.B.2 Exact Diffusion Method

When the first recursion in (2.139) is updated in an incremental manner, we will reach exact diffusion; see the derivation in Sec. 2.4.

2.B.3 Tracking Method

The tracking method [92–98] is another variant of primal-dual approach. The DIGing method (which is one of the tracking approaches) was originally proposed as follows:

$$\begin{cases} \mathcal{W}_i = \mathcal{A}\mathcal{W}_{i-1} - \mu\mathcal{X}_{i-1} \\ \mathcal{X}_i = \mathcal{A}\mathcal{X}_{i-1} + \nabla\mathcal{J}^o(\mathcal{W}_i) - \nabla\mathcal{J}^o(\mathcal{W}_{i-1}) \end{cases} \tag{2.143}$$

where $\mathcal{X}_i$ is the auxiliary variable that aims to track the gradient $\frac{1}{K}(\mathbb{1}_k \otimes I_M)\nabla\mathcal{J}^o(\mathcal{W}_i) = \frac{1}{K}\sum_{k=1}^{K}\nabla J_k(w_{k,i})$. In fact, the second recursion (2.143) falls into the family of the dynamic average algorithm [134, 135]:

$$\mathcal{X}_i = \mathcal{A}\mathcal{X}_{i-1} + \mathcal{R}_i - \mathcal{R}_{i-1} \tag{2.144}$$

where $\mathcal{R}_i$ is a dynamic and time-varying signal. When $\mathcal{R}_i$ converges, it is proved in [134–136] that $\mathcal{X}_{k,i} \to \frac{1}{K}\sum_{k=1}^{K}\mathcal{R}_{k,i}$. Inspired by this result, it holds that $\mathcal{X}_{k,i} \to \frac{1}{K}\sum_{k=1}^{K}\nabla J_k(w_{k,i})$ as iteration i increases. Next we explain how the tracking method converges to the optimal solution. When the iteration i is large enough, the first recursion in (2.143) can be approximated by

$$\mathcal{W}_i = \mathcal{A}\mathcal{W}_{i-1} - \mu\mathbb{1}_K \otimes \frac{1}{K}\sum_{k=1}^{K}\nabla J_k(w_{k,i}). \tag{2.145}$$

Suppose each $w_{k,i}$ generated by the above recursion converges to a fixed point $w^\star$, it then holds that $\frac{1}{K}\sum_{k=1}^{K}\nabla J_k(w^\star) = 0$ and, hence, the fixed point $w^\star$ is the optimal solution to problem (1.1). A formal proof of convergence for DIGing is presented in [93].

The DIGing method can also be interpreted as a primal-dual algorithm. Note that problem (1.1) can be reformulated as the following constrained problem

$$\begin{aligned} \min_{\mathcal{W}} \quad & \mathcal{J}^o(\mathcal{W}) \\ \text{s.t.} \quad & (I - \mathcal{A})\mathcal{W} = 0 \end{aligned} \tag{2.146}$$

which is equivalent to (2.131) since $\mathcal{V}\mathcal{W} = 0 \Longleftrightarrow (I - \mathcal{A})\mathcal{W} = 0$. The augmented Lagrangian function associated with the above problem is given by

$$L_a(\mathcal{W}, \mathcal{Y}) = \mathcal{J}^o(\mathcal{W}) + \frac{1}{\mu}\mathcal{Y}^{\mathsf{T}}(I - \mathcal{A})\mathcal{W} + \frac{1}{2\mu}\|\mathcal{W}\|^2_{I-\mathcal{A}^2}, \tag{2.147}$$

where $\mathcal{Y}$ is the dual variable. The primal-descent and dual-ascent approach to solve the saddle-point of the above Lagrangian function is

$$\begin{cases} \mathcal{W}_{i+1} = \mathcal{W}_i - \mu\nabla\mathcal{J}^o(\mathcal{W}_i) - (I-\mathcal{A})\mathcal{Y}_i - (I-\mathcal{A}^2)\mathcal{W}_i = \mathcal{A}^2\mathcal{W}_i - \mu\nabla\mathcal{J}^o(\mathcal{W}_i) - (I-\mathcal{A})\mathcal{Y}_i, \\ \mathcal{Y}_{i+1} = \mathcal{Y}_i + (I-\mathcal{A})\mathcal{W}_{i+1} \end{cases} \tag{2.148}$$

The above recursion is essentially equivalent to DIGing. To see that, we substitute the second recursion of (2.148) into the first one and remove the dual variable to reach

$$\mathcal{W}_{i+1} = 2\mathcal{A}\mathcal{W}_i - \mathcal{A}^2\mathcal{W}_{i-1} - \mu\big(\nabla\mathcal{J}^o(\mathcal{W}_i) - \nabla\mathcal{J}^o(\mathcal{W}_{i-1})\big). \tag{2.149}$$

On the other hand, if we substitute the second recursion of the DIGing method (2.143) into the first one and remove the auxiliary variable $\mathcal{X}_i$, we will get the same recursion as in (2.149). In this sense, the primal-dual recursion (2.148) is equivalent to the DIGing recursion (2.143).

Tracking methods, as reported in [93, 97, 98, 137], work well in time-varying or directed networks. However, they require twice the amount of communications per iteration than EXTRA or exact diffusion. Tracking methods also have variants that fall into the adapt-then-combine (ATC) framework. These variants can also be interpreted as the primal-dual method as described in [138].

2.B.4 Distributed ADMM

Distributed ADMM is among the first algorithms that were proved to converge linearly to the global solution under the assumption that each local cost function $J_k(w)$ is strongly convex with Lipschitz-continuous gradient. Instead of solving the constrained problem (2.131) or (2.146), distributed ADMM solves an alternative constrained problem

$$\begin{aligned} \min_{\mathcal{W}} \quad & \frac{1}{K}\sum_{k=1}^{K} J_k(w_k) \\ \text{s.t.} \quad & w_k = w_\ell \quad \forall (i,j) \in \mathcal{E} \end{aligned} \tag{2.150}$$

where $\mathcal{E}$ is the set of all edges in the graph. We remark that problem (2.150) is essentially equivalent to the constrained problem (2.131) and (2.146). However, the ADMM approach

requires an explicit constraint $w_i = w_j$ while the formulations in (2.131) and (2.146) just imply $w_i = w_j$ and they do not have such constraints explicitly. We introduce an auxiliary variable z_{ij} to decouple w_i and w_j in the constraints. As a result, problem (2.150) can be reformulated as

$$\begin{aligned} \min_{\mathcal{W}} \quad & \frac{1}{K}\sum_{k=1}^{K} J_k(w_k) \\ \text{s.t.} \quad & w_k = z_{k\ell}, \quad w_\ell = z_{k\ell} \quad \forall (i,j) \in \mathcal{E} \end{aligned} \tag{2.151}$$

which can be further rewritten as

$$\begin{aligned} \min_{\mathcal{W},\mathcal{Z}} \quad & \mathcal{J}^o(\mathcal{W}) \\ \text{s.t.} \quad & \mathcal{C}\mathcal{W} + \mathcal{Z} = 0 \end{aligned} \tag{2.152}$$

where $\mathcal{Z} = \text{col}\{z_{ij}\}$, $\mathcal{C} = C \otimes I_M$, $C = [C_1; C_2] \in \mathbb{R}^{2E\times K}$, and C_1 and C_2 are defined as

$$[C_1]_{ek} \begin{cases} = 1, & \text{if } e = (k,\ell), \\ = 0, & \text{otherwise,} \end{cases} \qquad [C_2]_{e\ell} \begin{cases} = 1, & \text{if } e = (k,\ell), \\ = 0, & \text{otherwise,} \end{cases} \tag{2.153}$$

Problem (2.152) falls into the framework of Alternating Direction Method of Multipliers (ADMM). Now we introduce the augmented Lagrangian function associated with problem (2.152) as

$$L_a(\mathcal{W},\mathcal{Z},\mathcal{Y}) = \mathcal{J}^o(\mathcal{W}) + \mathcal{Y}^{\mathsf{T}}(\mathcal{C}\mathcal{W}+\mathcal{Z}) + \frac{\rho}{2}\|\mathcal{C}\mathcal{W}+z\|^2. \tag{2.154}$$

Compared with the augmented Lagrangian function used in EXTRA and DIGing method, the function in (2.154) involves two primal variables $\mathcal{W}$ and $\mathcal{Z}$. The ADMM approach to find the saddle-point of the Lagrangian function (2.154) is

$$\begin{cases} \mathcal{W}_{i+1} = \arg\min_{\mathcal{W}} L(\mathcal{W},\mathcal{Z}_i,\mathcal{Y}_i) = \arg\min_{\mathcal{W}}\{\mathcal{J}^o(\mathcal{W}) + \mathcal{Y}_i^{\mathsf{T}}(\mathcal{C}\mathcal{W}) + \frac{\rho}{2}\|\mathcal{C}\mathcal{W}+z_i\|^2\}, \\ \mathcal{Z}_{i+1} = \arg\min_{\mathcal{Z}} L(\mathcal{W}_i,\mathcal{Z},\mathcal{Y}_{i-1}) = \arg\min_{\mathcal{Z}}\{\mathcal{Y}_i^{\mathsf{T}}\mathcal{Z} + \frac{\rho}{2}\|\mathcal{C}\mathcal{W}_{i+1}+z\|^2\}, \\ \mathcal{Y}_{i+1} = \mathcal{Y}_i + \rho(\mathcal{C}\mathcal{W}_{i+1}+\mathcal{Z}_{i+1}). \end{cases} \tag{2.155}$$

Note that the second subproblem related to $\mathcal{Z}$ is a quadratic problem and it has a closed-form solution. Substituting the special structure of the matrix $\mathcal{C}$ and removing the variable

z from the above recursion, we reach a distributed implementation of the recursion (also see Algorithm 1 in [74]):

$$\begin{cases} \text{find } w_{k,i+1} \text{ by solving} \nabla J_k(w_{k,i+1}) + \beta_{k,i} + 2\rho|\mathcal{N}_k| w_{k,i+1} - \rho\Big(|\mathcal{N}_k| w_{k,i} + \sum_{\ell\in\mathcal{N}_k} w_{\ell,i}\Big) = 0, \\ \beta_{k,i+1} = \beta_{k,i} + \rho\Big(|\mathcal{N}_k| w_{k,i+1} + \sum_{\ell\in\mathcal{N}_k} w_{\ell,i+1}\Big). \end{cases} \tag{2.156}$$

2.C Formulation of Dual Methods

The Lagrangian function of problem (2.131) is

$$L(w, y) = \mathcal{J}^o(w) + y^{\mathsf{T}}\mathcal{V}w. \tag{2.157}$$

The Lagrangian dual function is

$$\begin{aligned} g(y) &= \inf_{w}\{L(w, y)\} = \inf_{w}\{\mathcal{J}^o(w) + y^{\mathsf{T}}\mathcal{V}w\} = -\sup_{w}\{-(\mathcal{V}y)^{\mathsf{T}}w - \mathcal{J}^o(w)\} \\ &= -\mathcal{J}^*(-\mathcal{V}y) \end{aligned} \tag{2.158}$$

where $\mathcal{J}^*(z) = \sup_{\mathcal{W}}(z^{\mathsf{T}}w - \mathcal{J}^o(w))$ is the conjugate of the function $\mathcal{J}^o(w)$. In this section, we only consider the family of problems where $\mathcal{J}^*(\cdot)$ has a closed-from. For example, if $\mathcal{J}^o(w)$ is affine, negative logarithm, exponential, strictly convex quadratic, log-determinant, then the conjugate function $\mathcal{J}^*(\cdot)$ has a closed-form, see [124, Sec. 3.1].

Since problem (2.131) is feasible (with at least the quantity $\mathbb{1}_K \otimes I_M$ as a feasible solution) and the constraints are all linear equations, the Slater condition [124, Sec. 5.2.3] implies that strong duality holds for problem (2.131). In other words, it holds that

$$\mathcal{J}^o(w^\star) = g(y^\star) = \inf_{w}\{\mathcal{J}^o(w) + (y^\star)^{\mathsf{T}}\mathcal{V}w\} \tag{2.159}$$

where $w^\star$ and $y^\star$ are the optimal primal and dual solution respectively. The first equality holds because of strong duality and the second equality holds because of the definition of the Lagrangian dual function in (2.158). Relation (2.159) implies

$$w^\star = \arg\min_{w}\{\mathcal{J}^o(w) + (y^\star)^{\mathsf{T}}\mathcal{V}w\}. \tag{2.160}$$

As a result, one can solve for $w^\star$ with the following two steps. First, we calculate $y^\star$ according to

$$y^\star = \arg\max_{y} \{g(y)\}. \tag{2.161}$$

Second, we calculate $w^\star$ according to (2.160). This approach is named as the dual method [110, 111].

Now we derive a dual method that can solve problem (1.1) in a distributed manner. Note that since $g(y) = -\mathcal{J}^*(-\mathcal{V}y)$ as derived in (2.158), we can therefore reach $y^\star$ by solving

$$\min_{y} \quad \mathcal{J}^*(-\mathcal{V}y). \tag{2.162}$$

There are many approaches to solve the above unconstrained problem. For example, the gradient descent method is

$$y_i = y_{i-1} + \mu \mathcal{V} \nabla \mathcal{J}^*(-\mathcal{V}y_{i-1}). \tag{2.163}$$

Multiplying $-\mathcal{V}$ to both sides of the above recursion, we reach

$$-\mathcal{V}y_i = -\mathcal{V}y_{i-1} - \mu \frac{(I-\mathcal{A})}{2} \nabla \mathcal{J}^*(-\mathcal{V}y_{i-1}) \tag{2.164}$$

which is equivalent to

$$z_i = z_{i-1} - \mu \frac{(I-\mathcal{A})}{2} \nabla \mathcal{J}^*(z_{i-1}) \tag{2.165}$$

where we defined $z = \mathcal{V}y_i$. Note that

$$\begin{aligned} \mathcal{J}^*(z) &= \sup_{\mathcal{W}} \{z^{\mathsf{T}} w - \mathcal{J}^o(w)\} \\ &= \sup_{\mathcal{W}} \{\sum_{k=1}^{K} [z_k^{\mathsf{T}} w_k - \frac{1}{K} J_k(w_k)]\} \\ &= \sum_{k=1}^{K} J_k^*(z_k) \end{aligned} \tag{2.166}$$

where we defined $J_k^*(z_k) = \sup_{w_k} \{z_k^{\mathsf{T}} w_k - \frac{1}{K} J_k(w_k)\}$ and it has a closed-form[1]. With the above relation, we have

$$\nabla \mathcal{J}^*(z) = \mathrm{col}\{\nabla J_1^*(z_1), \nabla J_2^*(z_2), \cdots, \nabla J_K^*(z_K)\} \tag{2.167}$$

[1]Recall that we only consider special form of $J_k(w)$ in this section such that the conjugate $J_k^*(\cdot)$ has a closed-form

and hence the update in (2.165) can be conducted in a decentralized manner. Finally, once the optimal dual solution $z^\star$ is reached, we can derive the primal optimal solution $w^\star$ as

$$\begin{aligned} w^\star &= \arg\min_{w}\{\mathcal{J}^o(w) + (z^\star)^{\mathsf{T}} w\} \\ &= \arg\min_{w}\{\sum_{k=1}^{K}[\frac{1}{K}J_k(w_k) + (z_k^\star)^{\mathsf{T}} w_k]\} \end{aligned} \tag{2.168}$$

which implies that

$$w_k^\star = \arg\min_{w_k}\{J_k(w_k) + K(z_k^\star)^{\mathsf{T}} w_k\} \tag{2.169}$$

The dual method is summarized in Algorithm 2.3. It is observed that all communication occurs when solving the dual problem (2.162), see the recursion (2.171). If the gradient descent recursion (2.162) is accelerated, we can reduce the communication cost. An important observation is that the dual problem (2.162) is unconstrained, which exactly falls into the Nesterov's acceleration framework. As a result, the authors in [110] propose to solve the dual problem (2.162) with Nesterov's recursion [139]:

$$\begin{cases} x_i = z_{i-1} - \mu\frac{(I-\mathcal{A})}{2}\nabla\mathcal{J}^*(z_{i-1}), \\ z_i = x_i + \beta(x_i - x_{i-1}) \end{cases} \tag{2.170}$$

where x_i is an auxiliary variable and β is the momentum coefficient. This accelerated dual method is listed in Algorithm 1 of [110]. According to [110, 111], the above accelerated dual method reaches the theoretical lower communication bound for distributed algorithms, and is theoretically better than the other accelerated algorithms based on EXTRA and DIGing. However, one should note that The dual methods usually require the conjugate function $\mathcal{J}^*(z)$ to have a closed-form, which significantly limits the application of this family of methods.

2.D Proof of (2.116)

The characteristic polynomial of $\mathcal{F} - \mathcal{G}_{i-1}$ is given by

$$Q(\lambda) = (\lambda - 1)D(\lambda), \quad \text{where} \quad D(\lambda) = \sum_{k=0}^{7} a_k\lambda^k \tag{2.172}$$

Algorithm 2.3 Basic dual approach

Setting: Each agent k derives the closed-form of the conjugate function $J_k^*(z_k) = \sup_{w_k}\{z_k^{\mathsf{T}} w_k - \frac{1}{K} J_k(w_k)\}$

Repeat until $z_{k,i} \to z_k^\star$**:**

$$z_{k,i} = z_{k,i-1} - \frac{\mu}{2}\Big(\nabla J_k^*(z_{k,i-1}) - \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\nabla J_\ell^*(z_{\ell,i-1})\Big) \tag{2.171}$$

Output: Each agent k derives w_k^* according to $w_k^\star = \arg\min_{w_k}\{J_k(w_k) + K(z_k^\star)^{\mathsf{T}} w_k\}$.

and

$$\begin{aligned}
&a_7 = 32, \quad a_6 = 384\mu - 128, \quad a_5 = 682\mu^2 - 1512\mu + 248,\\
&a_4 = 429\mu^3 - 2458\mu^2 + 2712\mu - 288,\\
&a_3 = 80\mu^4 - 1346\mu^3 + 3672\mu^2 - 2692\mu + 210,
\end{aligned} \tag{2.173}$$

$$\begin{aligned}
&a_2 = -240\mu^4 + 1649\mu^3 - 2904\mu^2 + 1593\mu - 98,\\
&a_1 = 240\mu^4 - 976\mu^3 + 1260\mu^2 - 552\mu + 28,\\
&a_0 = -80\mu^4 + 244\mu^3 - 252\mu^2 + 92\mu - 4.
\end{aligned} \tag{2.174}$$

It is easy to observe from (2.172) that $\lambda = 1$ is one eigenvalue of $\mathcal{F} - \mathcal{G}_{i-1}$. As mentioned in (2.112) and its following paragraph, this eigenvalue $\lambda = 1$ does not influence the convergence of recursion (2.106) because of the initial conditions. It is the roots of $D(\lambda)$ that decide the convergence of the exact diffusion recursion (2.106). Now we will prove that there always exists some root that stays outside the unit-circle no matter what the step-size μ is. In other words, $D(\lambda)$ is not stable for any μ.

Since $D(\lambda)$ is a 7-th order polynomial, its roots are not easy to calculate directly. Instead, we apply the Jury stability criterion [140] to decide whether $D(\lambda)$ has roots outside the unit-

circle. First we construct the Jury table as shown in Fig. 2.11, where

$$b_k = \begin{vmatrix} a_0 & a_{7-k} \\ a_7 & a_k \end{vmatrix} = a_0 a_k - a_7 a_{7-k},\ k = 0, \cdots, 6 \tag{2.175}$$

$$c_k = \begin{vmatrix} b_0 & b_{6-k} \\ b_6 & b_k \end{vmatrix} = b_0 b_k - b_6 b_{6-k},\ k = 0, \cdots, 5 \tag{2.176}$$

$$\vdots$$

$$f_k = \begin{vmatrix} e_0 & e_{3-k} \\ e_3 & e_k \end{vmatrix} = e_0 e_k - e_3 e_{3-k},\ k = 0, \cdots, 2. \tag{2.177}$$

According to the Jury stability criterion, $D(\lambda)$ is stable (i.e., all roots of $D(\lambda)$ are within the unit-circle) if, and only if, the following conditions hold:

$$\begin{aligned} &D(1) > 0, \quad (-1)^7 D(-1) > 0, \quad |a_0| < a_7, \quad |b_0| > |b_6| \\ &|c_0| > |c_5|, \quad |d_0| > |d_4|, \quad |e_0| > |e_3|, \quad |f_0| > |f_2|. \end{aligned} \tag{2.178}$$

If any one of the above conditions is violated, $D(\lambda)$ is not stable. Next we check each of the conditions:

(1) $D(1) > 0$ is satisfied for any $\mu > 0$ since

$$D(1) = \sum_{k=0}^{7} a_k = 25\mu > 0. \tag{2.179}$$

(2) $(-1)^7 D(-1) > 0$. To guarantee this condition, we need to require that

$$\begin{aligned} &(-1)^7 D(-1) \\ &= 640\mu^4 - 4644\mu^3 + 11228\mu^2 - 9537\mu + 1036 > 0. \end{aligned} \tag{2.180}$$

With the help of Matlab, we can verify that

$$(-1)^7 D(-1) > 0 \ \text{ when } \mu < 0.1265 \text{ or } \mu > 3.0410. \tag{2.181}$$

(3) $|a_0| < a_7$. To guarantee this condition, we need

$$|-80\mu^4 + 244\mu^3 - 252\mu^2 + 92\mu - 4| < 32, \tag{2.182}$$

λ^0	λ^1	λ^2	λ^3	λ^4	λ^5	λ^6	λ^7
a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0
b_0	b_1	b_2	b_3	b_4	b_5	b_6	
b_6	b_5	b_4	b_3	b_2	b_1	b_0	
c_0	c_1	c_2	c_3	c_4	c_5		
c_5	c_4	c_3	c_2	c_1	c_0		
$\vdots$	$\vdots$	$\vdots$					
f_0	f_1	f_2					

Figure 2.11: The Jury table for the 7-th order system.

which is equivalent to requiring

$$-0.1884 < \mu < 1.6323. \tag{2.183}$$

With (2.179), (2.181) and (2.183), we conclude that when

$$0 < \mu < 0.1265, \tag{2.184}$$

conditions (1), (2) and (3) will be satisfied simultaneously. Moreover, with the help of Matlab, we can also verify that the step-size range (2.184) will also meet conditions (4) $|b_0| > |b_6|$, (5) $|c_0| > |c_5|$ and (6) $|d_0| > |d_4|$. Now we check the last two conditions.

(7) $|e_0| > |e_3|$. To guarantee this condition, the step-size μ is required to satisfy

$$0.0438 < \mu < 0.1265. \tag{2.185}$$

(8) $|f_0| > |f_2|$. To guarantee this condition, the step-size μ is required to satisfy

$$0 < \mu < 0.0412. \tag{2.186}$$

Comparing (2.184), (2.185) and (2.186), it is observed that the intersection of these three ranges is empty, which implies that there does not exist a value for μ that makes all conditions (1)–(8) hold. Therefore, we conclude that $D(\lambda)$ is not stable for any step-size μ.

CHAPTER 3

Exact Diffusion for Distributed Optimization: Convergence Analysis

In this chapter, we will establish the linear convergence of exact diffusion using its primal-dual form (2.88). This is a challenging task due to the coupled dynamics among the agents. To facilitate the analysis, we first apply a useful coordinate transformation and characterize the error dynamics in this transformed domain. Then, we show analytically that exact diffusion is stable, converges linearly, and has a wider stability range than EXTRA consensus strategy [75]. We also compare the performance of exact diffusion to other existing linearly convergent algorithms besides EXTRA, such as DIGing [93] and Aug-DGM [95, 96] with numerical simulations.

3.1 Convergence of Exact Diffusion

The purpose of the analysis in this section is to establish the exact convergence of $w_{k,i}$ to $w^\star$, for all agents in the network, and to show that this convergence attains an exponential rate.

3.1.1 The Optimality Condition

Lemma 3.1 (Optimality Condition) *If condition* (2.9) *holds and block vectors* $(\mathcal{w}^\star, \mathcal{y}^\star)$ *exist that satisfy:*

$$\overline{\mathcal{A}}^{\mathsf{T}}\mathcal{M}\nabla\mathcal{J}^o(\mathcal{w}^\star) + \mathcal{P}^{-1}\mathcal{V}\mathcal{y}^\star = 0, \tag{3.1}$$

$$\mathcal{V}\mathcal{w}^\star = 0. \tag{3.2}$$

then it holds that the block entries of $\mathcal{W}^\star$ satisfy:

$$w_1^\star = w_2^\star = \cdots = w_K^\star = w^\star \tag{3.3}$$

where $w^\star$ is the unique solution to problem (2.1).

Proof. From (2.67), we have

$$\mathcal{V}\mathcal{W}^\star = 0 \Longleftrightarrow w_1^\star = w_2^\star = \cdots = w_K^\star. \tag{3.4}$$

Next we check $w_k^\star = w^\star$. Since $\mathcal{P} > 0$, condition (3.1) is equivalent to

$$\mathcal{P}\overline{\mathcal{A}}^{\mathsf{T}}\mathcal{M}\nabla\mathcal{J}^o(\mathcal{W}^\star) + \mathcal{V}\mathcal{Y}^\star = 0. \tag{3.5}$$

Let $\mathcal{I} = \mathbb{1}_K \otimes I_M \in \mathbb{R}^{MK\times M}$. Multiplying by $\mathcal{I}^{\mathsf{T}}$ gives

$$\begin{aligned} 0 &= \mathcal{I}^{\mathsf{T}}\big(\mathcal{P}\overline{\mathcal{A}}^{\mathsf{T}}\mathcal{M}\nabla\mathcal{J}^o(\mathcal{W}^\star) + \mathcal{V}\mathcal{Y}^\star\big) \overset{(a)}{=} \mathcal{I}^{\mathsf{T}}\mathcal{P}\overline{\mathcal{A}}^{\mathsf{T}}\mathcal{M}\nabla\mathcal{J}^o(\mathcal{W}^\star) \\ &= \sum_{k=1}^{K} p_k\mu_k\nabla J_k(w_k^\star) \overset{(2.9)}{=} \frac{1}{\beta}\sum_{k=1}^{K} q_k\nabla J_k(w_k^\star), \end{aligned} \tag{3.6}$$

where equality (a) holds because $\mathcal{V}$ is symmetric and (2.67). Since $\beta \neq 0$, we conclude that $\sum_{k=1}^{K} q_k\nabla J_k(w_k^\star) = 0$, which shows that the entries $\{w_k^\star\}$, which are identical, must coincide with the minimizer $w^\star$ of (2.1) Observe that since $\mathcal{J}^\star(w)$ is assumed strongly-convex, then the solution to problem (2.1), $w^\star$, is unique, and hence $\mathcal{W}^\star$ is also unique. However, since $\mathcal{V}$ is rank-deficient, there can be multiple solutions $\mathcal{Y}^\star$ satisfying (3.3). Using an argument similar to [74,75], we can show that among all possible $\mathcal{Y}^\star$, there is a unique solution $\mathcal{Y}_o^\star$ lying in the column span of $\mathcal{V}$.

Lemma 3.2 (Particular solution pair) *When condition* (2.9) *holds and $\mathcal{J}^o(w)$ defined by* (1.1) *is strongly-convex, there exists a unique pair of variables $(\mathcal{W}^\star, \mathcal{Y}_o^\star)$, in which $\mathcal{Y}_o^\star$ lies in the range space of $\mathcal{V}$, that satisfies conditions* (3.1)-(3.2).

Proof. First we prove that there always exist some block vectors $(\mathcal{W}^\star, \mathcal{Y}^\star)$ satisfying (3.1)–(3.2). Indeed, when $\mathcal{J}^o(w)$ is strongly-convex, the solution to problem (2.1), $w^\star$, exists and

is unique. Let $w^\star = \mathbb{1}_K \otimes w^\star$. We conclude from Lemma 2.4 that condition (3.2) holds. Next we check whether there exists some $y^\star$ such that

$$\mathcal{P}^{-1}\mathcal{V}y^\star = -\overline{\mathcal{A}}^{\mathsf{T}}\mathcal{M}\nabla\mathcal{J}^o(w^\star), \tag{3.7}$$

or equivalently,

$$\begin{aligned}\mathcal{V}y^\star &= -\mathcal{P}\overline{\mathcal{A}}^{\mathsf{T}}\mathcal{M}\nabla\mathcal{J}^o(w^\star) \\ &= -\overline{\mathcal{A}}\mathcal{P}\mathcal{M}\nabla\mathcal{J}^o(w^\star) = -\frac{1}{\beta}\overline{\mathcal{A}}\nabla\mathcal{J}^\star(w^\star),\end{aligned} \tag{3.8}$$

where the last equality holds because

$$\mathcal{P}\mathcal{M}\nabla\mathcal{J}^o(w^\star) = \begin{bmatrix} \mu_1 p_1 \nabla J_1(w^\star) \\ \vdots \\ \mu_K p_K \nabla J_K(w^\star) \end{bmatrix} \overset{(2.9)}{=} \begin{bmatrix} \frac{q_1}{\beta}\nabla J_1(w^\star) \\ \vdots \\ \frac{q_K}{\beta}\nabla J_K(w^\star) \end{bmatrix} \overset{(2.74)}{=} \frac{1}{\beta}\nabla\mathcal{J}^\star(w^\star), \tag{3.9}$$

To prove the existence of $y^\star$, we need to show that $\overline{\mathcal{A}}\nabla\mathcal{J}^\star(w^\star)$ lies in range$(\mathcal{V})$. Indeed, observe that

$$\mathcal{I}^{\mathsf{T}}\overline{\mathcal{A}}\nabla\mathcal{J}^\star(w^\star) = \mathcal{I}^{\mathsf{T}}\nabla\mathcal{J}^\star(w^\star) \overset{(a)}{=} \sum_{k=1}^{K} q_k \nabla J_k(w^\star) = 0 \tag{3.10}$$

where the equality (a) holds because of equation (2.74). Equality (3.10) implies that $\overline{\mathcal{A}}\nabla\mathcal{J}^\star(w^\star)$ is orthogonal to span$(\mathcal{I})$, i.e., span$(\mathbb{1}_K \otimes I_M)$. With (2.67) we have

$$\begin{aligned}\overline{\mathcal{A}}\nabla\mathcal{J}^\star(w^\star) \perp \text{null}(\mathcal{V}) &\Leftrightarrow \overline{\mathcal{A}}\nabla\mathcal{J}^\star(w^\star) \in \text{range}(\mathcal{V}^{\mathsf{T}}) \\ &\Leftrightarrow \overline{\mathcal{A}}\nabla\mathcal{J}^\star(w^\star) \in \text{range}(\mathcal{V}),\end{aligned} \tag{3.11}$$

where the last "$\Leftrightarrow$" holds because $\mathcal{V}$ is symmetric.

We now establish the existence of the unique pair $(w^\star, y_o^\star)$. Thus, let $(w^\star, y^\star)$ denote an arbitrary solution to (3.3). Let further $y_o^\star$ denote the projection of $y^\star$ onto the column span of $\mathcal{V}$. It follows that $\mathcal{V}(y^\star - y_o^\star) = 0$ and, hence, $\mathcal{V}y^\star = \mathcal{V}y_o^\star$. Therefore, the pair $(w^\star, y_o^\star)$ also satisfies conditions (3.1)-(3.2).

Next we verify the uniqueness of $y_o^\star$ by contradiction. Suppose there is a different $y_1^\star$ lying in $\mathcal{R}(\mathcal{V})$ that also satisfies condition (3.1). We let $y_o^\star = \mathcal{V}x_o^\star$ and $y_1^\star = \mathcal{V}x_1^\star$. Substituting $y_o^\star$

and $\mathcal{Y}_1^\star$ into condition (3.1), we have

$$\overline{\mathcal{A}}^{\mathsf{T}}\mathcal{M}\nabla\mathcal{J}^o(\mathcal{W}^\star)+\mathcal{P}^{-1}\mathcal{V}^2\mathcal{X}_o^\star=0, \tag{3.12}$$

$$\overline{\mathcal{A}}^{\mathsf{T}}\mathcal{M}\nabla\mathcal{J}^o(\mathcal{W}^\star)+\mathcal{P}^{-1}\mathcal{V}^2\mathcal{X}_1^\star=0. \tag{3.13}$$

Subtracting (3.13) from (3.12) and recall $\mathcal{P}>0$, we have $\mathcal{V}^2(\mathcal{X}_o^\star-\mathcal{X}_1^\star)=0$, which leads to $\mathcal{V}(\mathcal{X}_o^\star-\mathcal{X}_1^\star)=0\Longleftrightarrow\mathcal{Y}_o^\star=\mathcal{Y}_1^\star$. This contradicts the assumption that $\mathcal{Y}_o^\star\neq\mathcal{Y}_1^\star$. ■

Using the above auxiliary results, we will show that $(\mathcal{W}_i,\mathcal{Y}_i)$ generated through the exact diffusion (2.88) will converge exponentially fast to $(\mathcal{W}^\star,\mathcal{Y}_o^\star)$.

3.1.2 Error Recursion

Let $\mathcal{W}^\star=\mathbb{1}_K\otimes w^\star$, which corresponds to a block vector with $w^\star$ repeated K times. Introduce further the error vectors

$$\widetilde{\mathcal{W}}_i=\mathcal{W}^\star-\mathcal{W}_i,\quad\widetilde{\mathcal{Y}}_i=\mathcal{Y}_o^\star-\mathcal{Y}_i. \tag{3.14}$$

The first step in the convergence analysis is to examine the evolution of these error quantities. Multiplying the second recursion of (2.88) by $\mathcal{V}$ from the left gives:

$$\mathcal{V}\mathcal{Y}_i=\mathcal{V}\mathcal{Y}_{i-1}+\frac{1}{2}(\mathcal{P}-\mathcal{P}\mathcal{A})\mathcal{W}_i. \tag{3.15}$$

Substituting (3.15) into the first recursion of (2.88), we have

$$\begin{cases}\overline{\mathcal{A}}^{\mathsf{T}}\widetilde{\mathcal{W}}_i=\overline{\mathcal{A}}^{\mathsf{T}}\Big(\widetilde{\mathcal{W}}_{i-1}+\mathcal{M}\nabla\mathcal{J}^o(\mathcal{W}_{i-1})\Big)+\mathcal{P}^{-1}\mathcal{V}\mathcal{Y}_i,\\ \quad\widetilde{\mathcal{Y}}_i=\widetilde{\mathcal{Y}}_{i-1}-\mathcal{V}\mathcal{W}_i.\end{cases} \tag{3.16}$$

Subtracting optimality conditions (3.1)–(3.2) from (3.16) leads to

$$\begin{cases}\overline{\mathcal{A}}^{\mathsf{T}}\widetilde{\mathcal{W}}_i=\overline{\mathcal{A}}^{\mathsf{T}}\Big(\widetilde{\mathcal{W}}_{i-1}+\mathcal{M}\big[\nabla\mathcal{J}^o(\mathcal{W}_{i-1})-\nabla\mathcal{J}^o(\mathcal{W}^\star)\big]\Big)-\mathcal{P}^{-1}\mathcal{V}\widetilde{\mathcal{Y}}_i,\\ \quad\widetilde{\mathcal{Y}}_i=\widetilde{\mathcal{Y}}_{i-1}+\mathcal{V}\widetilde{\mathcal{W}}_i.\end{cases} \tag{3.17}$$

Next we examine the difference $\nabla\mathcal{J}^o(\mathcal{W}_{i-1})-\nabla\mathcal{J}^o(\mathcal{W}^\star)$. To begin with, we get from (2.78) that

$$\nabla\mathcal{J}^o(\mathcal{W}_{i-1})-\nabla\mathcal{J}^o(\mathcal{W}^\star)=\begin{bmatrix}\nabla J_1(w_{1,i-1})-\nabla J_1(w^\star)\\ \vdots\\ \nabla J_K(w_{K,i-1})-\nabla J_K(w^\star)\end{bmatrix} \tag{3.18}$$

When $\nabla J_k(w)$ is twice-differentiable (see Assumption 3.1), we can appeal to the mean-value theorem from Lemma D.1 in [1], which allows us to express each difference in (3.18) in the following integral form in terms of Hessian matrices for any $k = 1, 2, \ldots, N$:

$$\nabla J_k(w_{k,i-1}) - \nabla J_k(w^\star) = -\Big(\int_0^1 \nabla^2 J_k\big(w^\star - r\widetilde{w}_{k,i-1}\big)\,dr\Big)\,\widetilde{w}_{k,i-1}.$$

If we let

$$H_{k,i-1} \triangleq \int_0^1 \nabla^2 J_k\big(w^\star - r\widetilde{w}_{k,i-1}\big)\,dr \in \mathbb{R}^{M\times M}, \tag{3.19}$$

and introduce the block diagonal matrix:

$$\mathcal{H}_{i-1} \triangleq \mathrm{diag}\{H_{1,i-1}, H_{2,i-1}, \cdots, H_{K,i-1}\}, \tag{3.20}$$

then we can rewrite (3.18) in the form:

$$\nabla \mathcal{J}^o(\mathcal{w}_{i-1}) - \nabla \mathcal{J}^o(\mathcal{w}^\star) = -\mathcal{H}_{i-1}\widetilde{\mathcal{w}}_{i-1}. \tag{3.21}$$

Substituting into (3.17) we get

$$\begin{cases} \overline{\mathcal{A}}^{\mathsf{T}}\widetilde{\mathcal{w}}_i = \overline{\mathcal{A}}^{\mathsf{T}}(I_{MK} - \mathcal{M}\mathcal{H}_{i-1})\widetilde{\mathcal{w}}_{i-1} - \mathcal{P}^{-1}\mathcal{V}\widetilde{\mathcal{y}}_i, \\ \quad \widetilde{\mathcal{y}}_i = \widetilde{\mathcal{y}}_{i-1} + \mathcal{V}\widetilde{\mathcal{w}}_i. \end{cases} \tag{3.22}$$

which is also equivalent to

$$\begin{bmatrix} \overline{\mathcal{A}}^{\mathsf{T}} & \mathcal{P}^{-1}\mathcal{V} \\ -\mathcal{V} & I_{MK} \end{bmatrix} \begin{bmatrix} \widetilde{\mathcal{w}}_i \\ \widetilde{\mathcal{y}}_i \end{bmatrix} = \begin{bmatrix} \overline{\mathcal{A}}^{\mathsf{T}}(I_{MK} - \mathcal{M}\mathcal{H}_{i-1}) & 0 \\ 0 & I_{MK} \end{bmatrix} \begin{bmatrix} \widetilde{\mathcal{w}}_{i-1} \\ \widetilde{\mathcal{y}}_{i-1} \end{bmatrix}. \tag{3.23}$$

Using the relations $\overline{\mathcal{A}}^{\mathsf{T}} = \frac{I_{MK}+\mathcal{A}^{\mathsf{T}}}{2}$ and $\mathcal{V}^2 = \frac{\mathcal{P}-\mathcal{P}\mathcal{A}^{\mathsf{T}}}{2}$, it is easy to verify that

$$\begin{bmatrix} \overline{\mathcal{A}}^{\mathsf{T}} & \mathcal{P}^{-1}\mathcal{V} \\ -\mathcal{V} & I_{MK} \end{bmatrix}^{-1} = \begin{bmatrix} I_{MK} & -\mathcal{P}^{-1}\mathcal{V} \\ \mathcal{V} & I_{MK} - \mathcal{V}\mathcal{P}^{-1}\mathcal{V} \end{bmatrix}. \tag{3.24}$$

Substituting into (3.24) gives

$$\begin{bmatrix} \widetilde{\mathcal{w}}_i \\ \widetilde{\mathcal{y}}_i \end{bmatrix} = \begin{bmatrix} \overline{\mathcal{A}}^{\mathsf{T}}(I_{MK} - \mathcal{M}\mathcal{H}_{i-1}) & -\mathcal{P}^{-1}\mathcal{V} \\ \mathcal{V}\overline{\mathcal{A}}^{\mathsf{T}}(I_{MK} - \mathcal{M}\mathcal{H}_{i-1}) & I_{MK} - \mathcal{V}\mathcal{P}^{-1}\mathcal{V} \end{bmatrix} \begin{bmatrix} \widetilde{\mathcal{w}}_{i-1} \\ \widetilde{\mathcal{y}}_{i-1} \end{bmatrix}. \tag{3.25}$$

That is, the error vectors evolve according to:

$$\boxed{\begin{bmatrix} \widetilde{\mathcal{W}}_i \\ \widetilde{\mathcal{Y}}_i \end{bmatrix} = (\mathcal{B} - \mathcal{T}_{i-1}) \begin{bmatrix} \widetilde{\mathcal{W}}_{i-1} \\ \widetilde{\mathcal{Y}}_{i-1} \end{bmatrix}} \tag{3.26}$$

where

$$\mathcal{B} \triangleq \begin{bmatrix} \overline{\mathcal{A}}^{\mathsf{T}} & -\mathcal{P}^{-1}\mathcal{V} \\ \mathcal{V}\overline{\mathcal{A}}^{\mathsf{T}} & I_{MK} - \mathcal{V}\mathcal{P}^{-1}\mathcal{V} \end{bmatrix}, \tag{3.27}$$

$$\mathcal{T}_i \triangleq \begin{bmatrix} \overline{\mathcal{A}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_i & 0 \\ \mathcal{V}\overline{\mathcal{A}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_i & 0 \end{bmatrix}. \tag{3.28}$$

Relation (3.26) is the error dynamics for the exact diffusion algorithm. We next examine its convergence properties.

3.1.3 Proof of Convergence

We first introduce a common assumption.

Assumption 3.1 (Conditions on cost functions) *Each $J_k(w)$ is twice differentiable, and its Hessian matrix satisfies*

$$\nabla^2 J_k(w) \leq \delta I_M. \tag{3.29}$$

Moreover, there exists at least one agent k_o such that $J_{k_o}(w)$ is ν-strongly convex, i.e.

$$\nabla^2 J_{k_o}(w) > \nu I_M. \tag{3.30}$$

■

Note that when $J_k(w)$ is twice differentiable, condition (3.29) is equivalent to requiring each $\nabla J_k(w)$ to be δ-Lipschitz continuous [1]. In addition, condition (3.30) ensures the strong convexity of $\mathcal{J}^o(w)$ and $\mathcal{J}^\star(w)$, and the uniqueness of w^o and $w^\star$. It follows from (3.29)–(3.30) and the definition (3.19) that

$$H_{k,i-1} \leq \delta I_M, \ \forall k \quad \text{and} \quad H_{k_o,i-1} \geq \nu I_M. \tag{3.31}$$

The direct convergence analysis of recursion (3.26) is challenging. To facilitate the analysis, we identify a convenient change of basis and transform (3.26) into another equivalent form that is easier to handle. To do that, we first let

$$B \triangleq \begin{bmatrix} \overline{A}^{\mathsf{T}} & -P^{-1}V \\ V\overline{A}^{\mathsf{T}} & I_K - VP^{-1}V \end{bmatrix} \in \mathbb{R}^{2K\times 2K}. \tag{3.32}$$

It holds that $\mathcal{B} = B \otimes I_M$. In the following lemma we introduce a decomposition for matrix B that will be fundamental to the subsequent analysis.

Lemma 3.3 (Fundamental Decomposition) *The matrix B admits the following eigen-decomposition*

$$B = XDX^{-1}, \tag{3.33}$$

where

$$D = \left[\begin{array}{c|c} I_2 & 0 \\ \hline 0 & D_1 \end{array}\right] \tag{3.34}$$

and $D_1 \in \mathbb{R}^{(2K-2)\times(2K-2)}$ is a diagonal matrix with complex entries. The magnitudes of the diagonal entries satisfy

$$\begin{aligned} |D_1(2k-3,2k-3)| = |D_1(2k-2,2k-2)| = \sqrt{\lambda_k(\overline{A})} < 1, \\ \forall\ k = 2,3,\cdots N. \end{aligned} \tag{3.35}$$

Moreover,

$$X = \left[\begin{array}{c|c} R & X_R \end{array}\right], \quad X^{-1} = \left[\begin{array}{c} L \\ \hline X_L \end{array}\right], \tag{3.36}$$

where $X_R \in \mathbb{R}^{2K\times(2K-2)}$ and $X_L \in \mathbb{R}^{(2K-2)\times 2K}$, and R and L are given by

$$R = \begin{bmatrix} \mathbb{1}_K & 0 \\ 0 & \mathbb{1}_K \end{bmatrix} \in \mathbb{R}^{2K\times 2}, L = \begin{bmatrix} p^{\mathsf{T}} & 0 \\ 0 & \frac{1}{K}\mathbb{1}_K^{\mathsf{T}} \end{bmatrix} \in \mathbb{R}^{2\times 2K}. \tag{3.37}$$

Proof See Appendix 3.A. ■

Remark 3.1 (Other possible decompositions) *The eigendecomposition* (3.33) *for* B *is not unique because we can always scale* X *and* X^{-1} *to achieve different decompositions. In this paper, we will study the following family of decompositions:*

$$B = X' D (X')^{-1}, \tag{3.38}$$

where

$$X' = \left[\begin{array}{c|c} R & \frac{1}{c} X_R \end{array} \right], \quad (X')^{-1} = \left[\begin{array}{c} L \\ \hline c X_L \end{array} \right], \tag{3.39}$$

and c *can be set to any nonzero constant value. We will exploit later the choice of* c *in identifying the stability range for exact diffusion.* ■

For convenience, we introduce the vectors:

$$r_1 = \begin{bmatrix} \mathbb{1}_K \\ 0 \end{bmatrix}, r_2 = \begin{bmatrix} 0 \\ \mathbb{1}_K \end{bmatrix}, \ell_1 = \begin{bmatrix} p \\ 0 \end{bmatrix}, \ell_2 = \begin{bmatrix} 0 \\ \frac{1}{K}\mathbb{1}_K \end{bmatrix}, \tag{3.40}$$

so that

$$R = [r_1 \ r_2], \quad L = \begin{bmatrix} \ell_1^{\mathsf{T}} \\ \ell_2^{\mathsf{T}} \end{bmatrix}. \tag{3.41}$$

Using (3.33)–(3.41), we write

$$\begin{aligned} \mathcal{B} &= (X' \otimes I_M)(D \otimes I_M)((X')^{-1} \otimes I_M) \ \stackrel{\Delta}{=} \ \mathcal{X}' \mathcal{D} (\mathcal{X}')^{-1} \\ &= \begin{bmatrix} \mathcal{R}_1 & \mathcal{R}_2 & \frac{1}{c}\mathcal{X}_R \end{bmatrix} \begin{bmatrix} I_M & 0 & 0 \\ 0 & I_M & 0 \\ 0 & 0 & \mathcal{D}_1 \end{bmatrix} \begin{bmatrix} \mathcal{L}_1^{\mathsf{T}} \\ \mathcal{L}_2^{\mathsf{T}} \\ c\mathcal{X}_L \end{bmatrix}, \end{aligned} \tag{3.42}$$

where $\mathcal{D}_1 = D_1 \otimes I_M$,

$$\mathcal{R}_1 = \begin{bmatrix} \mathcal{I} \\ 0 \end{bmatrix} \in \mathbb{R}^{2KM \times M}, \ \ \mathcal{R}_2 = \begin{bmatrix} 0 \\ \mathcal{I} \end{bmatrix} \in \mathbb{R}^{2KM \times M}, \tag{3.43}$$

$$\mathcal{L}_1 = \begin{bmatrix} \mathcal{P} \\ 0 \end{bmatrix} \in \mathbb{R}^{2KM \times M}, \ \ \mathcal{L}_2 = \begin{bmatrix} 0 \\ \frac{1}{K}\mathcal{I} \end{bmatrix} \in \mathbb{R}^{2KM \times M}, \tag{3.44}$$

while $\mathcal{X}_R = X_R \otimes I_M \in \mathbb{R}^{2KM\times 2(K-1)M}$ and $\mathcal{X}_L = x_L \otimes I_M \in \mathbb{R}^{2(K-1)M\times 2KM}$. Moreover, we are also introducing

$$\mathcal{I}{=}\mathbb{1}_K \otimes I_M \in \mathbb{R}^{KM\times M},\ \overline{\mathcal{P}}{=}p\otimes I_M \in \mathbb{R}^{KM\times M}, \tag{3.45}$$

where the variable $\overline{\mathcal{P}}$ defined above is different from the earlier variable $\mathcal{P} = P\otimes I_M \in^{KM\times NM}$. Multiplying both sides of (3.26) by $(\mathcal{X}')^{-1}$:

$$(\mathcal{X}')^{-1}\begin{bmatrix}\widetilde{\mathcal{W}}_i\\ \widetilde{\mathcal{Y}}_i\end{bmatrix}{=}[(\mathcal{X}')^{-1}(\mathcal{B}-\mathcal{T}_{i-1})\mathcal{X}'](\mathcal{X}')^{-1}\begin{bmatrix}\widetilde{\mathcal{W}}_{i-1}\\ \widetilde{\mathcal{Y}}_{i-1}\end{bmatrix} \tag{3.46}$$

leads to

$$\begin{bmatrix}\bar{\mathcal{x}}_i\\ \widehat{\mathcal{x}}_i\\ \check{\mathcal{x}}_i\end{bmatrix} = \left(\begin{bmatrix}I_M & 0 & 0\\ 0 & I_M & 0\\ 0 & 0 & \mathcal{D}_1\end{bmatrix} - \mathcal{S}_{i-1}\right)\begin{bmatrix}\bar{\mathcal{x}}_{i-1}\\ \widehat{\mathcal{x}}_{i-1}\\ \check{\mathcal{x}}_{i-1}\end{bmatrix}, \tag{3.47}$$

where we defined

$$\begin{bmatrix}\bar{\mathcal{x}}_i\\ \widehat{\mathcal{x}}_i\\ \check{\mathcal{x}}_i\end{bmatrix} \triangleq (\mathcal{X}')^{-1}\begin{bmatrix}\widetilde{\mathcal{W}}_i\\ \widetilde{\mathcal{Y}}_i\end{bmatrix} = \begin{bmatrix}\mathcal{L}_1^{\mathsf{T}}\\ \mathcal{L}_2^{\mathsf{T}}\\ c\mathcal{X}_L\end{bmatrix}\begin{bmatrix}\widetilde{\mathcal{W}}_i\\ \widetilde{\mathcal{Y}}_i\end{bmatrix}, \tag{3.48}$$

and

$$\mathcal{S}_{i-1} \triangleq (\mathcal{X}')^{-1}\mathcal{T}_{i-1}\mathcal{X}'{=}\begin{bmatrix}\mathcal{L}_1^{\mathsf{T}}\mathcal{T}_{i-1}\mathcal{R}_1 & \mathcal{L}_1^{\mathsf{T}}\mathcal{T}_{i-1}\mathcal{R}_2 & \frac{1}{c}\mathcal{L}_1^{\mathsf{T}}\mathcal{T}_{i-1}\mathcal{X}_R\\ \mathcal{L}_2^{\mathsf{T}}\mathcal{T}_{i-1}\mathcal{R}_1 & \mathcal{L}_2^{\mathsf{T}}\mathcal{T}_{i-1}\mathcal{R}_2 & \frac{1}{c}\mathcal{L}_2^{\mathsf{T}}\mathcal{T}_{i-1}\mathcal{X}_R\\ c\mathcal{X}_L\mathcal{T}_{i-1}\mathcal{R}_1 & c\mathcal{X}_L\mathcal{T}_{i-1}\mathcal{R}_2 & \mathcal{X}_L\mathcal{T}_{i-1}\mathcal{X}_R\end{bmatrix}. \tag{3.49}$$

To evaluate the block entries of $\mathcal{S}_{i-1}$, we partition

$$\mathcal{X}_R = \begin{bmatrix}\mathcal{X}_{R,u}\\ \mathcal{X}_{R,d}\end{bmatrix}, \tag{3.50}$$

where $\mathcal{X}_{R,u} \in \mathbb{R}^{KM\times 2(K-1)M}$ and $\mathcal{X}_{R,d} \in \mathbb{R}^{KM\times 2(K-1)M}$. Then, it can be verified that

$$\mathcal{L}_1^{\mathsf{T}}\mathcal{T}_{i-1}\mathcal{R}_1 = \overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{I}, \tag{3.51}$$

$$\mathcal{L}_1^{\mathsf{T}}\mathcal{T}_{i-1}\mathcal{R}_2 = 0, \tag{3.52}$$

$$\frac{1}{c}\mathcal{L}_1^{\mathsf{T}}\mathcal{T}_{i-1}\mathcal{X}_R = \frac{1}{c}\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{X}_{R,u}. \tag{3.53}$$

While

$$\mathcal{L}_2^{\mathsf{T}}\mathcal{T}_{i-1} = \begin{bmatrix} 0 & \frac{1}{K}\mathcal{I}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \overline{\mathcal{A}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1} & 0 \\ \mathcal{V}\overline{\mathcal{A}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1} & 0 \end{bmatrix} \overset{(2.67)}{=} \begin{bmatrix} 0 & 0 \end{bmatrix}, \tag{3.54}$$

Therefore, it follows that

$$\mathcal{L}_2^{\mathsf{T}}\mathcal{T}_{i-1}\mathcal{R}_1 = 0, \;\; \mathcal{L}_2^{\mathsf{T}}\mathcal{T}_{i-1}\mathcal{R}_2 = 0, \;\; \frac{1}{c}\mathcal{L}_2^{\mathsf{T}}\mathcal{T}_{i-1}\mathcal{X}_R = 0. \tag{3.55}$$

Substituting (3.49), (3.51)–(3.53) and (3.55) into (3.47), we have

$$\begin{bmatrix} \bar{x}_i \\ \widehat{x}_i \\ \check{x}_i \end{bmatrix} = \begin{bmatrix} I_M - \overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{I} & 0 & -\frac{1}{c}\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{X}_{R,u} \\ 0 & I_M & 0 \\ -c\mathcal{X}_L\mathcal{T}_{i-1}\mathcal{R}_1 & -c\mathcal{X}_L\mathcal{T}_{i-1}\mathcal{R}_2 & \mathcal{D}_1 - \mathcal{X}_L\mathcal{T}_{i-1}\mathcal{X}_R \end{bmatrix} \begin{bmatrix} \bar{x}_{i-1} \\ \widehat{x}_{i-1} \\ \check{x}_{i-1} \end{bmatrix} \tag{3.56}$$

From the second line of (3.56), we get

$$\widehat{x}_i = \widehat{x}_{i-1}. \tag{3.57}$$

As a result, $\widehat{x}_i$ will stay at 0 only if the initial value $\widehat{x}_0 = 0$. From the definition of $\mathcal{L}_2$ in (3.40) and (3.48) we have

$$\begin{aligned} \widehat{x}_0 = \mathcal{L}_2^{\mathsf{T}} \begin{bmatrix} \widetilde{w}_0 \\ \widetilde{y}_0 \end{bmatrix} &= \frac{1}{K}\mathcal{I}^{\mathsf{T}}\widetilde{y}_0 \\ \overset{(3.14)}{=} \frac{1}{K}\mathcal{I}^{\mathsf{T}}(y_o^\star - y_0) &\overset{(2.89)}{=} \frac{1}{K}\mathcal{I}^{\mathsf{T}}(y_o^\star - \mathcal{V}w_0). \end{aligned} \tag{3.58}$$

Recall from Lemma 3.2 that $y_o^\star$ lies in the range($\mathcal{V}$), so that $y_o^\star - \mathcal{V}w_0$ also lies in range($\mathcal{V}$). From Lemma 2.4 we conclude that $\widehat{x}_0 = 0$. Therefore, from (3.57) we have

$$\widehat{x}_i = 0, \quad \forall i \geq 0 \tag{3.59}$$

With (3.59), recursion (3.56) is equivalent to

$$\boxed{\begin{bmatrix} \bar{x}_i \\ \check{x}_i \end{bmatrix} = \begin{bmatrix} I_M - \overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{I} & -\frac{1}{c}\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{X}_{R,u} \\ -c\mathcal{X}_L\mathcal{T}_{i-1}\mathcal{R}_1 & \mathcal{D}_1 - \mathcal{X}_L\mathcal{T}_{i-1}\mathcal{X}_R \end{bmatrix} \begin{bmatrix} \bar{x}_{i-1} \\ \check{x}_{i-1} \end{bmatrix}} \tag{3.60}$$

The convergence of the above recursion is stated as follows.

Theorem 3.1 (Linear Convergence) *Suppose each cost function $J_k(w)$ satisfies Assumption 3.1, the left-stochastic matrix A satisfies the local balance condition* (2.13)*, and also condition* (2.9) *holds. The exact diffusion recursion* (2.88) *converges exponentially fast to $(\mathcal{W}^\star, \mathcal{Y}_o^\star)$ for step-sizes satisfying*

$$\mu_{\max} \leq \frac{p_{k_o}\tau_{k_o}\nu(1-\lambda)}{2\sqrt{p_{\max}}\alpha_d\delta^2}, \tag{3.61}$$

where $\lambda{=}\sqrt{\lambda_2(\overline{A})}{<}1$, $\tau_{k_o}{=}\mu_{k_o}/\mu_{\max}$, $p_{\max}{=}\max_k\{p_k\}$ and

$$\alpha_d \triangleq \|\mathcal{X}_L\|\|\mathcal{T}_d\|\|\mathcal{X}_R\|, \text{ where } \mathcal{T}_d \triangleq \begin{bmatrix} \overline{\mathcal{A}}^{\mathsf{T}} & 0 \\ \mathcal{V}\overline{\mathcal{A}}^{\mathsf{T}} & 0 \end{bmatrix}. \tag{3.62}$$

The convergence rate for the error variables is given by

$$\left\| \begin{bmatrix} \widetilde{\mathcal{W}}_i \\ \widetilde{\mathcal{Y}}_i \end{bmatrix} \right\|^2 \leq C\rho^i, \tag{3.63}$$

where C is some constant and $\rho = 1 - O(\mu_{\max})$, namely,

$$\begin{aligned} \rho = \max\Big\{ & 1 - p_{k_o}\tau_{k_o}\nu\mu_{\max} + \frac{2\sqrt{p_{\max}}\alpha_d\delta^2\mu_{\max}^2}{1-\lambda}, \\ & \lambda + \frac{\sqrt{p_{\max}}\alpha_d\delta^2\mu_{\max}}{p_{k_o}\tau_{k_o}\nu} + \frac{2\alpha_d^2\delta^2\mu_{\max}^2}{1-\lambda} \Big\} < 1. \end{aligned} \tag{3.64}$$

Proof. See Appendix 3.B. ■

With similar arguments shown above, we can also establish the convergence property of the exact diffusion algorithm 2.2 in Chapter 2. Compared to the above convergence analysis, the error dynamics for algorithm 1' will now be perturbed by a mismatch term caused by the power iteration. Nevertheless, once the analysis is carried out we arrive at a similar conclusion.

Theorem 3.2 (Linear convergence of Algorithm 2.2) *Under the conditions of Theorem 3.1, there exists a positive constant $\bar{\mu} > 0$ such that for step-sizes satisfying $\mu < \bar{\mu}$, the exact diffusion Algorithm 2.2 will converge exponentially fast to $(\mathcal{W}^\star, \mathcal{Y}_o^\star)$.*

Proof. See Appendix 3.C. ■

3.2 Stability Comparison with EXTRA

3.2.1 Stability Range of EXTRA

In the case where the combination matrix A is symmetric and *doubly-stochastic*, and all agents choose the *same* step-size μ, the exact diffusion recursion (2.88) reduces to

$$\begin{cases} \mathcal{W}_i = \overline{\mathcal{A}}\Big(\mathcal{W}_{i-1} - \mu\nabla\mathcal{J}^o(\mathcal{W}_{i-1})\Big) - \mathcal{P}^{-1}\mathcal{V}\mathcal{Y}_{i-1}, \\ \mathcal{Y}_i = \mathcal{Y}_{i-1} + \mathcal{V}\mathcal{W}_i. \end{cases} \tag{3.65}$$

where $\mathcal{P} = I_{MK}/K$. In comparison, the EXTRA consensus algorithm [75] has the following form for the same $\mathcal{P}$ (recall though that exact diffusion (2.88) was derived and is applicable to a larger class of balanced left-stochastic matrices and is not limited to symmetric doubly stochastic matrices; it also allows for heterogeneous step-sizes):

$$\begin{cases} \mathcal{W}_i^e = \overline{\mathcal{A}}\mathcal{W}_{i-1}^e - \mu\nabla\mathcal{J}^o(\mathcal{W}_{i-1}^e) - \mathcal{P}^{-1}\mathcal{V}\mathcal{Y}_{i-1}^e, \\ \mathcal{Y}_i^e = \mathcal{Y}_{i-1}^e + \mathcal{V}\mathcal{W}_i^e, \end{cases} \tag{3.66}$$

where we are using the notation $\mathcal{W}_i^e$ and $\mathcal{Y}_i^e$ to refer to the primal and dual iterates in the EXTRA implementation. Similar to (2.89), the initial condition for (3.66) is

$$\begin{cases} \mathcal{W}_0^e = \overline{\mathcal{A}}\mathcal{W}_{-1}^e - \mu\nabla\mathcal{J}^o(\mathcal{W}_{-1}^e), \\ \mathcal{Y}_0^e = \mathcal{V}\mathcal{W}_0^e. \end{cases} \tag{3.67}$$

Comparing (3.65) and (3.66) we observe one key difference; the diffusion update in (3.65) involves a traditional gradient descent step in the form of $\mathcal{W}_{i-1} - \mu\nabla\mathcal{J}^o(\mathcal{W}_{i-1})$. This step starts from $\mathcal{W}_{i-1}$ and evaluates the graduate vector at the same location. The result is then multiplied by the combination policy $\widetilde{\mathcal{A}}$. The same is *not* true for exact consensus in (3.66); we observe an asymmetry in its update: the gradient vector is evaluated at $\mathcal{W}_{i-1}^e$ while the starting point is at a different location given by $\widetilde{\mathcal{A}}\mathcal{W}_{i-1}^e$. This type of asymmetry was shown in [1, 4] to result in instabilities for the traditional consensus implementation in comparison to the traditional diffusion implementation. It turns out that a similar problem continues to exist for the EXTRA consensus solution (3.66). In particular, we will show that its stability range is smaller than exact diffusion (i.e., the latter is stable for a larger range of step-sizes,

which in turn helps attain faster convergence rates). We will illustrate this behavior in the simulations in some detail. Here, though, we establish these observations analytically. The arguments used to examine the stability range of EXTRA consensus are similar to what we did in Sec. 3.1 for exact diffusion; we shall therefore be brief and highlight only the differences.

As already noted in [75], the optimality conditions for the EXTRA consensus algorithm require the existence of block vectors $(\mathcal{w}^\star, \mathcal{y}^\star)$ such that

$$\mu\nabla\mathcal{J}^o(\mathcal{w}^\star) + \mathcal{P}^{-1}\mathcal{V}\mathcal{y}^\star = 0, \tag{3.68}$$

$$\mathcal{V}\mathcal{w}^\star = 0. \tag{3.69}$$

Moreover, as argued in Lemma 3.2, there also exists a unique pair of variables $(\mathcal{w}^\star, \mathcal{y}_o^\star)$, in which $\mathcal{y}_o^\star$ lies in the range space of $\mathcal{V}$, that satisfies (3.68)–(3.69). Now we introduce the block error vectors:

$$\widetilde{\mathcal{w}}_i^e = \mathcal{w}^\star - \mathcal{w}_i^e, \quad \widetilde{\mathcal{y}}_i^e = \mathcal{y}_o^\star - \mathcal{y}_i^e, \tag{3.70}$$

and examine the evolution of these error quantities. Using similar arguments in Section 3.1.2, and recalling the facts that $\overline{\mathcal{A}}$ is symmetric doubly-stochastic, and $\mathcal{M} = \mu I_{MK}$, we arrive at the error recursion for EXTRA consensus (see Appendix 3.D):

$$\begin{aligned}\begin{bmatrix} \widetilde{\mathcal{w}}_i^e \\ \widetilde{\mathcal{y}}_i^e \end{bmatrix} &= \begin{bmatrix} \overline{\mathcal{A}} - \mu\mathcal{H}_{i-1} & -\mathcal{P}^{-1}\mathcal{V} \\ \mathcal{V}(\overline{\mathcal{A}} - \mu\mathcal{H}_{i-1}) & I_{MK} - \mathcal{V}\mathcal{P}^{-1}\mathcal{V} \end{bmatrix} \begin{bmatrix} \widetilde{\mathcal{w}}_{i-1}^e \\ \widetilde{\mathcal{y}}_{i-1}^e \end{bmatrix} \\ &\triangleq (\mathcal{B}^e - \mathcal{T}_{i-1}^e) \begin{bmatrix} \widetilde{\mathcal{w}}_{i-1}^e \\ \widetilde{\mathcal{y}}_{i-1}^e \end{bmatrix},\end{aligned} \tag{3.71}$$

where

$$\mathcal{B}^e \triangleq \begin{bmatrix} \overline{\mathcal{A}} & -\mathcal{P}^{-1}\mathcal{V} \\ \mathcal{V}\overline{\mathcal{A}} & I_{MK} - \mathcal{V}\mathcal{P}^{-1}\mathcal{V} \end{bmatrix}, \mathcal{T}_i^e \triangleq \begin{bmatrix} \mu\mathcal{H}_i & 0 \\ \mu\mathcal{V}\mathcal{H}_i & 0 \end{bmatrix}. \tag{3.72}$$

It is instructive to compare (3.71)–(3.72) with (3.26)–(3.28). These recursions capture the error dynamics for the exact consensus and diffusion strategies. Observe that $\mathcal{B}^e = \mathcal{B}$ when $\overline{\mathcal{A}}$ is symmetric and $\mathcal{M} = \mu I_{MK}$. Therefore, $\mathcal{B}^e$ has the same eigenvalue decomposition as in

(3.42)–(3.45). With similar arguments to (3.33)–(3.60), we conclude that the reduced error recur-sion for EXTRA consensus takes the form (see Appendix 3.E):

$$\begin{bmatrix} \bar{x}_i^e \\ \check{x}_i^e \end{bmatrix} = \begin{bmatrix} I_M - \mu\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{I} & -\frac{\mu}{c}\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{X}_{R,u} \\ -c\mathcal{X}_L\mathcal{T}_{i-1}^e\mathcal{R}_1 & \mathcal{D}_1 - \mathcal{X}_L\mathcal{T}_{i-1}^e\mathcal{X}_R \end{bmatrix} \begin{bmatrix} \bar{x}_{i-1}^e \\ \check{x}_{i-1}^e \end{bmatrix}. \tag{3.73}$$

Following the same proof technique as for Theorem 3.1, we can now establish the following result concerning stability conditions and convergence rate for EXTRA consensus.

Theorem 3.3 (Linear Convergence of EXTRA) *Suppose each cost function $J_k(w)$ satisfies Assumption 3.1, and the combination matrix A is primitive, symmetric and doubly-stochastic. The EXTRA recursion* (3.71) *converges exponentially fast to $(\mathcal{w}^\star, \mathcal{y}_o^\star)$ for step-sizes μ satisfying*

$$\mu \leq \frac{\nu(1-\lambda)}{2\sqrt{K}\alpha_e\delta^2}, \tag{3.74}$$

where $\lambda = \sqrt{\lambda_2(\overline{A})} < 1$ and

$$\alpha_e = \|\mathcal{X}_L\|\|\mathcal{T}_e\|\|\mathcal{X}_R\|, \text{ where } \mathcal{T}_e = \begin{bmatrix} I_{MK} & 0 \\ \mathcal{V} & 0 \end{bmatrix}. \tag{3.75}$$

The convergence rate for the error variables is given by

$$\left\| \begin{bmatrix} \widetilde{\mathcal{w}}_i^e \\ \widetilde{\mathcal{y}}_i^e \end{bmatrix} \right\|^2 \leq C\rho^i, \tag{3.76}$$

where C is some constant and $\rho = 1 - O(\mu_{\max})$, namely,

$$\begin{aligned} \rho_e = \max\Big\{ & 1 - \frac{\nu}{K}\mu_{\max} + \frac{2\alpha_e\delta^2\mu_{\max}^2}{\sqrt{K}(1-\lambda)}, \\ & \lambda + \frac{\sqrt{K}\alpha_e\delta^2\mu_{\max}}{\nu} + \frac{2\alpha_e^2\delta^2\mu_{\max}^2}{1-\lambda} \Big\} < 1. \end{aligned} \tag{3.77}$$

Proof. See Appendix 3.F. ■

3.2.2 Comparison of Stability Ranges

When $\overline{\mathcal{A}}$ is symmetric and $\mathcal{M} = \mu I_{MK}$, from Theorem 3.1 we get the stability range of exact diffusion:

$$\mu \leq \frac{\nu(1-\lambda)}{2\sqrt{K}\|\mathcal{X}_L\|\|\mathcal{T}_d\|\|\mathcal{X}_R\|\delta^2}, \tag{3.78}$$

where

$$\mathcal{T}_d = \begin{bmatrix} \overline{\mathcal{A}} & 0 \\ \mathcal{V}\overline{\mathcal{A}} & 0 \end{bmatrix}. \tag{3.79}$$

Comparing (3.78) with (3.74), we observe that the expressions differ by the terms $\|\mathcal{T}_e\|$ and $\|\mathcal{T}_d\|$. We therefore need to compare these two norms.

Notice that

$$\|\mathcal{T}_e\|^2 = \lambda_{\max}(\mathcal{T}_e^{\mathsf{T}}\mathcal{T}_e) = \lambda_{\max}(I_{MK} + \mathcal{V}^2), \tag{3.80}$$

$$\|\mathcal{T}_d\|^2 = \lambda_{\max}(\mathcal{T}_d^{\mathsf{T}}\mathcal{T}_d) = \lambda_{\max}\big(\overline{\mathcal{A}}(I_{MK} + \mathcal{V}^2)\overline{\mathcal{A}}\big). \tag{3.81}$$

It is easy to recognize that $\lambda_{\max}(I_{MK} + \mathcal{V}^2) = \lambda_{\max}(I_K + V^2)$. Now, since A is assumed symmetric doubly-stochastic and $P = \frac{1}{K}I_K$, we have

$$\begin{aligned} I_K + V^2 &= I_K + \frac{P - PA}{2} \\ &= I_K + \frac{I_K - A}{2K} = \frac{(2K+1)I_K - A}{2K}, \end{aligned} \tag{3.82}$$

Moreover, since A is primitive, symmetric and doubly stochastic, we can decompose it as

$$A = U\Lambda U^{\mathsf{T}}, \tag{3.83}$$

where U is orthogonal, $\Lambda = \mathrm{diag}\{\lambda_1(A), \cdots, \lambda_K(A)\}$ and

$$1 = \lambda_1(A) > \lambda_2(A) \geq \cdots \geq \lambda_K(A) > -1. \tag{3.84}$$

With this decomposition, expression (3.82) can be rewritten as

$$I_K + V^2 = U\frac{(2K+1)I_K - \Lambda}{2K}U^{\mathsf{T}}. \tag{3.85}$$

from which we conclude that

$$\boxed{\lambda_{\max}(I_K + V^2) = \frac{(2K+1) - \lambda_K(A)}{2K}} \tag{3.86}$$

Similarly, $\lambda_{\max}(\overline{\mathcal{A}}(I_{MK} + \mathcal{V}^2)\overline{\mathcal{A}}) = \lambda_{\max}(\overline{A}(I_K + V^2)\overline{A})$. Using $\overline{A} = \frac{I_K + A}{2}$, and equations (3.83) and (3.85), we have

$$\begin{aligned} &\overline{A}(I_K + V^2)\overline{A} \\ &= \left(\frac{I_K + A}{2}\right)\left(\frac{(2K+1)I_K - A}{2K}\right)\left(\frac{I_K + A}{2}\right) \end{aligned} \tag{3.87}$$

$$= U\left(\frac{I_K + \Lambda}{2}\right)\left(\frac{(2K+1)I_K - \Lambda}{2K}\right)\left(\frac{I_K + \Lambda}{2}\right)U^{\mathsf{T}}. \tag{3.88}$$

Therefore, we have

$$\begin{aligned} &\lambda_{\max}\left(\overline{A}(I_K + V^2)\overline{A}\right) \\ &= \max_k \left\{\left(\frac{\lambda_k(A)+1}{2}\right)^2 \left(\frac{2K+1-\lambda_k(A)}{2K}\right)\right\} \\ &\overset{(a)}{\leq} \max_k \left\{\left(\frac{\lambda_k(A)+1}{2}\right)^2\right\} \max_k \left\{\frac{2K+1-\lambda_k(A)}{2K}\right\} \\ &\overset{(3.84)}{=} \frac{2K+1-\lambda_K(A)}{2K}. \end{aligned} \tag{3.89}$$

It is worth noting that the "=" sign cannot hold in (a) because

$$\arg\max_k \left\{\left(\frac{\lambda_k(A)+1}{2}\right)^2\right\} = 1, \tag{3.90}$$

$$\arg\max_k \left\{\frac{2K+1-\lambda_k(A)}{2K}\right\} = N. \tag{3.91}$$

In other words, $\left(\frac{\lambda_k(A)+1}{2}\right)^2$ and $\frac{2K+1-\lambda_k(A)}{2K}$ cannot reach their maximum values at the same k. As a result,

$$\|\mathcal{T}_d\|^2 < \|\mathcal{T}_e\|^2 \Longrightarrow \alpha_d < \alpha_e. \tag{3.92}$$

This means that the upper bound on μ in (3.74) is smaller than the upper bound on μ in (3.78).

We can also compare the convergence rates of EXTRA consensus and exact diffusion when both algorithms converge. When $\overline{\mathcal{A}}$ is symmetric and $\mathcal{M} = \mu I_{MK}$, from Theorem 3.1 we get the convergence rate of exact diffusion:

$$\rho_d = \max\Big\{1 - \frac{\nu}{K}\mu_{\max} + \frac{2\alpha_d\delta^2\mu_{\max}^2}{\sqrt{K}(1-\lambda)}, \lambda + \frac{\sqrt{K}\alpha_d\delta^2\mu_{\max}}{\nu} + \frac{2\alpha_d^2\delta^2\mu_{\max}^2}{1-\lambda}\Big\}. \tag{3.93}$$

It is clear from (3.93) and (3.77) that EXTRA consensus and exact diffusion have the same convergence rate to first-order in $\mu_{\max}$, namely,

$$\widehat{\rho}_d = 1 - \frac{\nu}{K}\mu_{\max} = \widehat{\rho}_e \tag{3.94}$$

More generally, when higher-order terms in $\mu_{\max}$ cannot be ignored, it holds that $\rho_d < \rho_e$ because $\alpha_d < \alpha_e$ (see (3.92)). In this situation, exact diffusion converges faster than EXTRA.

3.2.3 An Analytical Example

In this subsection we illustrate the stability of exact diffusion by considering the example of mean-square-error (MSE) networks [1]. Suppose K agents are observing streaming data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ that satisfy the regression model

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}^{\mathsf{T}} w^o + \boldsymbol{v}_k(i), \tag{3.95}$$

where w^o is unknown and $\boldsymbol{v}_k(i)$ is the noise process that is independent of the regression data $\boldsymbol{u}_{k,j}$ for any k, j. Furthermore, we assume $\boldsymbol{u}_{k,i}$ is zero-mean with covariance matrix $R_{u,k} = \mathbb{E}\boldsymbol{u}_{k,i}\boldsymbol{u}_{k,i}^{\mathsf{T}} > 0$, and $\boldsymbol{v}_k(i)$ is also zero-mean with power $\sigma_{v,k}^2 = \mathbb{E}\boldsymbol{v}_k^2(i)$. We denote the cross covariance vector between $\boldsymbol{d}_k(i)$ and $\boldsymbol{u}_{k,i}$ by $r_{du,k} = \mathbb{E}\boldsymbol{d}_k(i)\boldsymbol{u}_{k,i}$. To discover the unknown w^o, the agents cooperate to solve the following mean-square-error problem:

$$\min_{w\in\mathbb{R}^M} \ \tfrac{1}{2}\sum_{k=1}^K \mathbb{E}\big(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}^{\mathsf{T}} w\big)^2. \tag{3.96}$$

It was shown in Example 6.1 of [1] that the global minimizer of problem (3.96) coincides with the unknown w^o in (3.95).

When $R_{u,k}$ and $r_{du,k}$ are unknown and only realizations of $\boldsymbol{u}_{k,i}$ and $\boldsymbol{d}_k(i)$ are observed by agent k, one can employ the diffusion algorithm with stochastic gradient descent to solve

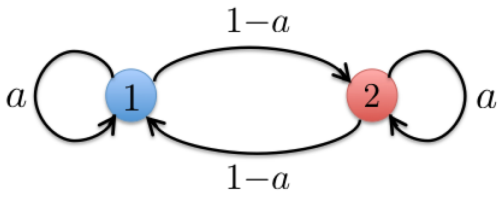

Figure 3.1: A two-agent network using combination weights $\{a, 1-a\}$

(3.96). However, when $R_{u,k}$ and $r_{du,k}$ are known in advance, problem (3.96) reduces to deterministic optimization problem:

$$\min_{w\in\mathbb{R}^M} \ \frac{1}{2}\sum_{k=1}^{K}\big(w^{\mathsf{T}}R_{u,k}w - 2r_{du,k}^{\mathsf{T}}w\big). \tag{3.97}$$

We can then employ the exact diffusion or the EXTRA consensus algorithm to solve (3.97).

To illustrate the stability issue, it is sufficient to consider a network with 2 agents (see Fig. 3.1) and with diagonal Hessian matrices, i.e.,

$$R_{u,1} = R_{u,2} = \sigma^2 I_M. \tag{3.98}$$

We assume the agents use the combination weights $\{a, 1-a\}$ with $a \in (0,1)$, so that

$$A = \begin{bmatrix} a & 1-a \\ 1-a & a \end{bmatrix} \in \mathbb{R}^{2\times 2}, \tag{3.99}$$

which is symmetric and doubly stochastic. The two agents employ the same step-size μ (or μ^e in the EXTRA recursion). It is worth noting that the following analysis can be extended to K agents with some more algebra.

Under (3.98), we have $H_1 = H_2 = \sigma^2 I_M$ and $\mathcal{H} = \mathrm{diag}\{H_1, H_2\} = \sigma^2 I_{2M}$. For the matrix A in (3.99), we have

$$\lambda_1(A) = 1, \quad \lambda_2(A) = 2a - 1 \in (-1, 1), \tag{3.100}$$

and $p = [0.5; 0.5]$, $P = 0.5 I_2$.

Let $\widetilde{z}_i = [\widetilde{w}_i; \widetilde{y}_i] \in \mathbb{R}^{2M}$, and $\widetilde{z}_i^e = [\widetilde{w}_i^e; \widetilde{y}_i^e] \in \mathbb{R}^{2M}$. The exact diffusion error recursion (3.26) and the EXTRA error recursion (3.71) reduce to

$$\widetilde{z}_i = \mathcal{Q}_d \widetilde{z}_{i-1}, \tag{3.101}$$

$$\widetilde{z}_i^e = \mathcal{Q}_e \widetilde{z}_{i-1}^e, \tag{3.102}$$

where

$$\mathcal{Q}_d = \underbrace{\begin{bmatrix} (1-\mu\sigma^2)\overline{A} & -2V \\ (1-\mu\sigma^2)V\overline{A} & \overline{A} \end{bmatrix}}_{Q_d} \otimes I_M, \tag{3.103}$$

$$\mathcal{Q}_e = \underbrace{\begin{bmatrix} \overline{A} - \mu^e\sigma^2 I_2 & -2V \\ V(\overline{A} - \mu^e\sigma^2 I_2) & \overline{A} \end{bmatrix}}_{Q_e} \otimes I_M. \tag{3.104}$$

To guarantee the convergence of $\widetilde{z}_i$ and $\widetilde{z}_i^e$, we need to examine the eigenstructure of the 4×4 matrices Q_d and Q_e. The proof of the next lemma is quite similar to Lemma 3.3; if desired, see Appendix F of the arXiv version [16].

Lemma 3.4 (Eigenstructure of Q_d) *The matrix Q_d admits the following eigendecomposition*

$$Q_d = X\overline{Q}_d X^{-1}, \tag{3.105}$$

where

$$\overline{Q}_d = \begin{bmatrix} 1 & 0 \\ 0 & E_d \end{bmatrix} \tag{3.106}$$

and

$$E_d = \begin{bmatrix} 1-\mu\sigma^2 & 0 & 0 \\ 0 & (1-\mu\sigma^2)a & -\sqrt{2-2a} \\ 0 & (1-\mu\sigma^2)a\sqrt{\frac{1-a}{2}} & a \end{bmatrix}. \tag{3.107}$$

Moreover, the matrices X and X^{-1} are given by

$$X = \begin{bmatrix} r & X_R \end{bmatrix}, \quad X^{-1} = \begin{bmatrix} \ell^{\mathsf{T}} \\ X_L \end{bmatrix}, \tag{3.108}$$

where $X_R \in \mathbb{R}^{4\times 3}$, $X_L \in \mathbb{R}^{3\times 4}$, and

$$r = \frac{1}{2}\begin{bmatrix} 0 \\ \mathbb{1}_2 \end{bmatrix} \in \mathbb{R}^4, \quad \ell = \begin{bmatrix} 0 \\ \mathbb{1}_2 \end{bmatrix} \in \mathbb{R}^4. \tag{3.109}$$

■

It is observed that Q_d always has an eigenvalue at 1, which implies that Q_d is not stable no matter what the step-size μ is. However, this eigenvalue does not influence the convergence of recursions (3.101). To see that, from Lemma 3.4 we have

$$\mathcal{Q}_d = \mathcal{X}\overline{\mathcal{Q}}_d\mathcal{X}^{-1} = \begin{bmatrix} R & \mathcal{X}_R \end{bmatrix} \begin{bmatrix} I_M & 0 \\ 0 & \mathcal{E}_d \end{bmatrix} \begin{bmatrix} L^{\mathsf{T}} \\ \mathcal{X}_L \end{bmatrix} \tag{3.110}$$

where $\mathcal{X}_R = X_R \otimes I_M$, $\mathcal{X}_L = X_L \otimes I_M$, $\mathcal{E}_d = E_d \otimes I_M$, and

$$R = \frac{1}{2}\begin{bmatrix} 0 \\ \mathbb{1}_2 \otimes I_M \end{bmatrix}, \quad L = \begin{bmatrix} 0 \\ \mathbb{1}_2 \otimes I_M \end{bmatrix}. \tag{3.111}$$

Let

$$\begin{bmatrix} \widehat{z}_i \\ \check{z}_i \end{bmatrix} = \mathcal{X}^{-1}\widetilde{z}_i = \begin{bmatrix} L^{\mathsf{T}}\widetilde{z}_i \\ \mathcal{X}_L\widetilde{z}_i \end{bmatrix}. \tag{3.112}$$

The exact diffusion recursion (3.101) can be transformed into

$$\begin{bmatrix} \widehat{z}_i \\ \check{z}_i \end{bmatrix} = \begin{bmatrix} I_M & 0 \\ 0 & \mathcal{E}_d \end{bmatrix} \begin{bmatrix} \widehat{z}_{i-1} \\ \check{z}_{i-1} \end{bmatrix}, \tag{3.113}$$

which can be further divided into two separate recursions:

$$\widehat{z}_i = \widehat{z}_{i-1}, \quad \check{z}_i = \mathcal{E}_d\check{z}_{i-1}. \tag{3.114}$$

Therefore, $\widehat{z}_i = 0$ if $\widehat{z}_0 = 0$. Since $\mathcal{y}_0 = \mathcal{V}\mathcal{w}_0$ and $\mathcal{y}_o^\star \in \text{range}(\mathcal{V})$, we have $\widetilde{\mathcal{y}}_0 = \mathcal{y}_o^\star - \mathcal{y}_0 \in \text{range}(\mathcal{V})$. Therefore,

$$\widehat{z}_0 \overset{(3.112)}{=} L^{\mathsf{T}}\widetilde{z}_0 = \begin{bmatrix} 0 & (\mathbb{1}_2 \otimes I_M)^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \widetilde{\mathcal{w}}_0 \\ \widetilde{\mathcal{y}}_0 \end{bmatrix} \overset{(2.67)}{=} 0, \tag{3.115}$$

As a result, we only need to focus on the other recursion:

$$\check{z}_i = \mathcal{E}_d\check{z}_{i-1}, \quad \text{where} \quad \mathcal{E}_d = E_d \otimes I_M. \tag{3.116}$$

If we select the step-size μ such that all eigenvalues of E_d stay inside the unit-circle, then we guarantee the convergence of $\check{z}_i$ and, hence, $\widetilde{z}_i$.

Lemma 3.5 (Stability of exact diffusion) *When μ is chosen such that*

$$0 < \mu\sigma^2 < 2, \tag{3.117}$$

all eigenvalues of E_d will lie inside the unit-circle, which implies that $\widetilde{z}_i$ in (3.101) *converges to* 0*, i.e., $\widetilde{z}_i \to 0$.*

Proof. See Appendix 3.G. Next we turn to the EXTRA error recursion (3.102). ■

Lemma 3.6 (Instability of EXTRA) *When μ^e is chosen such that*

$$\mu^e\sigma^2 \geq a + 1, \tag{3.118}$$

it holds that $\widetilde{z}_i^e$ generated through EXTRA (3.102) *will diverge.*

Proof. See Appendix 3.H. ■

Comparing the statements of Lemmas 3.5 and 3.6, and since $1 + a < 2$, exact diffusion has a larger range of stability than EXTRA (i.e., exact diffusion is stable for a wider range of step-size values). In particular, if agents place small weights on their own data, i.e., when $a \approx 0$, the stability range for exact diffusion will be almost twice as large as that of EXTRA.

3.3 Numerical Experiments

In this section we compare the performance of the proposed exact diffusion algorithm with existing linearly convergent algorithms such as EXTRA [75], DIGing [93], and Aug-DGM [95, 96]. In all figures, the y-axis indicates the relative error, i.e., $\|\mathcal{w}_i - \mathcal{w}^o\|^2/\|\mathcal{w}_0 - \mathcal{w}^o\|^2$, where $\mathcal{w}_i = \mathrm{col}\{w_{1,i}, \cdots, w_{K,i}\} \in \mathbb{R}^{KM}$ and $\mathcal{w}^o = \mathrm{col}\{w^o, \cdots, w^o\} \in \mathbb{R}^{KM}$. All simulations employ the connected network topology with $N = 20$ nodes shown in Fig. 2.3 in Chapter 2.

3.3.1 Distributed Least-squares

In this experiment, we focus on the least-squares problem:

$$w^o = \arg\min_{w\in\mathbb{R}^M} \quad \frac{1}{2}\sum_{k=1}^{K} \|U_k w - d_k\|^2. \tag{3.119}$$

The simulation setting is the same as Sec. 2.6 in Chapter 2.

In the simulation we compare exact diffusion with EXTRA, DIGing, and Aug-DGM. These algorithms work with symmetric doubly-stochastic or right-stochastic matrices A. Therefore, we now employ doubly-stochastic matrices for a proper comparison. Moreover, there are two information combinations per iteration in DIGing and Aug-DGM algorithms, and each information combination corresponds to one round of communication. In comparison, there is only one information combination (or round of communication) in EXTRA and exact diffusion. For fairness we will compare the algorithms based on the amount of communications, rather than the iterations. In the figures, we use one unit amount of communication to represent $2ME$ communicated variables, where M is the dimension of the variable while E is the number of edges in the network. The problem setting is the same as in the simulations in Chapter 2, except that A is generated through the Metropolis rule [1]. In the top plot in Fig. 3.2, all algorithms are carefully adjusted to reach their fastest convergence. It is observed that exact diffusion is slightly better than EXTRA, and both of them are more communication efficient than DIGing and Aug-DGM. When a larger step-size $\mu = 0.02$ is chosen for all algorithms, it is observed that EXTRA and DIGing diverge while exact diffusion and Aug-DGM converge, and exact diffusion is much faster than Aug-DGM algorithm.

We also compare exact diffusion with Push-EXTRA [100, 123] and Push-DIGing [93] for non-symmetric combination policies. We consider the unbalanced network topology shown in Fig. 2.3 in chapter 2. The combination matrix is generated through the averaging rule. Note that the Perron eigenvector p is known beforehand for such combination matrix A, and we can therefore substitute p directly into the recursions of Push-EXTRA and Push-DIGing. In the simulation, all algorithms are adjusted to reach their fastest convergence. In Fig. 3.3, it is observed that exact diffusion is the most communication efficient among all three algorithms. This figure illustrates that exact diffusion has superior performance for locally-balanced combination policies.

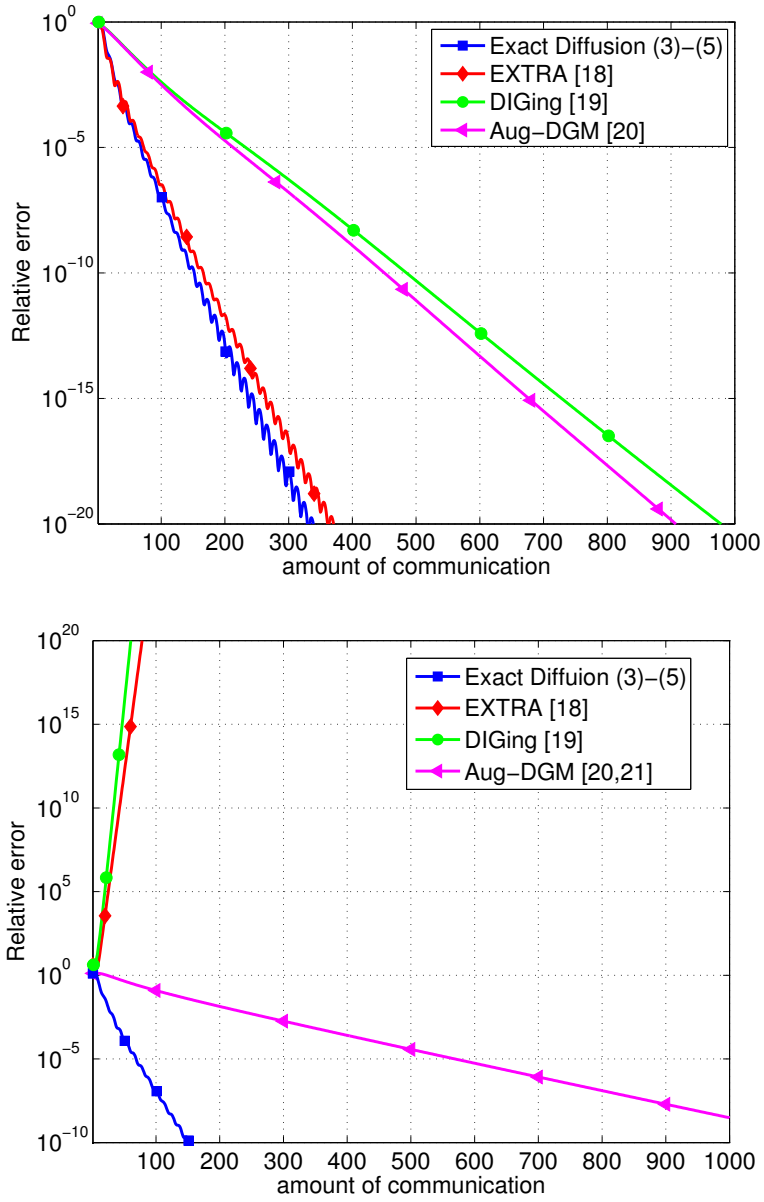

Figure 3.2: Convergence comparison between exact diffusion, EXTRA, DIGing, and Aug-DGM for distributed least-squares problem (3.119). In the top plot, the step-sizes for Exact diffusion, EXTRA, DIGing and Aug-DGM are 0.013, 0.007, 0.0028 and 0.003. In the bottom plot, all step-sizes are set as 0.04.

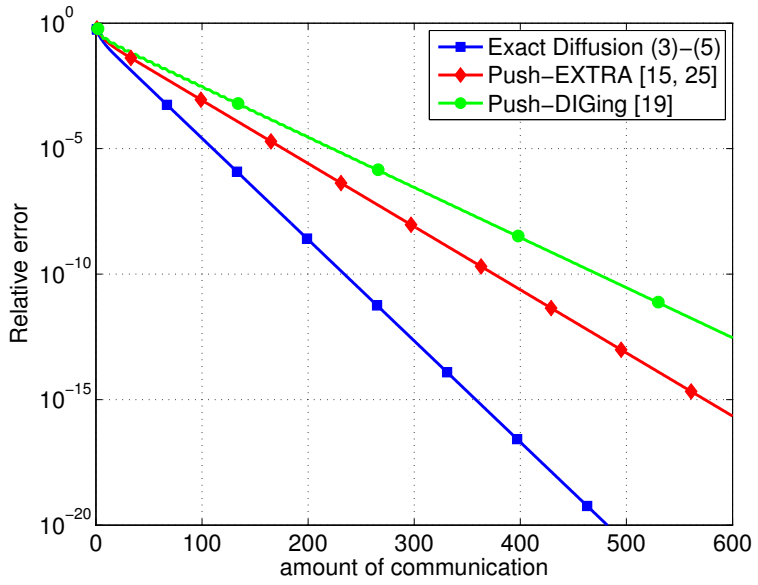

Figure 3.3: Convergence comparison between exact diffusion, EXTRA, DIGing, and Aug-DGM for distributed least-squares problem (3.119) with non-symmetric combination policy.

3.3.2 Distributed Logistic Regression

We next consider a pattern classification scenario. Each agent k holds local data samples $\{h_{k,j}, \gamma_{k,j}\}_{j=1}^{L}$, where $h_{k,j} \in \mathbb{R}^M$ is a feature vector and $\gamma_{k,j} \in \{-1, +1\}$ is the corresponding label. Moreover, the value L is the number of local samples at each agent. All agents will cooperatively solve the regularized logistic regression problem:

$$w^o = \arg\min_{w \in \mathbb{R}^M} \sum_{k=1}^{K} \Big[\frac{1}{L} \sum_{\ell=1}^{L} \ln\big(1+\exp(-\gamma_{k,\ell} h_{k,\ell}^{\mathsf{T}} w)\big) + \frac{\rho}{2}\|w\|^2 \Big]. \tag{3.120}$$

The simulation setting is the same as Sec. 2.6 in Chapter 2.

In this simulation, we also compare exact diffusion with EXTRA, DIGing, and Aug-DGM. A symmetric doubly-stochastic A is generated through the Metropolis rule. In the top plot in Fig. 3.4, all algorithms are carefully adjusted to reach their fastest convergence. It is observed that exact diffusion is the most communication efficient among all algorithms. When a larger step-size $\mu = 0.04$ is chosen for all algorithms in the bottom plot in Fig. 3.4, it is observed that both exact diffusion and Aug-DGM are still able to converge linearly to w^o, while EXTRA and DIGing fail to do so. Moreover, exact diffusion is observed much

more communication efficient than Aug-DGM.

3.A Proof of Lemma 3.3

Define $V' \triangleq V + \mathbb{1}_K p^\mathsf{T} \in \mathbb{R}^{K\times K}$, we claim that V' is a full rank matrix. Suppose to the contrary that there exists some $x \neq 0$ such that $V'x = 0$, i.e.,$(V+\mathbb{1}_K p^\mathsf{T})x = Vx+(p^\mathsf{T}x)\mathbb{1}_K = 0$, which requires

$$Vx = -(p^\mathsf{T}x)\mathbb{1}_K. \tag{3.121}$$

When $p^\mathsf{T}x \neq 0$, relation (3.121) implies that $\mathbb{1}_K \in \text{range}(V)$. However, from Lemma 2.4 we know that

$$\begin{aligned}\text{null}(V) = \text{span}\{\mathbb{1}_K\} &\iff \text{range}(V^\mathsf{T})^\perp = \text{span}\{\mathbb{1}_K\}\\ &\iff \text{range}(V)^\perp = \text{span}\{\mathbb{1}_K\},\end{aligned} \tag{3.122}$$

where the last "$\Leftrightarrow$" holds because V is symmetric. Relation (3.122) is contradictory to $\mathbb{1}_K \in \text{range}(V)$. Therefore, $V'x \neq 0$. When $p^\mathsf{T}x = 0$, relation (3.121) implies that $Vx = 0$, which together with Lemma 2.4 implies that $x = c\mathbb{1}_K$ for some constant $c \neq 0$. However, since $p^\mathsf{T}\mathbb{1}_K = 1$, we have $p^\mathsf{T}x = c \neq 0$, which also contradicts with $p^\mathsf{T}x = 0$. As a result, V' has full rank and hence $(V')^{-1}$ exists.

With $V' = V + \mathbb{1}_K$ and the fact $V\mathbb{1}_K = 0$ (see Lemma 2.4), we also have

$$VV' = V(V + \mathbb{1}_K p^\mathsf{T}) = V^2 + V\mathbb{1}_K p^\mathsf{T} = V^2, \tag{3.123}$$

$$V'(I_K - \mathbb{1}_K p^\mathsf{T}) = (V + \mathbb{1}_K p^\mathsf{T})(I_K - \mathbb{1}_K p^\mathsf{T}) = V. \tag{3.124}$$

With relations (3.123) and (3.124), we can verify that

$$\begin{aligned}B &= \begin{bmatrix} I_K & 0 \\ 0 & V' \end{bmatrix}\begin{bmatrix} \overline{A}^\mathsf{T} & -P^{-1}V^2 \\ (V')^{-1}V\overline{A}^\mathsf{T} & I_K - (V')^{-1}VP^{-1}V^2 \end{bmatrix}\begin{bmatrix} I_K & 0 \\ 0 & (V')^{-1} \end{bmatrix}\\ &\overset{(a)}{=} \begin{bmatrix} I_K & 0 \\ 0 & V' \end{bmatrix}\begin{bmatrix} \overline{A}^\mathsf{T} & \overline{A}^\mathsf{T} - I_K \\ \overline{A}^\mathsf{T} - \mathbb{1}_K p^\mathsf{T} & \overline{A}^\mathsf{T} \end{bmatrix}\begin{bmatrix} I_K & 0 \\ 0 & (V')^{-1} \end{bmatrix}\end{aligned} \tag{3.125}$$

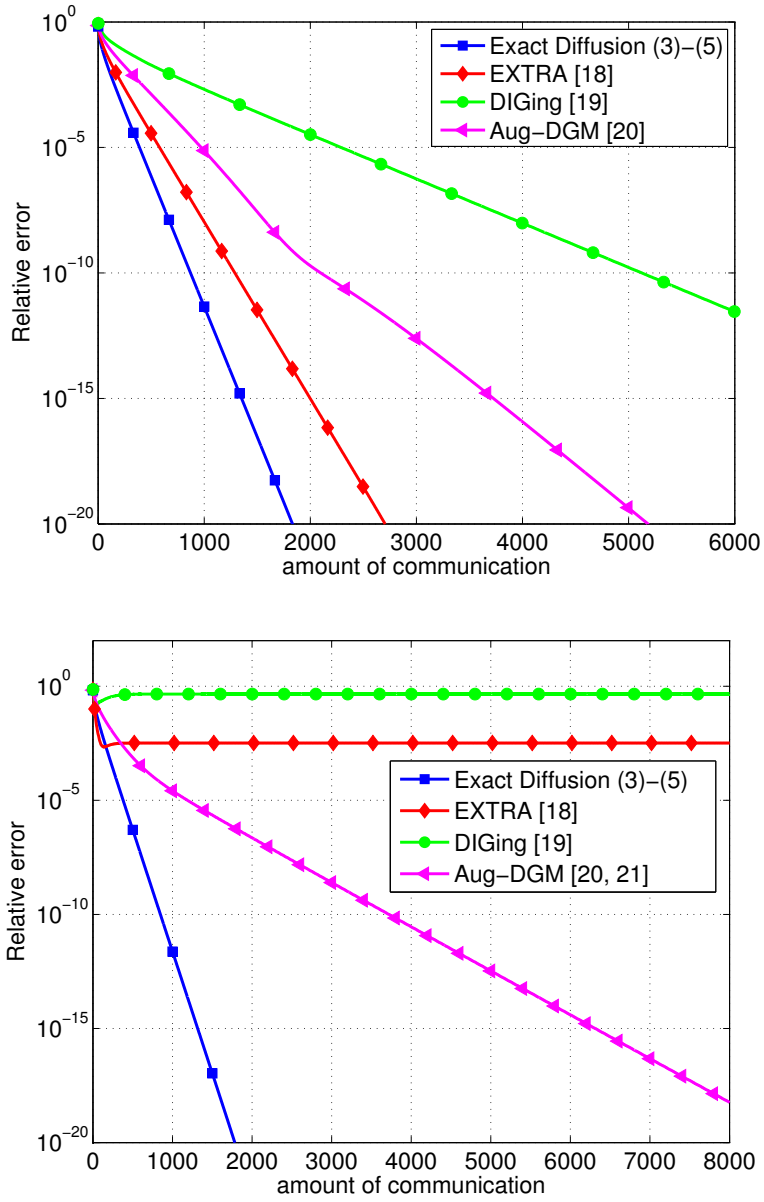

Figure 3.4: Convergence comparison between exact diffusion, EXTRA, DIGing, and Aug-DGM for problem (3.120). In the top plot, the step-sizes for Exact Diffusion, EXTRA, DIGing and AUG-DGM are 0.041, 0.028, 0.014 and 0.033. In the bottom plot, all step-sizes are set as 0.04.

where in (a) we used V^2=$(P$–$PA)/2$ and $\overline{A}^{\mathsf{T}}$=$(I_K$+$A^{\mathsf{T}})/2$. Using $A = Y\Lambda Y^{-1}$ from Lemma 2.3, we have

$$\overline{A}^{\mathsf{T}} = (Y^{-1})^{\mathsf{T}}\overline{\Lambda}Y^{\mathsf{T}}, \quad \overline{A}^{\mathsf{T}} - I_K = (Y^{-1})^{\mathsf{T}}(\overline{\Lambda} - I_K)Y^{\mathsf{T}} \tag{3.126}$$

where $\overline{\Lambda} \triangleq (I_K + \Lambda)/2$. Obviously, $\overline{\Lambda} > 0$ is also a real diagonal matrix. If we let $\overline{\Lambda} = \mathrm{diag}\{\lambda_1(\overline{A}), \cdots, \lambda_K(\overline{A})\}$, it holds that

$$\lambda_k(\overline{A}) = (\lambda_k(A) + 1)/2 > 0, \quad \forall\, k = 1, \cdots, N, \tag{3.127}$$

and $\lambda_1(\overline{A}) = 1$. Moreover, we can also verify that

$$\overline{A}^{\mathsf{T}} - \mathbb{1}_K\, p^{\mathsf{T}} = (Y^{-1})^{\mathsf{T}}\overline{\Lambda}_1 Y^{\mathsf{T}}, \tag{3.128}$$

where $\overline{\Lambda}_1 = \mathrm{diag}\{0, \lambda_2(\overline{A}), \cdots, \lambda_K(\overline{A})\}$. This is because the vectors $\mathbb{1}_K^{\mathsf{T}}$ and p are the left- and right-eigenvectors of $\overline{A}$. Combining relations (3.127) and (3.128), we have

$$\begin{aligned} &\begin{bmatrix} \overline{A}^{\mathsf{T}} & \overline{A}^{\mathsf{T}} - I_K \\ \overline{A}^{\mathsf{T}} - \mathbb{1}_K\, p^{\mathsf{T}} & \overline{A}^{\mathsf{T}} \end{bmatrix} \\ &= \begin{bmatrix} (Y^{-1})^{\mathsf{T}} & 0 \\ 0 & (Y^{-1})^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \overline{\Lambda} & \overline{\Lambda} - I_K \\ \overline{\Lambda}_1 & \overline{\Lambda} \end{bmatrix} \begin{bmatrix} Y^{\mathsf{T}} & 0 \\ 0 & Y^{\mathsf{T}} \end{bmatrix}. \end{aligned} \tag{3.129}$$

With permutation operations, it holds that

$$\begin{bmatrix} \overline{\Lambda} & \overline{\Lambda} - I_K \\ \overline{\Lambda}_1 & \overline{\Lambda} \end{bmatrix} = \Pi \begin{bmatrix} E_1 & 0 & \cdots & 0 \\ 0 & E_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & E_K \end{bmatrix} \Pi^{\mathsf{T}}, \tag{3.130}$$

where $\Pi \in \mathbb{R}^{K\times K}$ is a permutation matrix, and

$$E_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; E_k = \begin{bmatrix} \lambda_k(\overline{A}) & \lambda_k(\overline{A}) - 1 \\ \lambda_k(\overline{A}) & \lambda_k(\overline{A}) \end{bmatrix}, \; \forall k = 2, \cdots, N. \tag{3.131}$$

Now we seek the eigenvalues of E_k. Let d denote an eigenvalue of E_k. The characteristic polynomial of E_k is

$$d^2 - 2\lambda_k(\overline{A})d + \lambda_k(\overline{A}) = 0. \tag{3.132}$$

Therefore, we have

$$d = \frac{2\lambda_k(\overline{A}) \pm \sqrt{4\lambda_k^2(\overline{A}) - 4\lambda_k(\overline{A})}}{2}. \tag{3.133}$$

Since $\lambda_k(\overline{A}) \in (0,1)$ when $k = 2, 3, \cdots, N$, it holds that $4\lambda_k^2(\overline{A}) < 4\lambda_k(\overline{A})$. Therefore, d is a complex number, and its magnitude is $\sqrt{\lambda_k(\overline{A})}$. Therefore, E_k can be diagonalized as

$$E_k = Z_k \begin{bmatrix} d_{k,1} & 0 \\ 0 & d_{k,2} \end{bmatrix} Z_k^{-1} \tag{3.134}$$

where $d_{k,1}$ and $d_{k,2}$ are complex numbers and

$$|d_{k,1}| = |d_{k,2}| = \sqrt{\lambda_k(\overline{A})} < 1. \tag{3.135}$$

Define Z and $\overline{X}$ as

$$Z \triangleq \text{diag}\{I_2, Z_2, Z_3, \cdots, Z_K\} \tag{3.136}$$

$$\overline{X} \triangleq \begin{bmatrix} I_K & 0 \\ 0 & V' \end{bmatrix} \begin{bmatrix} (Y^{-1})^{\mathsf{T}} & 0 \\ 0 & (Y^{-1})^{\mathsf{T}} \end{bmatrix} \Pi Z \tag{3.137}$$

Since each factor in $\overline{X}$ is invertible, $\overline{X}^{-1}$ must exist. Combining (3.125) and (3.129)–(3.135), we finally arrive at

$$B = \overline{X} D \overline{X}^{-1}, \text{ where } D = \begin{bmatrix} I_2 & 0 \\ 0 & D_1 \end{bmatrix}, \tag{3.138}$$

and D_1 has the structure claimed in (3.35).

Therefore, we have established so far the form of the eigenvalue decomposition of B. In this decomposition, each k-th column of $\overline{X}$ is a right-eigenvector associated with the eigenvalue $D(k,k)$, and each k-th row of $\overline{X}^{-1}$ is the left-eigenvector associated with $D(k,k)$. Recall, however, that eigenvectors are not unique. We now verify that we can find eigenvector matrices $\overline{X}$ and $\overline{X}^{-1}$ that have the structure shown in (3.36) and (3.37). To do so, it is sufficient to examine whether the two columns of R are independent right-eigenvectors associated with eigenvalue 1, and the two rows of L are independent left-eigenvectors associated

with 1. Let

$$R = \begin{bmatrix} r_1 & r_2 \end{bmatrix}, \text{ where } r_1 \triangleq \begin{bmatrix} \mathbb{1}_K \\ 0 \end{bmatrix}, \quad r_2 \triangleq \begin{bmatrix} 0 \\ \mathbb{1}_K \end{bmatrix}. \tag{3.139}$$

Obviously, r_1 and r_2 are independent. Since

$$Br_1 = r_1, \quad Br_2 = r_2, \tag{3.140}$$

we know r_1 and r_2 are right-eigenvectors associated with eigenvalue 1. As a result, an eigenvector matrix X can be chosen in the form $X = \left[\begin{array}{c|c} R & X_R \end{array}\right]$, where each k-th column of X_R corresponds to the right-eigenvector associated with eigenvalue $D_1(k,k)$. Similarly, we let

$$L = \begin{bmatrix} \ell_1^{\mathsf{T}} \\ \ell_2^{\mathsf{T}} \end{bmatrix}, \text{ where } \ell_1 \triangleq \begin{bmatrix} p \\ 0 \end{bmatrix}, \ell_2 \triangleq \begin{bmatrix} 0 \\ \frac{1}{K}\mathbb{1}_K \end{bmatrix}. \tag{3.141}$$

It is easy to verify that ℓ_1 and ℓ_2 are independent left-eigenvectors associated with eigenvalue 1. Moreover, since $LR = I_2$, X^{-1} has the structure

$$X^{-1} = \begin{bmatrix} L \\ X_L \end{bmatrix}, \tag{3.142}$$

where each k-th row of X_L corresponds to a left-eigenvector associated with eigenvalue $D_1(k,k)$.

3.B Proof of Theorem 3.1

From the first line of recursion (3.60), we have

$$\bar{x}_i = \left(I_M - \overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{I}\right)\bar{x}_{i-1} - \frac{1}{c}\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{X}_{R,u}\check{x}_{i-1}. \tag{3.143}$$

Squaring both sides and using Jensen's inequality [124] gives

$$\begin{aligned} \|\bar{x}_i\|^2 &= \left\|\left(I_M - \overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{I}\right)\bar{x}_{i-1} - \frac{1}{c}\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{X}_{R,u}\check{x}_{i-1}\right\|^2 \\ &\le \frac{1}{1-t}\left\|I_M - \overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{I}\right\|^2 \|\bar{x}_{i-1}\|^2 \\ &\quad + \frac{1}{t}\frac{1}{c^2}\|\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{X}_{R,u}\|^2\|\check{x}_{i-1}\|^2 \end{aligned} \tag{3.144}$$

for any $t \in (0,1)$. Using $\tau_k = \mu_k/\mu_{\max}$, we obtain

$$\begin{aligned}\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{I} &= \mu_{\max}\sum_{k=1}^{K} p_k\tau_k H_{k,i-1} \\ &\overset{(3.31)}{\geq} \mu_{\max}p_{k_o}\tau_{k_o}\nu I_M \;\overset{\Delta}{=}\; \sigma_{11}\mu_{\max}I_M,\end{aligned} \tag{3.145}$$

where $\sigma_{11} = p_{k_o}\tau_{k_o}\nu$. Similarly, we can also obtain

$$\begin{aligned}\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{I} &= \mu_{\max}\sum_{k=1}^{K} p_k\tau_k H_{k,i-1} \\ &\overset{(3.31)}{\leq} \left(\sum_{k=1}^{K} p_k\tau_k\right)\delta\mu_{\max}I_M \overset{(a)}{\leq} \delta\mu_{\max}I_M,\end{aligned} \tag{3.146}$$

where inequality (a) holds because $\tau_k < 1$ and $\sum_{k=1}^{K} p_k = 1$. It is obvious that $\delta > \sigma_{11}$. As a result, we have

$$(1-\delta\mu_{\max})I_M \leq I_M - \overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{I} \leq (1-\sigma_{11}\mu_{\max})I_M \tag{3.147}$$

which implies that when the step-size satisfy

$$\mu_{\max} < 1/\delta, \tag{3.148}$$

it will hold that

$$\|I_M - \overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{I}\|^2 \leq (1-\sigma_{11}\mu_{\max})^2. \tag{3.149}$$

On the other hand, we have

$$\begin{aligned}\frac{1}{c^2}\|\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{X}_{R,u}\|^2 \;&\leq \frac{1}{c^2}\|\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\|^2\|\mathcal{H}_{i-1}\|^2\|\mathcal{X}_{R,u}\|^2 \\ &\overset{(a)}{\leq} \frac{1}{c^2}\left(\sum_{k=1}^{K}(\tau_k p_k)^2\right)\delta^2\|\mathcal{X}_{R,u}\|^2\mu_{\max}^2 \\ &\overset{(b)}{\leq} \frac{p_{\max}}{c^2}\delta^2\|\mathcal{X}_{R,u}\|^2\mu_{\max}^2\end{aligned} \tag{3.150}$$

where inequality (b) holds because $\tau_k < 1$, $p_k^2 < p_k p_{\max}$ (where $p_{\max} = \max_k\{p_k\}$) and $\sum_{k=1}^{K} p_k = 1$. Inequality (a) follows by noting that $\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M} = \mu_{\max}[p_1\tau_1, \cdots, p_K\tau_K] \otimes I_M$.

Introducing $s=[p_1\tau_1,p_2\tau_2,\cdots,p_K\tau_K]^{\mathsf{T}}\in\mathbb{R}^K$, we have

$$\begin{aligned}\|\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\|^2=&\mu_{\max}^2\|s^{\mathsf{T}}\otimes I_M\|^2=\mu_{\max}^2\lambda_{\max}\Big((s\otimes I_M)(s^{\mathsf{T}}\otimes I_M)\Big)\\ =&\mu_{\max}^2\lambda_{\max}\Big(ss^{\mathsf{T}}\otimes I_M\Big)=\mu_{\max}^2\lambda_{\max}(ss^{\mathsf{T}})\\ =&\mu_{\max}^2\|s\|^2=\mu_{\max}^2\sum_{k=1}^{K}(p_k\tau_k)^2.\end{aligned}\tag{3.151}$$

Recall (3.50) and by introducing $E=\left[\begin{array}{cc} I_{MK} & 0_{MK}\end{array}\right]$, we have $\mathcal{X}_{R,u}=E\mathcal{X}_R$. Therefore, it holds that

$$\|\mathcal{X}_{R,u}\|^2\le\|E\|^2\|\mathcal{X}_R\|^2=\|\mathcal{X}_R\|^2.\tag{3.152}$$

Substituting (3.152) into (3.150), we have

$$\frac{1}{c^2}\|\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{M}\mathcal{H}_{i-1}\mathcal{X}_{R,u}\|^2\le\frac{p_{\max}\delta^2}{c^2}\|\mathcal{X}_R\|^2\mu_{\max}^2\triangleq\sigma_{12}^2\mu_{\max}^2\tag{3.153}$$

where $\sigma_{12}\triangleq\sqrt{p_{\max}}\delta\|\mathcal{X}_R\|/c$. Notice that σ_{12} is independent of $\mu_{\max}$. Substituting (3.149) and (3.150) into (3.144), we get

$$\begin{aligned}\|\bar{x}_i\|^2\le&\frac{1}{1-t}(1-\sigma_{11}\mu_{\max})^2\|\bar{x}_{i-1}\|^2+\frac{1}{t}\sigma_{12}^2\mu_{\max}^2\|\check{x}_{i-1}\|^2\\ =&(1-\sigma_{11}\mu_{\max})\|\bar{x}_{i-1}\|^2+(\sigma_{12}^2/\sigma_{11})\mu_{\max}\|\check{x}_{i-1}\|^2\end{aligned}\tag{3.154}$$

where we are selecting $t=\sigma_{11}\mu_{\max}$.

Next we check the second line of recursion (3.60):

$$\begin{aligned}\check{x}_i=&-c\mathcal{X}_L\mathcal{T}_{i-1}\mathcal{R}_1\bar{x}_{i-1}+(\mathcal{D}_1-\mathcal{X}_L\mathcal{T}_{i-1}\mathcal{X}_R)\check{x}_{i-1}\\ =&\mathcal{D}_1\check{x}_{i-1}-\mathcal{X}_L\mathcal{T}_{i-1}(c\mathcal{R}_1\bar{x}_{i-1}+\mathcal{X}_R\check{x}_{i-1}).\end{aligned}\tag{3.155}$$

Squaring both sides and using Jensen's inequality again,

$$\begin{aligned}\|\check{x}_i\|^2=&\|\mathcal{D}_1\check{x}_{i-1}-\mathcal{X}_L\mathcal{T}_{i-1}(c\mathcal{R}_1\bar{x}_{i-1}+\mathcal{X}_R\check{x}_{i-1})\|^2\\ \le&\frac{\|\mathcal{D}_1\|^2}{t}\|\check{x}_{i-1}\|^2+\frac{1}{1-t}\|\mathcal{X}_L\mathcal{T}_{i-1}(c\mathcal{R}_1\bar{x}_{i-1}+\mathcal{X}_R\check{x}_{i-1})\|^2\\ \le&\frac{\|\mathcal{D}_1\|^2}{t}\|\check{x}_{i-1}\|^2+\frac{2c^2}{1-t}\|\mathcal{X}_L\mathcal{T}_{i-1}\mathcal{R}_1\|^2\|\bar{x}_{i-1}\|^2\\ &+\frac{2}{1-t}\|\mathcal{X}_L\mathcal{T}_{i-1}\mathcal{X}_R\|^2\|\check{x}_{i-1}\|^2.\end{aligned}\tag{3.156}$$

where $t \in (0,1)$. From Lemma 3.3 we have that $\lambda \triangleq \|D_1\| = \sqrt{\lambda_2(\overline{A})} < 1$. By setting $t = \lambda$, we reach

$$\begin{aligned}\|\check{x}_i\|^2 \leq \lambda\|\check{x}_{i-1}\|^2 + 2c^2\|\mathcal{X}_L\mathcal{T}_{i-1}\mathcal{R}_1\|^2\|\bar{x}_{i-1}\|^2/(1-\lambda) \\ + 2\|\mathcal{X}_L\mathcal{T}\mathcal{X}_R\|^2\|\check{x}_{i-1}\|^2/(1-\lambda).\end{aligned} \tag{3.157}$$

We introduce the matrix $\Gamma = \mathrm{diag}\{\tau_1 I_M, \cdots, \tau_K I_M\}$, and note that we can write $\mathcal{M} = \mu_{\max}\Gamma$. Substituting it into (3.28),

$$\begin{aligned}\mathcal{T}_{i-1} &= \mu_{\max}\begin{bmatrix} \overline{\mathcal{A}}^{\mathsf{T}}\Gamma\mathcal{H}_{i-1} & 0 \\ \mathcal{V}\overline{\mathcal{A}}^{\mathsf{T}}\Gamma\mathcal{H}_{i-1} & 0 \end{bmatrix} \\ &= \mu_{\max}\underbrace{\begin{bmatrix} \overline{\mathcal{A}}^{\mathsf{T}} & 0 \\ \mathcal{V}\overline{\mathcal{A}}^{\mathsf{T}} & 0 \end{bmatrix}}_{\triangleq\, \mathcal{T}_d}\begin{bmatrix} \Gamma\mathcal{H}_{i-1} & 0 \\ 0 & \Gamma\mathcal{H}_{i-1} \end{bmatrix},\end{aligned} \tag{3.158}$$

which implies that

$$\|\mathcal{T}_{i-1}\|^2 \leq \mu_{\max}^2\|\mathcal{T}_d\|^2\left(\max_{1\leq k\leq K}\|H_{k,i-1}\|^2\right) \leq \|\mathcal{T}_d\|^2\delta^2\mu_{\max}^2. \tag{3.159}$$

We also emphasize that $\|\mathcal{T}_d\|^2$ is independent of $\mu_{\max}$. With inequality (3.159), we further have

$$c^2\|\mathcal{X}_L\mathcal{T}_{i-1}\mathcal{R}_1\|^2 \leq c^2\mu_{\max}^2\|\mathcal{X}_L\|^2\|\mathcal{T}_d\|^2\|\mathcal{R}_1\|^2\delta^2 \triangleq \sigma_{21}^2\mu_{\max}^2 \tag{3.160}$$

$$\|\mathcal{X}_L\mathcal{T}_{i-1}\mathcal{X}_R\|^2 \leq \mu_{\max}^2\|\mathcal{X}_L\|^2\|\mathcal{T}_d\|^2\|\mathcal{X}_R\|^2\delta^2 \triangleq \sigma_{22}^2\mu_{\max}^2 \tag{3.161}$$

since $\|\mathcal{R}_1\| = 1$, and where σ_{21} and σ_{22} are defined as

$$\sigma_{21} = c\|\mathcal{X}_L\|\|\mathcal{T}_d\|\delta, \quad \sigma_{22} = \|\mathcal{X}_L\|\|\mathcal{T}_d\|\|\mathcal{X}_R\|\delta. \tag{3.162}$$

With (3.160) and (3.161), inequality (3.157) becomes

$$\|\check{x}_i\|^2 \leq \left(\lambda + \frac{2\sigma_{22}^2\mu_{\max}^2}{1-\lambda}\right)\|\check{x}_{i-1}\|^2 + \frac{2\sigma_{21}^2\mu_{\max}^2}{1-\lambda}\|\bar{x}_{i-1}\|^2. \tag{3.163}$$

Combining (3.154) and (3.163), we arrive at the inequality recursion

$$\begin{bmatrix} \|\bar{x}_i\|^2 \\ \|\check{x}_i\|^2 \end{bmatrix} \preceq \underbrace{\begin{bmatrix} 1-\sigma_{11}\mu_{\max} & \frac{\sigma_{12}^2}{\sigma_{11}}\mu_{\max} \\ \frac{2\sigma_{21}^2\mu_{\max}^2}{1-\lambda} & \lambda + \frac{2\sigma_{22}^2\mu_{\max}^2}{1-\lambda} \end{bmatrix}}_{\triangleq\, G}\begin{bmatrix} \|\bar{x}_{i-1}\|^2 \\ \|\check{x}_{i-1}\|^2 \end{bmatrix}. \tag{3.164}$$

Now we check the spectral radius of the matrix G. Recall the fact that the spectral radius of a matrix is upper bounded by any of its norms. Therefore,

$$\rho(G) \leq \|G\|_1 = \max\Big\{1 - \sigma_{11}\mu_{\max} + \frac{2\sigma_{21}^2\mu_{\max}^2}{1-\lambda}, \\ \lambda + \frac{\sigma_{12}^2}{\sigma_{11}}\mu_{\max} + \frac{2\sigma_{22}^2\mu_{\max}^2}{1-\lambda}\Big\}, \tag{3.165}$$

where we already know that $\lambda < 1$. To guarantee $\rho(G) < 1$, it is enough to select the step-size parameter small enough to satisfy

$$1 - \sigma_{11}\mu_{\max} + \frac{2\sigma_{21}^2\mu_{\max}^2}{1-\lambda} < 1, \tag{3.166}$$

$$\lambda + \frac{\sigma_{12}^2}{\sigma_{11}}\mu_{\max} + \frac{2\sigma_{22}^2\mu_{\max}^2}{1-\lambda} < 1. \tag{3.167}$$

To get a simpler upper bound, we transform (3.167) such that

$$\begin{aligned} &\lambda + \frac{\sigma_{12}^2}{\sigma_{11}}\mu_{\max} + \frac{2\sigma_{22}^2\mu_{\max}^2}{1-\lambda} \\ &= \lambda + \frac{2\sigma_{12}^2}{\sigma_{11}}\mu_{\max} - \left(\frac{\sigma_{12}^2}{\sigma_{11}}\mu_{\max} - \frac{2\sigma_{22}^2\mu_{\max}^2}{1-\lambda}\right) \leq \lambda + \frac{2\sigma_{12}^2}{\sigma_{11}}\mu_{\max}, \end{aligned} \tag{3.168}$$

where the last inequality holds when

$$\mu_{\max} \leq \frac{\sigma_{12}^2(1-\lambda)}{2\sigma_{11}\sigma_{22}^2}. \tag{3.169}$$

If, in addition, we let (3.168) be less than 1, which is equivalent to selecting

$$\mu_{\max} \leq \frac{\sigma_{11}(1-\lambda)}{2\sigma_{12}^2}, \tag{3.170}$$

then we guarantee equality (3.167). Combing (3.166), (3.169) and (3.170), we have

$$\mu_{\max} \leq \min\left\{\frac{\sigma_{11}(1-\lambda)}{2\sigma_{21}^2}, \frac{\sigma_{12}^2(1-\lambda)}{2\sigma_{11}\sigma_{22}^2}, \frac{\sigma_{11}(1-\lambda)}{2\sigma_{12}^2}\right\} \tag{3.171}$$

This together with (3.148), i.e.

$$\mu_{\max} < 1/\delta \tag{3.172}$$

will guarantee $\|G\|_1$ to be less than 1. In fact, the upper bound in (3.171) can be further simplified. From the definitions of σ_{11}, σ_{12}, σ_{21} and σ_{22}, we have

$$\frac{\sigma_{11}(1-\lambda)}{2\sigma_{21}^2} = \frac{p_{k_o}\tau_{k_o}\nu(1-\lambda)}{2c^2\|\mathcal{X}_L\|^2\|\mathcal{T}_d\|^2\delta^2}, \tag{3.173}$$

$$\frac{\sigma_{12}^2(1-\lambda)}{2\sigma_{11}\sigma_{22}^2} = \frac{p_{\max}(1-\lambda)}{2p_{k_o}\tau_{k_o}\nu\|\mathcal{X}_L\|^2\|\mathcal{T}_d\|^2c^2}, \tag{3.174}$$

$$\frac{\sigma_{11}(1-\lambda)}{2\sigma_{12}^2} = \frac{p_{k_o}\tau_{k_o}\nu(1-\lambda)c^2}{2p_{\max}\|\mathcal{X}_R\|^2\delta^2}. \tag{3.175}$$

First, notice that

$$\frac{\sigma_{11}(1-\lambda)}{2\sigma_{21}^2}\Big/\frac{\sigma_{12}^2(1-\lambda)}{2\sigma_{11}\sigma_{22}^2} = \frac{(p_{k_o}\tau_{k_o}\nu)^2}{p_{\max}\delta^2} < 1 \tag{3.176}$$

because $p_{k_o} < p_{\max}$, $\tau_{k_o} < 1$ and $\nu < \delta$. Therefore, the inequality in (3.171) is equivalent to

$$\begin{aligned}\mu_{\max} &\le \min\left\{\frac{p_{k_o}\tau_{k_o}\nu(1-\lambda)}{2c^2\|\mathcal{X}_L\|^2\|\mathcal{T}_d\|^2\delta^2}, \frac{p_{k_o}\tau_{k_o}\nu(1-\lambda)c^2}{2p_{\max}\|\mathcal{X}_R\|^2\delta^2}\right\} \\ &= \frac{p_{k_o}\tau_{k_o}\nu(1-\lambda)}{2\delta^2}\min\left\{\frac{1}{\|\mathcal{X}_L\|^2\|\mathcal{T}_d\|^2c^2}, \frac{c^2}{p_{\max}\|\mathcal{X}_R\|^2}\right\}.\end{aligned} \tag{3.177}$$

It is observed that the constant value c affects the upper bound in (3.177). If c is sufficiently large, then the first term in (3.177) dominates and $\mu_{\max}$ has a narrow feasible set. On the other hand, if c is sufficiently small, then the second term dominates and $\mu_{\max}$ will also have a narrow feasible set. To make the feasible set of $\mu_{\max}$ as large as possible, we should optimize c to maximize

$$\min\left\{\frac{1}{\|\mathcal{X}_L\|^2\|\mathcal{T}_d\|^2c^2}, \quad \frac{c^2}{p_{\max}\|\mathcal{X}_R\|^2}\right\}. \tag{3.178}$$

Notice that the first term $1/(\|\mathcal{X}_L\|^2\|\mathcal{T}_d\|^2c^2)$ is monotone decreasing with c^2, while the second term $c^2/\|\mathcal{X}_R\|^2$ is monotone increasing with c^2. Therefore, when

$$\frac{1}{\|\mathcal{X}_L\|^2\|\mathcal{T}_d\|^2c^2} = \frac{c^2}{p_{\max}\|\mathcal{X}_R\|^2} \Longleftrightarrow c^2 = \frac{\sqrt{p_{\max}}\|\mathcal{X}_R\|}{\|\mathcal{X}_L\|\|\mathcal{T}_d\|}, \tag{3.179}$$

we get the maximum upper bound for $\mu_{\max}$, i.e.

$$\mu_{\max} \le \frac{p_{k_o}\tau_{k_o}\nu(1-\lambda)}{2\sqrt{p_{\max}}\|\mathcal{X}_L\|\|\mathcal{T}_d\|\|\mathcal{X}_R\|\delta^2}. \tag{3.180}$$

Next we compare the above upper bound with $1/\delta$. Recall that for any matrix A, its spectral radius is smaller than its $2-$induced norm so that

$$\|\mathcal{T}_d\| \geq \rho(\mathcal{T}_d) \overset{(3.158)}{=} \rho(\overline{\mathcal{A}}) = 1. \tag{3.181}$$

Moreover, recall from Lemma 3.3 that $X_L X_R = I_{2(K-1)}$, so that $\mathcal{X}_L \mathcal{X}_R = X_L X_R \otimes I_M = I_{2M(K-1)}$, which implies that

$$\|\mathcal{X}_L\|\|\mathcal{X}_R\| \geq \|\mathcal{X}_L \mathcal{X}_R\| = 1. \tag{3.182}$$

Using relations (3.181) and (3.182), and recalling that $p_{k_o} \leq p_{\max} < \sqrt{p_{\max}}$, $\tau_{k_o} < 1$, $1-\lambda < 1$ and $\nu < \delta$, we have

$$\frac{p_{k_o}\tau_{k_o}\nu(1-\lambda)}{2\sqrt{p_{\max}}\|\mathcal{X}_L\|\|\mathcal{T}_d\|\|\mathcal{X}_R\|\delta^2} \leq \frac{\nu}{\delta^2} < \frac{\delta}{\delta^2} = \frac{1}{\delta}. \tag{3.183}$$

Therefore, the upper bounds in (3.171), (3.172) are determined by

$$\mu_{\max} \leq \frac{p_{k_o}\tau_{k_o}\nu(1-\lambda)}{2\sqrt{p_{\max}}\|\mathcal{X}_L\|\|\mathcal{T}_d\|\|\mathcal{X}_R\|\delta^2}. \tag{3.184}$$

In other words, when $\mu_{\max}$ satisfies (3.184), $\|G\|_1$ will be guaranteed to be less than 1, i.e.,

$$\begin{aligned}
\|G\|_1 &= \max\Big\{1 - \sigma_{11}\mu_{\max} + \frac{2\sigma_{21}^2\mu_{\max}^2}{1-\lambda}, \\
&\qquad\qquad \lambda + \frac{\sigma_{12}^2}{\sigma_{11}}\mu_{\max} + \frac{2\sigma_{22}^2\mu_{\max}^2}{1-\lambda}\Big\} \\
&= \max\Big\{1 - p_{k_o}\tau_{k_o}\nu\mu_{\max} + \frac{2c^2\|\mathcal{X}_L\|^2\|\mathcal{T}_d\|^2\delta^2\mu_{\max}^2}{1-\lambda} \\
&\quad \lambda + \frac{p_{\max}\|\mathcal{X}_R\|^2\delta^2}{c^2 p_{k_o}\tau_{k_o}\nu}\mu_{\max} + \frac{2\|\mathcal{X}_L\|^2\|\mathcal{T}_d\|^2\|\mathcal{X}_R\|^2\delta^2\mu_{\max}^2}{1-\lambda}\Big\} \\
&\overset{(3.179)}{=} \max\Big\{1 - p_{k_o}\tau_{k_o}\nu\mu_{\max} + \frac{2\sqrt{p_{\max}}\alpha_d\delta^2\mu_{\max}^2}{1-\lambda}, \\
&\qquad \lambda + \frac{\sqrt{p_{\max}}\alpha_d\delta^2\mu_{\max}}{p_{k_o}\tau_{k_o}\nu} + \frac{2\alpha_d^2\delta^2\mu_{\max}^2}{1-\lambda}\Big\} < 1,
\end{aligned} \tag{3.185}$$

where $\alpha_d \overset{\Delta}{=} \|\mathcal{X}_L\|\|\mathcal{T}_d\|\|\mathcal{X}_R\|$. Let

$$z_i \overset{\Delta}{=} \begin{bmatrix} \|\bar{x}_i\|^2 \\ \|\check{x}_i\|^2 \end{bmatrix} \succeq 0, \tag{3.186}$$

and note from (3.164) that

$$z_i \preceq Gz_{i-1}. \tag{3.187}$$

Computing the 1-norm of both sides gives

$$\begin{aligned}\|\bar{x}_i\|^2+\|\check{x}_i\|^2=\|z_i\|_1 &\leq \|G\|_1\|z_{i-1}\|_1 = \rho(\|\bar{x}_{i-1}\|^2+\|\check{x}_{i-1}\|^2),\\ &\leq \rho^i(\|\bar{x}_0\|^2+\|\check{x}_0\|^2),\end{aligned} \tag{3.188}$$

where we define $\rho \triangleq \|G\|_1$. Inequality (3.188) is equivalent to

$$\left\|\begin{bmatrix}\bar{x}_i\\ \check{x}_i\end{bmatrix}\right\|^2 \leq \rho^i \left\|\begin{bmatrix}\bar{x}_0\\ \check{x}_0\end{bmatrix}\right\|^2, \tag{3.189}$$

By re-incorporating $\widehat{x}_i = 0$, relation (3.189) also implies that

$$\left\|\begin{bmatrix}\bar{x}_i\\ \widehat{x}_i\\ \check{x}_i\end{bmatrix}\right\|^2 \leq \rho^i \left\|\begin{bmatrix}\bar{x}_0\\ \widehat{x}_0\\ \check{x}_0\end{bmatrix}\right\|^2 \triangleq C_0\rho^i. \tag{3.190}$$

From (3.48) we conclude that

$$\left\|\begin{bmatrix}\widetilde{\mathcal{W}}_i\\ \widetilde{\mathcal{Y}}_i\end{bmatrix}\right\|^2 \leq \|\mathcal{X}\|^2 \left\|\begin{bmatrix}\bar{x}_i\\ \widehat{x}_i\\ \check{x}_i\end{bmatrix}\right\|^2 \leq C\rho^i, \tag{3.191}$$

where the constant $C = \|\mathcal{X}\|^2 C_0$.

3.C Proof of Theorem 3.2

We define

$$\mathcal{M}_i' \triangleq \mu_o \mathrm{diag}\{q_1 I_M/z_{1,i}(1), \cdots, q_K I_M/z_{K,i}(K)\}. \tag{3.192}$$

Substituting recursions (2.97) and (2.98) into expression (2.99) we obtain (compare with (2.92)):

$$\mathcal{W}_i = \overline{\mathcal{A}}^{\mathsf{T}}\left[2\mathcal{W}_{i-1} - \mathcal{W}_{i-2} - \left(\mathcal{M}_i'\nabla\mathcal{J}^o(\mathcal{W}_{i-1}) - \mathcal{M}_{i-1}'\nabla\mathcal{J}^o(\mathcal{W}_{i-2})\right)\right], \tag{3.193}$$

which can be rewritten into a primal-dual form (compare with (2.88)):

$$\begin{cases} \mathcal{w}_i = \overline{\mathcal{A}}^{\mathsf{T}}\Big(\mathcal{w}_{i-1} - \mathcal{M}_i'\nabla\mathcal{J}^o(\mathcal{w}_{i-1})\Big) - \mathcal{P}^{-1}\mathcal{V}\mathcal{y}_{i-1}, \\ \mathcal{y}_i = \mathcal{y}_{i-1} + \mathcal{V}\mathcal{w}_i. \end{cases} \tag{3.194}$$

For the initialization, we set $y_{-1} = 0$ and $\mathcal{w}_{-1}$ to be any value, and hence for $i = 0$ we have

$$\begin{cases} \mathcal{w}_0 = \overline{\mathcal{A}}^{\mathsf{T}}\Big(\mathcal{w}_{-1} - \mathcal{M}_0'\nabla\mathcal{J}^o(\mathcal{w}_{-1})\Big), \\ \mathcal{y}_0 = \mathcal{V}\mathcal{w}_0. \end{cases} \tag{3.195}$$

Recursions (3.194) and (3.195) are very close to the standard exact diffusion recursions (2.88) and (2.89), except that the step-size matrix $\mathcal{M}_i'$ is now changing with iteration i. Following the arguments (3.14) – (3.16), we have

$$\begin{cases} \overline{\mathcal{A}}^{\mathsf{T}}\widetilde{\mathcal{w}}_i = \overline{\mathcal{A}}^{\mathsf{T}}\Big(\widetilde{\mathcal{w}}_{i-1} + \mathcal{M}_i'\nabla\mathcal{J}^o(\mathcal{w}_{i-1})\Big) + \mathcal{P}^{-1}\mathcal{V}\mathcal{y}_i, \\ \widetilde{\mathcal{y}}_i = \widetilde{\mathcal{y}}_{i-1} - \mathcal{V}\mathcal{w}_i. \end{cases} \tag{3.196}$$

Subtracting optimality conditions (3.1)–(3.2) from (3.196) leads to

$$\begin{cases} \overline{\mathcal{A}}^{\mathsf{T}}\widetilde{\mathcal{w}}_i = \overline{\mathcal{A}}^{\mathsf{T}}\Big(\widetilde{\mathcal{w}}_{i-1} + \mathcal{M}\big[\nabla\mathcal{J}^o(\mathcal{w}_{i-1}) - \nabla\mathcal{J}^o(w^\star)\big]\Big) - \mathcal{P}^{-1}\mathcal{V}\widetilde{\mathcal{y}}_i \\ \qquad\qquad + \overline{\mathcal{A}}^{\mathsf{T}}(\mathcal{M}_i' - \mathcal{M})\nabla\mathcal{J}^o(\mathcal{w}_{i-1}), \\ \widetilde{\mathcal{y}}_i = \widetilde{\mathcal{y}}_{i-1} + \mathcal{V}\widetilde{\mathcal{w}}_i. \end{cases} \tag{3.197}$$

Comparing recursions (3.197) and (3.17), it is observed that recursion (3.197) has an extra "mismatch" term, $\overline{\mathcal{A}}^{\mathsf{T}}(\mathcal{M}_i' - \mathcal{M})\nabla\mathcal{J}^o(\mathcal{w}_{i-1})$. This mismatch arises because we do not know the perron vector p in advance. We need to run the power iteration to learn it. Intuitively, since $\mathcal{M}_i' \to \mathcal{M}$ as $i \to \infty$, we can expect the mismatch term to vanish gradually. Let

$$e_i \;\triangleq\; (\mathcal{M}_i' - \mathcal{M})\nabla\mathcal{J}^o(\mathcal{w}_{i-1}). \tag{3.198}$$

By following arguments (3.18)–(3.23), recursion (3.197) is equivalent to

$$\begin{aligned} &\begin{bmatrix} \overline{\mathcal{A}}^{\mathsf{T}} & \mathcal{P}^{-1}\mathcal{V} \\ -\mathcal{V} & I_{MK} \end{bmatrix} \begin{bmatrix} \widetilde{\mathcal{w}}_i \\ \widetilde{\mathcal{y}}_i \end{bmatrix} \\ &= \begin{bmatrix} \overline{\mathcal{A}}^{\mathsf{T}}(I_{MK} - \mathcal{M}\mathcal{H}_{i-1}) & 0 \\ 0 & I_{MK} \end{bmatrix} \begin{bmatrix} \widetilde{\mathcal{w}}_{i-1} \\ \widetilde{\mathcal{y}}_{i-1} \end{bmatrix} + \begin{bmatrix} \overline{\mathcal{A}}^{\mathsf{T}} \\ 0 \end{bmatrix} e_i. \end{aligned} \tag{3.199}$$

By following (3.24)–(3.28), recursion (3.199) can be rewritten as

$$\boxed{\begin{bmatrix} \widetilde{\mathcal{W}}_i \\ \widetilde{\mathcal{Y}}_i \end{bmatrix} = (\mathcal{B} - \mathcal{T}_{i-1}) \begin{bmatrix} \widetilde{\mathcal{W}}_{i-1} \\ \widetilde{\mathcal{Y}}_{i-1} \end{bmatrix} + \mathcal{B}_\ell e_i} \tag{3.200}$$

where $\mathcal{B}$ and $\mathcal{T}_i$ are defined in (3.28), and

$$\mathcal{B}_\ell = \begin{bmatrix} \overline{\mathcal{A}}^{\mathsf{T}} \\ \mathcal{V}\overline{\mathcal{A}}^{\mathsf{T}} \end{bmatrix} \tag{3.201}$$

Relation (3.200) is the error dynamics for the exact diffusion algorithm 1′. Comparing (3.200) with (3.26), we find that algorithm 1′ is essentially the standard exact diffusion with error perturbation. Using Lemma (3.3) and by following arguments from (3.40) to (3.60), we can transform the error dynamics (3.200) into

$$\begin{aligned} \begin{bmatrix} \bar{x}_i \\ \check{x}_i \end{bmatrix} = & \begin{bmatrix} I_M - \overline{\mathcal{P}}^{\mathsf{T}} \mathcal{M}\mathcal{H}_{i-1}\mathcal{I} & -\frac{1}{c}\overline{\mathcal{P}}^{\mathsf{T}} \mathcal{M}\mathcal{H}_{i-1}\mathcal{X}_{R,u} \\ -c\mathcal{X}_L\mathcal{T}_{i-1}\mathcal{R}_1 & \mathcal{D}_1 - \mathcal{X}_L\mathcal{T}_{i-1}\mathcal{X}_R \end{bmatrix} \begin{bmatrix} \bar{x}_{i-1} \\ \check{x}_{i-1} \end{bmatrix} \\ & + \begin{bmatrix} \overline{\mathcal{P}}^{\mathsf{T}} \\ c\mathcal{X}_L\mathcal{B}_\ell \end{bmatrix} e_i. \end{aligned} \tag{3.202}$$

Next we analyze the convergence of the above recursion. From the first line we have

$$\begin{aligned} \|\bar{x}_i\|^2 = & \left\| \left(I_M - \overline{\mathcal{P}}^{\mathsf{T}} \mathcal{M}\mathcal{H}_{i-1}\mathcal{I} \right) \bar{x}_{i-1} \right. \\ & \left. -\frac{1}{c}\overline{\mathcal{P}}^{\mathsf{T}} \mathcal{M}\mathcal{H}_{i-1}\mathcal{X}_{R,u}\check{x}_{i-1} + \overline{\mathcal{P}}^{\mathsf{T}} e_i \right\|^2 \end{aligned} \tag{3.203}$$

$$\begin{aligned} \leq & \frac{1}{1-t} \left\| I_M - \overline{\mathcal{P}}^{\mathsf{T}} \mathcal{M}\mathcal{H}_{i-1}\mathcal{I} \right\|^2 \|\bar{x}_{i-1}\|^2 \\ & + \frac{2}{t}\frac{1}{c^2}\|\overline{\mathcal{P}}^{\mathsf{T}} \mathcal{M}\mathcal{H}_{i-1}\mathcal{X}_{R,u}\|^2 \|\check{x}_{i-1}\|^2 + \frac{2}{t}\|\overline{\mathcal{P}}^{\mathsf{T}}\|^2 \|e_i\|^2 \\ \leq & (1 - \sigma_{11}\mu_{\max})\|\bar{x}_{i-1}\|^2 + \frac{\sigma_{12}^2\mu_{\max}}{\sigma_{11}}\|\check{x}_{i-1}\|^2 + \frac{2\|e_i\|^2}{\sigma_{11}\mu_{\max}}, \end{aligned} \tag{3.204}$$

where the last inequality follows the arguments in (3.143)–(3.154). From the second line of recursion (3.202), we have

$$\begin{aligned}
&\|\check{x}_i\|^2 \\
=&\|\mathcal{D}_1\check{x}_{i-1}-\mathcal{X}_L\mathcal{T}_{i-1}(c\mathcal{R}_1\bar{x}_{i-1}+\mathcal{X}_R\check{x}_{i-1})+c\mathcal{X}_L\mathcal{B}_\ell e_i\|^2 \\
\leq&\frac{\|\mathcal{D}_1\|^2}{t}\|\check{x}_{i-1}\|^2+\frac{2c^2}{1-t}\|\mathcal{X}_L\mathcal{T}_{i-1}\mathcal{R}_1\|^2\|\bar{x}_{i-1}\|^2 \\
&+\frac{2}{1-t}\|\mathcal{X}_L\mathcal{T}_{i-1}\mathcal{X}_R\|^2\|\check{x}_{i-1}\|^2+\frac{2c^2}{1-t}\|\mathcal{X}_L\mathcal{B}_\ell\|^2\|e_i\|^2 \\
\leq&\left(\lambda+\frac{2\sigma_{22}^2\mu_{\max}^2}{1-\lambda}\right)\|\check{x}_{i-1}\|^2+\frac{2\sigma_{21}^2\mu_{\max}^2}{1-\lambda}\|\bar{x}_{i-1}\|^2+\frac{2c^2d\|e_i\|^2}{1-\lambda},
\end{aligned} \tag{3.205}$$

where $d \triangleq \|\mathcal{X}_L\mathcal{B}_\ell\|^2$ is independent of iteration i. Moreover, the last inequality holds because of arguments in (3.155)–(3.163). Combining (3.204) and (3.205), we arrive at the inequality recursion (compare with (3.164)):

$$\begin{aligned}
\begin{bmatrix}\|\bar{x}_i\|^2\\ \|\check{x}_i\|^2\end{bmatrix} \preceq& \underbrace{\begin{bmatrix}1-\sigma_{11}\mu_{\max} & \frac{\sigma_{12}^2}{\sigma_{11}}\mu_{\max}\\ \frac{2\sigma_{21}^2\mu_{\max}^2}{1-\lambda} & \lambda+\frac{2\sigma_{22}^2\mu_{\max}^2}{1-\lambda}\end{bmatrix}}_{\triangleq G}\begin{bmatrix}\|\bar{x}_{i-1}\|^2\\ \|\check{x}_{i-1}\|^2\end{bmatrix} \\
&+\begin{bmatrix}\frac{2}{\sigma_{11}\mu_{\max}}\\ \frac{2c^2d}{1-\lambda}\end{bmatrix}\|e_i\|^2.
\end{aligned} \tag{3.206}$$

Next let us bound the mismatch term $\|e_i\|^2$. From (3.198) we have

$$\begin{aligned}
e_i =& (\mathcal{M}_i'-\mathcal{M})\left(\nabla\mathcal{J}^o(\mathcal{w}_{i-1})-\nabla\mathcal{J}^o(\mathcal{w}^\star)\right) \\
&+(\mathcal{M}_i'-\mathcal{M})\nabla\mathcal{J}^o(\mathcal{w}^\star) \\
\overset{(3.21)}{=}& -(\mathcal{M}_i'-\mathcal{M})\mathcal{H}_{i-1}\widetilde{\mathcal{w}}_{i-1}+(\mathcal{M}_i'-\mathcal{M})\nabla\mathcal{J}^o(\mathcal{w}^\star).
\end{aligned} \tag{3.207}$$

which implies that

$$\|e_i\|^2 \leq 2\delta^2\|\mathcal{M}_i'-\mathcal{M}\|^2\|\widetilde{\mathcal{w}}_{i-1}\|^2+2\|\mathcal{M}_i'-\mathcal{M}\|^2 g, \tag{3.208}$$

where $g \triangleq \|\mathcal{J}^o(\mathcal{w}^\star)\|^2$ is a constant independent of iteration. Recall that $\mathcal{M}=M\otimes I_M$ and $\mathcal{M}_i'=M_i'\otimes I_M$ where

$$\begin{aligned}
M &= \mathrm{diag}\{\mu_1,\mu_2,\cdots,\mu_K\}, \\
M_i' &= \mathrm{diag}\left\{\frac{q_1\mu_o}{z_{1,i}(1)},\cdots,\frac{q_K\mu_o}{z_{K,i}(K)}\right\}.
\end{aligned} \tag{3.209}$$

Using the relation $\mu_k = q_k\mu_o/p_k$ (see (2.9)), we have

$$\begin{aligned}
&M - M_i' \\
=&\text{diag}\left\{\frac{q_1\mu_o}{p_1}\left(1-\frac{p_1}{z_{1,i}(1)}\right),\cdots,\frac{q_K\mu_o}{p_K}\left(1-\frac{p_K}{z_{K,i}(K)}\right)\right\} \\
=&\text{diag}\left\{\mu_1\left(1-\frac{p_1}{z_{1,i}(1)}\right),\cdots,\mu_K\left(1-\frac{p_K}{z_{K,i}(K)}\right)\right\} \\
=&\mu_{\max}\text{diag}\left\{\tau_1\left(1-\frac{p_1}{z_{1,i}(1)}\right),\cdots,\tau_K\left(1-\frac{p_K}{z_{K,i}(K)}\right)\right\},
\end{aligned} \tag{3.210}$$

where $\tau_k = \mu_k/\mu_{\max} \leq 1$.

Now we examine the convergence of $1 - p_k/z_{k,i}(k)$. From the discussion in Policy 5 form from Chapter 2, it is known that $\mathcal{z}_i$ generated from the power iteration (see equation (2.36)) will converge to $[(\mathbb{1}_K \otimes I_K)(p^\mathsf{T} \otimes I_K)]\mathcal{z}_{-1}$. Therefore,

$$\begin{aligned}
&\mathcal{z}_i - [(\mathbb{1}_K \otimes I_K)(p^\mathsf{T} \otimes I_K)]\mathcal{z}_{-1} \\
&\quad= \left[\left(\mathcal{A}^\mathsf{T}\right)^{i+1} - (\mathbb{1}_K \otimes I_K)(p^\mathsf{T} \otimes I_K)\right]\mathcal{z}_{-1} \\
&\quad= \left\{\left[\left(A^\mathsf{T}\right)^{i+1} - \mathbb{1}_K p^\mathsf{T}\right] \otimes I_K\right\} z_{-1} \\
&\quad= \left\{\left[A^\mathsf{T} - \mathbb{1}_K p^\mathsf{T}\right]^{i+1} \otimes I_K\right\} z_{-1}.
\end{aligned} \tag{3.211}$$

Recall from the discussion in Chapter 2 that

$$[(\mathbb{1}_K \otimes I_K)(p^\mathsf{T} \otimes I_K)]\mathcal{z}_{-1} = \text{col}\{p,\cdots,p\} \in \mathbb{R}^{K^2}. \tag{3.212}$$

As a result,

$$\begin{aligned}
|z_{k,i}(k) - p_k|^2 &\leq \|\mathcal{z}_i - [(\mathbb{1}_K \otimes I_K)(p^\mathsf{T} \otimes I_K)]\mathcal{z}_{-1}\|^2 \\
&\leq \|A^\mathsf{T} - \mathbb{1}_K p^\mathsf{T}\|^{2(i+1)}\|\mathcal{z}_{-1}\|^2 \\
&= h \cdot \rho_A^{2(i+1)}, \quad \forall k = 1,\cdots,N.
\end{aligned} \tag{3.213}$$

where $h \triangleq \|\mathcal{z}_{-1}\|^2$ is a constant, and ρ_A is the second largest eigenvalue magnitude of matrix A, i.e., $\rho_A = \max\{|\lambda_2(A)|, |\lambda_K(A)|\}$. Since A is locally balanced, we know A is diagonalizable with real eigenvalue in $(-1, 1]$, and it has a single eigenvalue at 1 (see Table 2.1), we conclude that $\rho_A < 1$. Also, recall from the discussion at the end of Policy 5 in

Chapter 2 that $z_{k,i}(k) > 0$ is guaranteed when $\bar{a}_{kk} > 0$. Let

$$\alpha_k \triangleq \min_i\{z_{k,i}(k)\} > 0, \quad \forall\, k = 1, \cdots, N \tag{3.214}$$

Combining (3.213) and (3.214), it holds that for $k = 1, \cdots, N$,

$$\left(1 - \frac{p_k}{z_{k,i}(k)}\right)^2 \leq \frac{h}{\alpha_k}\rho_A^{2(i+1)} = h_k \rho_A^{2(i+1)}, \tag{3.215}$$

where we define $h_k \triangleq h/\alpha_k$. Substituting (3.215) into (3.210), it holds that

$$\|\mathcal{M}_i' - \mathcal{M}\|^2 = \|M_i' - M\|^2 \leq \mu_{\max}^2 h' \rho_A^{2(i+1)}, \tag{3.216}$$

where $h' \triangleq \max_k\{\tau_k^2 h_k\}$ is a constant independent of iterations. Substituting (3.216) into (3.208), we have

$$\begin{aligned}
\|e_i\|^2 &\leq 2\delta^2 \|\mathcal{M}_i' - \mathcal{M}\|^2 \|\widetilde{\mathcal{W}}_{i-1}\|^2 + 2\|\mathcal{M}_i' - \mathcal{M}\|^2 g \\
&\leq 2\delta^2 \mu_{\max}^2 h' \rho_A^{2(i+1)} \|\widetilde{\mathcal{W}}_{i-1}\|^2 + 2\mu_{\max}^2 h' g \rho_A^{2(i+1)} \\
&\leq 2\delta^2 \mu_{\max}^2 h' \rho_A^{2(i+1)} \big(\|\widetilde{\mathcal{W}}_{i-1}\|^2 + \|\widetilde{\mathcal{Y}}_{i-1}\|^2\big) \\
&\quad + 2\mu_{\max}^2 h' g \rho_A^{2(i+1)}.
\end{aligned} \tag{3.217}$$

Recall from (3.48) that

$$\begin{bmatrix} \widetilde{\mathcal{W}}_i \\ \widetilde{\mathcal{Y}}_i \end{bmatrix} = \mathcal{X}' \begin{bmatrix} \bar{\mathcal{X}}_i \\ \widehat{\mathcal{X}}_i \\ \check{\mathcal{X}}_i \end{bmatrix}. \tag{3.218}$$

We therefore have

$$\begin{aligned}
\|\widetilde{\mathcal{W}}_i\|^2 + \|\widetilde{\mathcal{Y}}_i\|^2 &\leq \|\mathcal{X}'\|^2 \left(\|\bar{\mathcal{X}}_i\|^2 + \|\widehat{\mathcal{X}}_i\|^2 + \|\check{\mathcal{X}}_i\|^2\right) \\
&= \|\mathcal{X}'\|^2 \left(\|\bar{\mathcal{X}}_i\|^2 + \|\check{\mathcal{X}}_i\|^2\right),
\end{aligned} \tag{3.219}$$

where the last equality holds because $\widehat{\mathcal{X}}_i = 0$ for $i = 0, 1, \cdots$ (see (3.59)). Substituting (3.219) into (3.217), we have

$$\begin{aligned}
\|e_i\|^2 \leq\ & 2\delta^2 \mu_{\max}^2 h' \|\mathcal{X}'\|^2 \rho_A^{2(i+1)} \big(\|\bar{\mathcal{X}}_{i-1}\|^2 + \|\check{\mathcal{X}}_{i-1}\|^2\big) \\
& + 2\mu_{\max}^2 h' g \rho_A^{2(i+1)}
\end{aligned} \tag{3.220}$$

Substituting (3.220) into (3.206), we have

$$\begin{bmatrix} \|\bar{x}_i\|^2 \\ \|\check{x}_i\|^2 \end{bmatrix} \preceq \begin{bmatrix} 1-\sigma_{11}\mu_{\max}+b'\mu_{\max}\rho_A^{2(i+1)} & \frac{\sigma_{12}^2}{\sigma_{11}}\mu_{\max}+b'\mu_{\max}\rho_A^{2(i+1)} \\ \frac{2\sigma_{21}^2\mu_{\max}^2}{1-\lambda}+c'\mu_{\max}^2\rho_A^{2(i+1)} & \lambda+\frac{2\sigma_{22}^2\mu_{\max}^2}{1-\lambda}+c'\mu_{\max}^2\rho_A^{2(i+1)} \end{bmatrix} \cdot \begin{bmatrix} \|\bar{x}_{i-1}\|^2 \\ \|\check{x}_{i-1}\|^2 \end{bmatrix} + \begin{bmatrix} d'\mu_{\max}\rho_A^{2(i+1)} \\ e'\mu_{\max}^2\rho_A^{2(i+1)} \end{bmatrix}, \tag{3.221}$$

where b', c', d', e' are constants defined as

$$b' \triangleq 4\delta^2 h'\|\mathcal{X}'\|^2/\sigma_{11}, \quad c' \triangleq 4\delta^2 h'\|\mathcal{X}'\|^2 c^2 d/(1-\lambda), \tag{3.222}$$

$$d' \triangleq 4h'g/\sigma_{11}, \qquad e' \triangleq 4h'gc^2 d/(1-\lambda). \tag{3.223}$$

These constants are independent of iterations. It can be verified that when iteration i is large enough such that

$$\rho_A^{2(i+1)} \leq \min\left\{\frac{\sigma_{11}}{2b'}, \frac{\sigma_{12}^2}{\sigma_{11}b'}, \frac{\sigma_{21}^2}{(1-\lambda)c'}, \frac{\sigma_{22}^2}{(1-\lambda)c'}\right\}, \tag{3.224}$$

the inequality (3.221) becomes

$$\begin{bmatrix} \|\bar{x}_i\|^2 \\ \|\check{x}_i\|^2 \end{bmatrix} \preceq \underbrace{\begin{bmatrix} 1-\frac{\sigma_{11}\mu_{\max}}{2} & \frac{2\sigma_{12}^2}{\sigma_{11}}\mu_{\max} \\ \frac{3\sigma_{21}^2\mu_{\max}^2}{1-\lambda} & \lambda+\frac{3\sigma_{22}^2\mu_{\max}^2}{1-\lambda} \end{bmatrix}}_{G'} \begin{bmatrix} \|\bar{x}_{i-1}\|^2 \\ \|\check{x}_{i-1}\|^2 \end{bmatrix} + \begin{bmatrix} d'\mu_{\max} \\ e'\mu_{\max}^2 \end{bmatrix} \rho_A^{2(i+1)}, \tag{3.225}$$

where we can prove $\rho \triangleq \|G'\|_1 = 1-O(\mu_{\max}) < 1$ by following arguments (3.185). Inequality (3.225) further implies that

$$\left(\|\bar{x}_i\|^2+\|\check{x}_i\|^2\right) \leq \rho\left(\|\bar{x}_{i-1}\|^2+\|\check{x}_{i-1}\|^2\right) + f'\rho_A^{2(i+1)} \tag{3.226}$$

where $f' \triangleq d'\mu_{\max} + e'\mu_{\max}^2 > 0$. Let $\beta = \max\{\rho, \rho_A\} < 1$. Inequality (3.226) becomes

$$\left(\|\bar{x}_i\|^2+\|\check{x}_i\|^2\right) \leq \beta\left(\|\bar{x}_{i-1}\|^2+\|\check{x}_{i-1}\|^2\right)+f'\beta^{2(i+1)}. \tag{3.227}$$

By adding $\gamma f'\beta^{2i+4}$, where γ can be any positive constant to be chosen, to both sides of the above inequality, we get

$$\begin{aligned}
&\left(\|\bar{x}_i\|^2 + \|\check{x}_i\|^2\right) + \gamma f'\beta^{2i+4} \\
&\quad \leq \beta\left(\|\bar{x}_{i-1}\|^2 + \|\check{x}_{i-1}\|^2\right) + f'\beta^{2i+2} + \gamma f'\beta^{2i+4} \\
&\quad = \beta\left(\|\bar{x}_{i-1}\|^2 + \|\check{x}_{i-1}\|^2 + \frac{1+\gamma\beta^2}{\beta} f'\beta^{2i+2}\right)
\end{aligned} \tag{3.228}$$

By setting

$$\gamma = \frac{1}{\beta - \beta^2} > 0, \tag{3.229}$$

it can be verified that

$$\gamma = \frac{1+\gamma\beta^2}{\beta}. \tag{3.230}$$

Substituting (3.230) into (3.228), we have

$$\begin{aligned}
&\left(\|\bar{x}_i\|^2 + \|\check{x}_i\|^2\right) + \gamma f'\beta^{2(i+2)} \\
&\quad \leq \beta\left(\|\bar{x}_{i-1}\|^2 + \|\check{x}_{i-1}\|^2 + \gamma f'\beta^{2(i+1)}\right).
\end{aligned} \tag{3.231}$$

As a result, the quantity $(\|\bar{x}_i\|^2 + \|\check{x}_i\|^2) + \gamma f'\beta^{2(i+2)}$ converges to 0 linearly. Since $f' > 0, \gamma > 0$ and $\beta > 0$, we can conclude that $\|\bar{x}_i\|^2 + \|\check{x}_i\|^2$, and hence $\|\widetilde{w}_i\|^2 + \|\widetilde{\mathcal{Y}}_i\|^2$, converges to 0 linearly.

3.D Error Recursion for EXTRA Consensus

Multiplying the second recursion of (3.66) by $\mathcal{V}$ gives:

$$\mathcal{V}\mathcal{Y}_i^e = \mathcal{V}\mathcal{Y}_{i-1}^e + \frac{\mathcal{P} - \mathcal{P}\mathcal{A}}{2} w_i^e. \tag{3.232}$$

Substituting into the first recursion of (3.66) gives

$$\overline{\mathcal{A}} w_i^e = \overline{\mathcal{A}} w_{i-1}^e - \mu\nabla\mathcal{J}^o(w_{i-1}^e) - \mathcal{P}^{-1}\mathcal{V}\mathcal{Y}_i^e, \tag{3.233}$$

From (3.233) and the second recursion in (3.66) we conclude that

$$\begin{cases} \overline{\mathcal{A}}\widetilde{w}_i^e = \overline{\mathcal{A}}\widetilde{w}_{i-1}^e + \mu\nabla\mathcal{J}^o(w_{i-1}^e) + \mathcal{P}^{-1}\mathcal{V}y_i^e, \\ \widetilde{y}_i^e = \widetilde{y}_{i-1}^e - \mathcal{V}w_i^e. \end{cases} \tag{3.234}$$

Subtracting the optimality condition (3.68)–(3.69) from (3.234) leads to

$$\begin{cases} \overline{\mathcal{A}}\widetilde{w}_i^e = (\overline{\mathcal{A}} - \mu\mathcal{H}_{i-1})\widetilde{w}_{i-1}^e - \mathcal{P}^{-1}\mathcal{V}\widetilde{y}_i^e, \\ \widetilde{y}_i^e = \widetilde{y}_{i-1}^e + \mathcal{V}\widetilde{w}_i^e. \end{cases} \tag{3.235}$$

which is also equivalent to

$$\begin{bmatrix} \overline{\mathcal{A}} & \mathcal{P}^{-1}\mathcal{V} \\ -\mathcal{V} & I_{MK} \end{bmatrix} \begin{bmatrix} \widetilde{w}_i^e \\ \widetilde{y}_i^e \end{bmatrix} = \begin{bmatrix} \overline{\mathcal{A}} - \mu\mathcal{H}_{i-1} & 0 \\ 0 & I_{MK} \end{bmatrix} \begin{bmatrix} \widetilde{w}_{i-1}^e \\ \widetilde{y}_{i-1}^e \end{bmatrix}. \tag{3.236}$$

Using relations $\overline{\mathcal{A}} = \frac{I_{MK}+\mathcal{A}}{2}$ and $\mathcal{V}^2 = \frac{\mathcal{P}-\mathcal{P}\mathcal{A}}{2}$, it is easy to verify that

$$\begin{bmatrix} \overline{\mathcal{A}} & \mathcal{P}^{-1}\mathcal{V} \\ -\mathcal{V} & I_{MK} \end{bmatrix}^{-1} = \begin{bmatrix} I_{MK} & -\mathcal{P}^{-1}\mathcal{V} \\ \mathcal{V} & I_{MK} - \mathcal{V}\mathcal{P}^{-1}\mathcal{V} \end{bmatrix}. \tag{3.237}$$

Substituting (3.237) into (3.236) gives (3.71)–(3.72).

3.E Error Recursion in Transformed Domain

Multiplying both sides of (3.71) by $(\mathcal{X}')^{-1}$:

$$(\mathcal{X}')^{-1} \begin{bmatrix} \widetilde{w}_i^e \\ \widetilde{y}_i^e \end{bmatrix} = [(\mathcal{X}')^{-1}(\mathcal{B}^e - \mathcal{T}_{i-1}^e)\mathcal{X}'](\mathcal{X}')^{-1} \begin{bmatrix} \widetilde{w}_{i-1}^e \\ \widetilde{y}_{i-1}^e \end{bmatrix} \tag{3.238}$$

leads to

$$\begin{bmatrix} \bar{x}_i^e \\ \widehat{x}_i^e \\ \check{x}_i^e \end{bmatrix} = \left(\begin{bmatrix} I_M & 0 & 0 \\ 0 & I_M & 0 \\ 0 & 0 & \mathcal{D}_1 \end{bmatrix} - \mathcal{S}_{i-1}^e \right) \begin{bmatrix} \bar{x}_{i-1}^e \\ \widehat{x}_{i-1}^e \\ \check{x}_{i-1}^e \end{bmatrix}, \tag{3.239}$$

where we defined

$$\begin{bmatrix} \bar{x}_i^e \\ \widehat{x}_i^e \\ \check{x}_i^e \end{bmatrix} \triangleq (\mathcal{X}')^{-1} \begin{bmatrix} \widetilde{w}_i^e \\ \widetilde{y}_i^e \end{bmatrix} = \begin{bmatrix} \mathcal{L}_1^{\mathsf{T}} \\ \mathcal{L}_2^{\mathsf{T}} \\ \mathcal{X}_L \end{bmatrix} \begin{bmatrix} \widetilde{w}_i^e \\ \widetilde{y}_i^e \end{bmatrix}, \tag{3.240}$$

and

$$\begin{aligned}\mathcal{S}_{i-1}^{e} &\triangleq (\mathcal{X}')^{-1}\mathcal{T}_{i-1}^{e}\mathcal{X}' \\ &= \begin{bmatrix} \mathcal{L}_1^{\mathsf{T}}\mathcal{T}_{i-1}^{e}\mathcal{R}_1 & \mathcal{L}_1^{\mathsf{T}}\mathcal{T}_{i-1}^{e}\mathcal{R}_2 & \frac{1}{c}\mathcal{L}_1^{\mathsf{T}}\mathcal{T}_{i-1}^{e}\mathcal{X}_R \\ \mathcal{L}_2^{\mathsf{T}}\mathcal{T}_{i-1}^{e}\mathcal{R}_1 & \mathcal{L}_2^{\mathsf{T}}\mathcal{T}_{i-1}^{e}\mathcal{R}_2 & \frac{1}{c}\mathcal{L}_2^{\mathsf{T}}\mathcal{T}_{i-1}^{e}\mathcal{X}_R \\ c\mathcal{X}_L\mathcal{T}_{i-1}^{e}\mathcal{R}_1 & c\mathcal{X}_L\mathcal{T}_{i-1}^{e}\mathcal{R}_2 & \mathcal{X}_L\mathcal{T}_{i-1}^{e}\mathcal{X}_R \end{bmatrix}. \end{aligned} \tag{3.241}$$

To compute each entry of $\mathcal{S}_{i-1}^{e}$, we let

$$\mathcal{X}_R = \begin{bmatrix} \mathcal{X}_{R,u} \\ \mathcal{X}_{R,d} \end{bmatrix}, \tag{3.242}$$

where $\mathcal{X}_{R,u} \in \mathbb{R}^{KM\times 2(K-1)M}$ and $\mathcal{X}_{R,d} \in \mathbb{R}^{KM\times 2(K-1)M}$. For the first line of $\mathcal{S}_{i-1}^{e}$, it can be verified that

$$\mathcal{L}_1^{\mathsf{T}}\mathcal{T}_{i-1}^{e}\mathcal{R}_1 = \mu\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{I}, \tag{3.243}$$

$$\mathcal{L}_1^{\mathsf{T}}\mathcal{T}_{i-1}^{e}\mathcal{R}_2 = 0, \tag{3.244}$$

$$\frac{1}{c}\mathcal{L}_1^{\mathsf{T}}\mathcal{T}_{i-1}^{e}\mathcal{X}_R = \frac{\mu}{c}\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{X}_{R,u}. \tag{3.245}$$

Likewise, noting that

$$\mathcal{L}_2^{\mathsf{T}}\mathcal{T}_{i-1}^{e} = \begin{bmatrix} 0 & \frac{1}{K}\mathcal{I}^{\mathsf{T}} \end{bmatrix}\begin{bmatrix} \mu\mathcal{H}_{i-1} & 0 \\ \mu\mathcal{V}\mathcal{H}_{i-1} & 0 \end{bmatrix} \overset{(2.67)}{=} \begin{bmatrix} 0 & 0 \end{bmatrix}, \tag{3.246}$$

we find for the second line of $\mathcal{S}_{i-1}^{e}$ that

$$c\mathcal{L}_2^{\mathsf{T}}\mathcal{T}_{i-1}^{e}\mathcal{R}_1 = 0, \;\; c\mathcal{L}_2^{\mathsf{T}}\mathcal{T}_{i-1}^{e}\mathcal{R}_2 = 0, \;\; \mathcal{L}_2^{\mathsf{T}}\mathcal{T}_{i-1}^{e}\mathcal{X}_R = 0. \tag{3.247}$$

Substituting (3.241), (3.243) and (3.247) into (3.239), we rewrite (3.239) as

$$\begin{bmatrix} \bar{x}_i^e \\ \widehat{x}_i^e \\ \check{x}_i^e \end{bmatrix} = \begin{bmatrix} I_M - \mu\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{I} & 0 & -\frac{\mu}{c}\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{X}_{R,u} \\ 0 & I_M & 0 \\ -c\mathcal{X}_L\mathcal{T}_{i-1}^{e}\mathcal{R}_1 & -c\mathcal{X}_L\mathcal{T}_{i-1}^{e}\mathcal{R}_2 & \mathcal{D}_1 - \mathcal{X}_L\mathcal{T}_{i-1}^{e}\mathcal{X}_R \end{bmatrix}\begin{bmatrix} \bar{x}_{i-1}^e \\ \widehat{x}_{i-1}^e \\ \check{x}_{i-1}^e \end{bmatrix} \tag{3.248}$$

From the second line of (3.248), we get

$$\widehat{x}_i^e = \widehat{x}_{i-1}^e. \tag{3.249}$$

As a result, $\widehat{x}_i^e$ will converge to 0 only if the initial value $\widehat{x}_0^e = 0$. To verify that, from the definition of $\mathcal{L}_2$ in (3.40) and (3.240) we have

$$\widehat{x}_0^e = \mathcal{L}_2^{\mathsf{T}} \begin{bmatrix} \widetilde{\mathcal{W}}_0^e \\ \widetilde{\mathcal{Y}}_0^e \end{bmatrix} = \frac{1}{K}\mathcal{I}^{\mathsf{T}}\widetilde{\mathcal{Y}}_0^e$$

$$\stackrel{(3.70)}{=} \frac{1}{K}\mathcal{I}^{\mathsf{T}}(\mathcal{Y}_o^\star - \mathcal{Y}_0^e) \stackrel{(3.67)}{=} \frac{1}{K}\mathcal{I}^{\mathsf{T}}(\mathcal{Y}_o^\star - \mathcal{V}\mathcal{W}_0^e). \tag{3.250}$$

Recall that $\mathcal{Y}_o^\star$ lies in the $\mathcal{R}(\mathcal{V})$, so that $\mathcal{Y}_o^\star - \mathcal{V}\mathcal{W}_0$ also lies in $\mathcal{R}(\mathcal{V})$. Recall further from Lemma 2.4 that $\mathcal{I}^{\mathsf{T}}\mathcal{V} = 0$, and conclude that $\widehat{x}_0^e = 0$. Therefore, from (3.249) we have

$$\widehat{x}_i^e = 0, \quad \forall i \geq 0 \tag{3.251}$$

With (3.251), recursion (3.248) is equivalent to (3.73).

3.F Proof of Theorem 3.3

From the first line of recursion (3.73), we have

$$\bar{x}_i^e = \left(I_M - \mu\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{I}\right)\bar{x}_{i-1}^e - \frac{\mu}{c}\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{X}_{R,u}\check{x}_{i-1}^e. \tag{3.252}$$

Squaring both sides and using Jensen's inequality gives

$$\begin{aligned} \|\bar{x}_i^e\|^2 &= \left\|\left(I_M - \mu\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{I}\right)\bar{x}_{i-1}^e - \frac{\mu}{c}\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{X}_{R,u}\check{x}_{i-1}^e\right\|^2 \\ &\leq \frac{1}{1-t}\left\|I_M - \mu\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{I}\right\|^2 \|\bar{x}_{i-1}^e\|^2 \\ &\quad + \frac{1}{tc^2}\|\mu\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{X}_{R,u}\|^2\|\check{x}_{i-1}^e\|^2 \end{aligned} \tag{3.253}$$

for any $t \in (0,1)$. For the term $\mu\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{I}$, we have

$$\mu\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{I} = \mu\sum_{k=1}^K p_k H_{k,i-1} \stackrel{(3.31)}{\geq} \frac{\mu}{K}\nu I_M \;\triangleq\; \sigma_{11}^e \mu I_M, \tag{3.254}$$

where $\sigma_{11} = \nu/N$. Similarly, we can obtain the upper bound

$$\mu\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{I} = \mu\sum_{k=1}^K p_k H_{k,i-1} \stackrel{(3.31)}{\leq} \left(\sum_{k=1}^K p_k\right)\delta\mu I_M \stackrel{(a)}{=} \delta\mu I_M, \tag{3.255}$$

where equality (a) holds because $\sum_{k=1}^{K} p_k = 1$. It is obvious that $\delta > \sigma_{11}^e$. As a result, we have

$$(1-\delta\mu)I_M \leq I_M - \mu\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{I} \leq (1-\sigma_{11}^e\mu)I_M, \tag{3.256}$$

which implies that when the step-size is sufficiently small to satisfy

$$\mu < 1/\delta, \tag{3.257}$$

it will hold that

$$\left\| I_M - \mu\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{I} \right\|^2 \leq (1-\sigma_{11}^e\mu_{\max})^2. \tag{3.258}$$

On the other hand, we have

$$\begin{aligned}
&\frac{1}{c^2}\|\mu\overline{\mathcal{P}}^{\mathsf{T}}\mathcal{H}_{i-1}\mathcal{X}_{R,u}\|^2 \\
&\leq \frac{\mu^2}{c^2}\|\overline{\mathcal{P}}^{\mathsf{T}}\|^2\|\mathcal{H}_{i-1}\|^2\|\mathcal{X}_{R,u}\|^2 \\
&\leq \frac{1}{c^2}\left(\sum_{k=1}^{K} p_k^2\right)\delta^2\|\mathcal{X}_{R,u}\|^2\mu^2 \\
&= \frac{\delta^2}{c^2K}\|\mathcal{X}_{R,u}\|^2\mu^2 \overset{(3.152)}{\leq} \frac{\delta^2}{c^2K}\|\mathcal{X}_R\|^2\mu^2 \;\overset{\Delta}{=}\; (\sigma_{12}^e)^2\mu^2,
\end{aligned} \tag{3.259}$$

where $\sigma_{12}^e = \delta\|\mathcal{X}_R\|/(c\sqrt{K})$ and the "=" sign in the third line holds because $p_k = 1/N$. Notice that σ_{12}^e is independent of μ. Substituting (3.258) and (3.259) into (3.253), we get

$$\begin{aligned}
&\|\bar{x}_i^e\|^2 \\
&\leq \frac{1}{1-t}(1-\sigma_{11}^e\mu)^2\|\bar{x}_{i-1}^e\|^2 + \frac{1}{t}(\sigma_{12}^e)^2\mu^2\|\check{x}_{i-1}^e\|^2 \\
&= (1-\sigma_{11}^e\mu)\|\bar{x}_{i-1}^e\|^2 + \frac{(\sigma_{12}^e)^2}{\sigma_{11}^e}\mu\|\check{x}_{i-1}^e\|^2,
\end{aligned} \tag{3.260}$$

where we are selecting $t = \sigma_{11}^e\mu$.

Next we check the second line of recursion (3.73), which amounts to

$$\begin{aligned}
\check{x}_i^e &= -c\mathcal{X}_L\mathcal{T}_{i-1}^e\mathcal{R}_1\bar{x}_{i-1}^e + (\mathcal{D}_1 - \mathcal{X}_L\mathcal{T}_{i-1}^e\mathcal{X}_R)\check{x}_{i-1}^e \\
&= \mathcal{D}_1\check{x}_{i-1}^e - \mathcal{X}_L\mathcal{T}_{i-1}^e(c\mathcal{R}_1\bar{x}_{i-1}^e + \mathcal{X}_R\check{x}_{i-1}^e).
\end{aligned} \tag{3.261}$$

Squaring both sides of (3.261), and using Jensen's inequality again,

$$\begin{aligned}\|\check{x}_i^e\|^2 =&\|\mathcal{D}_1\check{x}_{i-1}^e - \mathcal{X}_L\mathcal{T}_{i-1}^e(c\mathcal{R}_1\bar{x}_{i-1}^e + \mathcal{X}_R\check{x}_{i-1}^e)\|^2\\ \leq&\frac{\|\mathcal{D}_1\|^2}{t}\|\check{x}_{i-1}^e\|^2+\frac{1}{1-t}\|\mathcal{X}_L\mathcal{T}_{i-1}^e(c\mathcal{R}_1\bar{x}_{i-1}^e + \mathcal{X}_R\check{x}_{i-1}^e)\|^2\\ \leq&\frac{\|\mathcal{D}_1\|^2}{t}\|\check{x}_{i-1}^e\|^2 + \frac{2c^2}{1-t}\|\mathcal{X}_L\mathcal{T}_{i-1}^e\mathcal{R}_1\|^2\|\bar{x}_{i-1}^e\|^2\\ &+\frac{2}{1-t}\|\mathcal{X}_L\mathcal{T}_{i-1}^e\mathcal{X}_R\|^2\|\check{x}_{i-1}^e\|^2. \end{aligned}\tag{3.262}$$

where $t \in (0,1)$. From Lemma 3.3 we have that $\lambda \triangleq \|D_1\| = \sqrt{\lambda_2(\widetilde{A})} < 1$. By setting $t = \lambda$, we reach

$$\begin{aligned}\|\check{x}_i^e\|^2 \leq \lambda\|\check{x}_{i-1}^e\|^2 + \frac{2c^2}{1-\lambda}\|\mathcal{X}_L\mathcal{T}_{i-1}^e\mathcal{R}_1\|^2\|\bar{x}_{i-1}^e\|^2\\ +\frac{2}{1-\lambda}\|\mathcal{X}_L\mathcal{T}_{i-1}^e\mathcal{X}_R\|^2\|\check{x}_{i-1}^2\|^2. \end{aligned}\tag{3.263}$$

From the definition of $\mathcal{T}_{i-1}^e$ in (3.72), we have

$$\mathcal{T}_{i-1}^e=\mu\begin{bmatrix}\mathcal{H}_{i-1} & 0\\ \mathcal{V}\mathcal{H}_{i-1} & 0\end{bmatrix}=\underbrace{\mu\begin{bmatrix}I_{MK} & 0\\ \mathcal{V} & 0\end{bmatrix}}_{\triangleq\,\mathcal{T}_e}\begin{bmatrix}\mathcal{H}_{i-1} & 0\\ 0 & \mathcal{H}_{i-1}\end{bmatrix},\tag{3.264}$$

which implies that

$$\|\mathcal{T}_{i-1}^e\|^2 \leq \mu^2\|\mathcal{T}_e\|^2\left(\max_{1\leq k\leq K}\|H_{k,i-1}\|^2\right) \leq \|\mathcal{T}_e\|^2\delta^2\mu^2.\tag{3.265}$$

We also emphasize that $\|\mathcal{T}_e\|^2$ is independent of μ. With inequality (3.265), we further have

$$\begin{aligned}c^2\|\mathcal{X}_L\mathcal{T}_{i-1}^e\mathcal{R}_1\|^2 &\leq c^2\mu^2\|\mathcal{X}_L\|^2\|\mathcal{T}_e\|^2\|\mathcal{R}_1\|^2\delta^2\\ &\triangleq (\sigma_{21}^e)^2\mu^2 \end{aligned}\tag{3.266}$$

$$\begin{aligned}\|\mathcal{X}_L\mathcal{T}_{i-1}^e\mathcal{X}_R\|^2 &\leq \mu^2\|\mathcal{X}_L\|^2\|\mathcal{T}_e\|^2\|\mathcal{X}_R\|^2\delta^2\\ &\triangleq (\sigma_{22}^e)^2\mu^2, \end{aligned}\tag{3.267}$$

notice that $\|\mathcal{R}_1\| = 1$, σ_{21}^e and σ_{22}^e are defined as

$$\sigma_{21}^e = c\|\mathcal{X}_L\|\|\mathcal{T}_e\|\delta, \quad \sigma_{22}^e = \|\mathcal{X}_L\|\|\mathcal{T}_e\|\|\mathcal{X}_R\|\delta.\tag{3.268}$$

With (3.266) and (3.267), inequality (3.263) becomes

$$\|\check{x}_i^e\|^2 \leq \left(\lambda + \frac{2(\sigma_{22}^e)^2\mu^2}{1-\lambda}\right)\|\check{x}_{i-1}^e\|^2 + \frac{2(\sigma_{21}^e)^2\mu^2}{1-\lambda}\|\bar{x}_{i-1}^e\|^2. \tag{3.269}$$

Combining (3.260) and (3.269), we arrive at the inequality recursion:

$$\begin{bmatrix} \|\bar{x}_i^e\|^2 \\ \|\check{x}_i^e\|^2 \end{bmatrix} \preceq \underbrace{\begin{bmatrix} 1-\sigma_{11}^e\mu & \frac{(\sigma_{12}^e)^2}{\sigma_{11}^e}\mu \\ \frac{2(\sigma_{21}^e)^2\mu^2}{1-\lambda} & \lambda + \frac{2(\sigma_{22}^e)^2\mu^2}{1-\lambda} \end{bmatrix}}_{\triangleq G_e} \begin{bmatrix} \|\bar{x}_{i-1}^e\|^2 \\ \|\check{x}_{i-1}^e\|^2 \end{bmatrix}. \tag{3.270}$$

From this point onwards, we follow exactly the same argument as in (3.166)–(3.191) to arrive at the conclusion in Theorem 3.3.

3.G Proof of Lemma 3.5

It is observed from expression (3.107) for E_d that one of the eigenvalues is $1-\mu\sigma^2$. It is easy to verify that when μ satisfies (3.117), it holds that $-1 < 1-\mu\sigma^2 < 1$. Next, we check the other two eigenvalues. Let θ denote a generic eigenvalue of E_d. From the right-bottom 2×2 block of E_d in (3.107), we know that θ will satisfy the following characteristic polynomial

$$\theta^2 - (2-\mu\sigma^2)a\,\theta + (1-\mu\sigma^2)a = 0, \tag{3.271}$$

where $a \in (0,1)$ is a combination weight (see the expression for A in (3.99)). Solving (3.271), the two roots are

$$\theta_{1,2} = \frac{(2-\mu\sigma^2)a \pm \sqrt{(2-\mu\sigma^2)^2a^2 - 4(1-\mu\sigma^2)a}}{2}. \tag{3.272}$$

Let

$$\Delta = (2-\mu\sigma^2)^2a^2 - 4(1-\mu\sigma^2)a. \tag{3.273}$$

Based on the value of $\mu\sigma^2$ and a, Δ can be negative, zero, or positive. Recall from (3.117) that $0 < \mu\sigma^2 < 2$. In that case, over the smaller interval $1 \leq \mu\sigma^2 < 2$, it holds that $(1-\mu\sigma^2) \geq 0$ and, from (3.273), $\Delta > 0$. For this reason, as indicated in cases 1 and 2 below, the scenarios corresponding to $\Delta < 0$ or $\Delta = 0$ can only occur over $0 < \mu\sigma^2 < 1$:

Case 1: $\Delta < 0$. It can be verified that when

$$1 - \mu\sigma^2 > 0, \quad \text{and} \quad a < \frac{4(1-\mu\sigma^2)}{(2-\mu\sigma^2)^2}, \tag{3.274}$$

it holds that $\Delta < 0$. In this situation, both θ_1 and θ_2 are imaginary numbers with magnitude

$$|\theta_1| = |\theta_2| = \frac{1}{4}\left((2-\mu\sigma^2)^2a^2 + (-\Delta)\right) = (1-\mu\sigma^2)a < 1, \tag{3.275}$$

where the last inequality holds because $0 < \mu\sigma^2 < 1$ (see (3.117) and (3.274)) and $a \in (0,1)$.

Case 2: $\Delta = 0$. It can be verified that when

$$1 - \mu\sigma^2 > 0, \quad \text{and} \quad a = \frac{4(1-\mu\sigma^2)}{(2-\mu\sigma^2)^2}, \tag{3.276}$$

it holds that $\Delta = 0$. In this situation, from (3.272) we have

$$\theta_1 = \theta_2 = \frac{(2-\mu\sigma^2)a}{2} < 1, \tag{3.277}$$

where the last inequality holds because $0 < \mu\sigma^2 < 1$ (see (3.117) and (3.274)) and $a \in (0,1)$. Observe further that the upper bound on a in (3.274) is positive and smaller than one when $0 < \mu\sigma^2 < 1$.

Case 3: $\Delta > 0$. It can be verified that when

$$1 - \mu\sigma^2 > 0, \quad \text{and} \quad a > \frac{4(1-\mu\sigma^2)}{(2-\mu\sigma^2)^2}, \tag{3.278}$$

or when $1 \leq \mu\sigma^2 < 2$, it holds that $\Delta > 0$. In this situation, θ is real and

$$\theta_1 = \frac{(2-\mu\sigma^2)a + \sqrt{(2-\mu\sigma^2)^2a^2 - 4(1-\mu\sigma^2)a}}{2}, \tag{3.279}$$

$$\theta_2 = \frac{(2-\mu\sigma^2)a - \sqrt{(2-\mu\sigma^2)^2a^2 - 4(1-\mu\sigma^2)a}}{2}. \tag{3.280}$$

Moreover, since $(2-\mu\sigma^2)a > 0$, we have

$$|\theta_2| < |\theta_1| = \theta_1. \tag{3.281}$$

We regard θ_1 as a function of a, i.e., $\theta_1 = f(a)$. It holds that $f(a)$ is monotone increasing with a. To prove it, note that

$$f'(a) = \frac{2-\mu\sigma^2}{2} + \frac{2(2-\mu\sigma^2)a - 4(1-\mu\sigma^2)}{4\sqrt{\Delta}}. \tag{3.282}$$

Now since

$$\begin{aligned}
&\Delta = (2-\mu\sigma^2)^2a^2-4(1-\mu\sigma^2)a > 0 \\
\iff &(2-\mu\sigma^2)^2a > 4(1-\mu\sigma^2) \quad (\text{because } a > 0) \\
\implies &2(2-\mu\sigma^2)a > 4(1-\mu\sigma^2),
\end{aligned} \tag{3.283}$$

we conclude that $f'(a) > 0$. Since $a < 1$, it follows that

$$\theta_1 = f(a) < f(1) = 1. \tag{3.284}$$

In summary, when μ satisfies (3.117), for any $a \in (0,1)$ it holds that all three eigenvalues of E_d stay within the unit-circle, which implies that $\rho(E_d) < 1$, and also $\rho(\mathcal{E}_d) < 1$. As a result, $\check{z}_i$ in (3.116) will converge to 0. Since $\widehat{z}_i = 0$ for any i, we conclude that $\widetilde{z}_i$ converges to 0.

3.H Proof of Lemma 3.6

Similar to the arguments used to establish Lemma 3.4 and (3.110)–(3.116), the EXTRA error recursion (3.102) can also be divided into two separate recursions

$$\widehat{z}_i^e = \widehat{z}_{i-1}^e, \quad \text{and} \quad \check{z}_i^e = \mathcal{E}_e\check{z}_{i-1}^e, \tag{3.285}$$

where $\mathcal{E}_e = E_e \otimes I_M$, and

$$E_e = \begin{bmatrix} 1-\mu^e\sigma^2 & 0 & 0 \\ 0 & a-\mu^e\sigma^2 & -\sqrt{2-2a} \\ 0 & (a-\mu^e\sigma^2)\sqrt{\frac{1-a}{2}} & a \end{bmatrix}. \tag{3.286}$$

Also, since both $\mathcal{Y}_0^e$ and $\mathcal{Y}_o^\star$ lie in the range$(\mathcal{V})$, it can be verified that $\widehat{z}_0^e = 0$. Therefore, we only focus on the convergence of $\check{z}_i^e$. Let θ^e denote a generic eigenvalue of E_e. From the right-bottom 2×2 block of E_e in (3.286), we know that θ^e will satisfy the following characteristic polynomial

$$(\theta^e)^2 - (2a-\mu^e\sigma^2)(\theta^e) + (a-\mu^e\sigma^2) = 0. \tag{3.287}$$

Solving it, we have

$$\theta^e_{1,2} = \frac{2a - \mu^e\sigma^2 \pm \sqrt{(2a - \mu^e\sigma^2)^2 - 4(a - \mu^e\sigma^2)}}{2}. \tag{3.288}$$

Now we suppose $\mu^e\sigma^2 \geq a + 1$ as noted in (3.118), it then follows that

$$a - \mu^e\sigma^2 \leq -1 \tag{3.289}$$

and hence both θ^e_1 and θ^e_2 are real numbers with

$$\theta^e_1 = \frac{2a - \mu^e\sigma^2 + \sqrt{(2a - \mu^e\sigma^2)^2 + 4(\mu^e\sigma^2 - a)}}{2}, \tag{3.290}$$

$$\theta^e_2 = \frac{2a - \mu^e\sigma^2 - \sqrt{(2a - \mu^e\sigma^2)^2 + 4(\mu^e\sigma^2 - a)}}{2}. \tag{3.291}$$

Moreover, with $\mu^e\sigma^2 \geq a + 1$ we further have

$$2a - \mu^e\sigma^2 \leq a - 1 < 0, \tag{3.292}$$

which implies that

$$|\theta^e_2| = \frac{\mu^e\sigma^2 - 2a + \sqrt{(2a - \mu^e\sigma^2)^2 + 4(\mu^e\sigma^2 - a)}}{2} > 1, \tag{3.293}$$

where the last inequality holds because of (3.289) and (3.292). Therefore, when μ^e is chosen such that $\mu^e\sigma^2 \geq 1 + a$, there always exists one eigenvalue θ^e such that $|\theta^e| > 1$ which implies that $\check{z}^e_i$ diverges, and so does $\widetilde{z}^e_i$.

CHAPTER 4

Exact Diffusion For Distributed Adaptation and Online Learning

4.1 Introduction

This chapter considers stochastic optimization problems where a collection of K networked agents work cooperatively to solve an aggregate optimization problem of the form:

$$w^\star = \arg\min_{w\in\mathbb{R}^M} \sum_{k=1}^{K} J_k(w), \text{ where } J_k(w) = \mathbb{E}\, Q(w;\boldsymbol{x}_k) \tag{4.1}$$

The local risk function $J_k(w)$ held by agent k is assumed to be differentiable and ν-strongly convex, and it is constructed as the expectation of some loss function $Q(w;\boldsymbol{x}_k)$. The random variable $\boldsymbol{x}_k$ represents the streaming data received by agent k, and the expectation in $J_k(w)$ is over the distribution of $\boldsymbol{x}_k$. While the cost functions $J_k(w)$ may have *different* local minimizers, all agents seek to determine the *common* global solution $w^\star$ under the constraint that agents can only communicate with their direct neighbors. Problem (4.1) can find applications in a wide range of areas including wireless sensor networks [20, 21], distributed statistical learning [13], and distributed adaptation and learning [1, 4, 14].

There are several techniques that can be used to solve problems of the type (4.1) such as diffusion [1, 4] and consensus (also known as decentralized gradient descent) [5, 9, 141, 142] strategies. The latter class of strategies has been shown to be particularly well-suited for stochastic and adaptive learning scenarios from streaming data due to their enhanced stability range over other methods, as well as their ability to track drifts in the underlying models and statistics [1, 4]. We therefore focus on this class of algorithms since we are mainly interested in methods that are able to learn and adapt from data. For example, the

adapt-then-combine (ATC) formulation [1, 4] of diffusion takes the following form:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu \nabla Q(\boldsymbol{w}_{k,i-1}; \boldsymbol{x}_{k,i}), \quad \text{(Adapt)} \tag{4.2}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\psi}_{\ell,i}, \quad \text{(Combine)} \tag{4.3}$$

where the subscript k denotes the agent index and i denotes the iteration index. The variable $\boldsymbol{x}_{k,i}$ is the data realization observed by agent k at iteration i. The nonnegative scalar $a_{\ell k}$ is the weight used by agent k to scale information received from agent ℓ, $\mathcal{N}_k$ is the set of neighbors of agent k (including k itself), and it is required that $\sum_{\ell \in \mathcal{N}_k} a_{\ell k} = 1$ for any k. In (4.2)–(4.3), variable $\boldsymbol{\psi}_{k,i}$ is an intermediate estimate for $w^\star$ at agent k, while $\boldsymbol{w}_{k,i}$ is the updated estimate. Note that step (4.2) uses the gradient of the loss function, $Q(\cdot)$, rather than the gradient of its expected value $J_k(w)$. This is because the statistical properties of the data are not known beforehand. If $J_k(w)$ were known, then we could use its gradient vector in (4.2). In that case, we would refer to the resulting method as a *deterministic* rather than *stochastic* solution. Throughout this chapter, we employ a *constant* step-size μ to enable continuous adaptation and learning in response to drifts of the global minimizer due to changes in the statistical properties of the data. The adaptation and tracking abilities are crucial in many applications, as already explained in [1].

Previous studies have shown that both consensus and diffusion methods are able to solve problems of the type (4.1) well for sufficiently small step-sizes. That is, the squared error $\mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$ approaches a small neighborhood around zero for all agents, where $\widetilde{\boldsymbol{w}}_{k,i} = w^\star - \boldsymbol{w}_{k,i}$. These methods do not converge to the *exact* minimizer $w^\star$ of (4.1) but rather approach a small neighborhood around $w^\star$ with a small steady-state bias under *both* stochastic and deterministic optimization scenarios. For example, in deterministic settings where the individual costs $J_k(w)$ are known, it is shown in [1, 14] that the squared errors $\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$ generated by the diffusion iterates converge to a $O(\mu^2)$-neighborhood. Note that, in the deterministic case, this inherent limiting bias is not due to any gradient noise arising from stochastic approximations; it is instead due to the update structure in diffusion and consensus implementations — see the explanations in Sec. III.B in [15]. For stochastic optimization problems, on the other hand, the size of the bias is $O(\mu)$ rather than $O(\mu^2)$ because of the gradient noise.

When high precision is desired, especially in deterministic optimization problems, it would be preferable to remove the $O(\mu^2)$ bias altogether. Motivated by these considerations, the works [15, 16] showed that a simple correction step inserted between the adaptation and combination steps (4.2) and (4.3) is sufficient to ensure *exact* convergence of the algorithm to $w^\star$ by all agents — see expression (4.10) further ahead. In this way, the $O(\mu^2)$ bias is removed completely, and the convergence rate is also improved.

While the correction of the second order $O(\mu^2)$ bias is critical in the deterministic setting, it is not clear *whether* it can help in the stochastic and adaptive settings. This motivates us to study exact diffusion under these settings in this paper and compare against standard diffusion. To this end, we carry out a *higher-order* analysis of the error dynamics for both methods, and derive their steady-state performance as an expansion in the first two powers of the step-size parameter, i.e., μ and μ^2. In contrast, traditional analysis for diffusion and consensus focus mainly on performance expressions that depend on a first-order expansion in μ [1, 4]. Our analysis will reveal conditions under which bias correction improves the performance of diffusion.

4.1.1 Main Results

In particular, we will prove in Theorem 4.1, that, with small step-sizes, the exact diffusion strategy will converge exponentially fast, at a rate $\rho = 1 - O(\mu\nu)$, to a neighborhood around $w^\star$ where ν is the strong convexity constant. Moreover, the size of the neighborhood will be characterized as

$$\limsup_{i\to\infty} \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|_{ed}^2 = O\left(\frac{\mu\sigma^2}{K} + \frac{\mu^2\sigma^2}{1-\lambda}\right) \tag{4.4}$$

where the quantity σ^2 is a measure of the variance of the gradient noise, and $\lambda \in (0, 1)$ is the second largest magnitude of the eigenvalues of the combination matrix $A = [a_{\ell k}]$ which reflects the level of network connectivity. The subscript ed indicates that $\boldsymbol{w}_{k,i}$ is generated by the exact diffusion method. In comparison, we will show that the traditional diffusion

strategy converges at a similar rate albeit to the following neighborhood:

$$\limsup_{i\to\infty} \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|_{\rm d}^2 = O\left(\frac{\mu\sigma^2}{K} + \frac{\mu^2\lambda^2\sigma^2}{1-\lambda} + \frac{\mu^2\lambda^2 b^2}{(1-\lambda)^2}\right) \tag{4.5}$$

where the subscript d indicates that $\boldsymbol{w}_{k,i}$ is generated by the diffusion method (4.2)–(4.3), and

$$b^2 = (1/K)\sum_{k=1}^{K} \|\nabla J_k(w^\star)\|^2 \tag{4.6}$$

is a bias constant independent of the gradient noise. Observe that the expressions on the right-hand side of (4.4) and (4.5) depend on μ and μ^2. These are therefore more refined performance expressions, which are more challenging to derive than earlier expressions that just depend on μ (see [1,4,5,9,14,141]). The terms that depend on μ^2 in (4.4) and (4.5) help reveal the following important insights that arise from using the exact diffusion strategy.

First, it is obvious that diffusion suffers from an additional bias term $\mu^2\lambda^2 b^2/(1-\lambda)^2$, which is independent of the gradient noise σ^2, while exact diffusion removes it completely. In the deterministic setting when the gradient noise $\sigma^2 = 0$, it is observed from (4.4) and (4.5) that diffusion converges to an $O(\mu^2)$-neighborhood around the global solution $w^\star$ while exact diffusion converges exactly to $w^\star$. This result is consistent with [1, 14, 16].

Second, it is further observed that the performance of diffusion and exact diffusion differs only on the $O(\mu^2)$ terms inside (4.4) and (4.5). When the step-size is moderately small so that these $O(\mu^2)$ terms are non-negligible, the superiority of exact diffusion or diffusion will highly depend on the network topology. In particular, when the network topology is sparsely-connected (in which case λ approaches 1), the bias term $\mu^2\lambda^2 b^2/(1-\lambda)^2$ will be significantly large and the correction of this term will greatly improve the steady-state performance. It should be emphasized that the bias-correction property of exact diffusion is particularly critical for large-scale *linear* or *cyclic* networks where $1-\lambda = O(1/K^2)$ and *grid* networks where $1-\lambda = O(1/K)$ since the bias term will grow rapidly on these network topologies as the size K increases. On the other hand, when the network is well-connected (in which case λ approaches 0), one can find that the $O(\mu^2)$ terms in diffusion (4.5) diminishes while the $O(\mu^2)$ term in exact diffusion (4.4) still exists. This implies that for well connected

Moderately small step-size μ				
Scenario	**Network**	**Diffusion**	**Exact Diffusion**	**Better Algorithm**
$b^2=0, \sigma^2 \neq 0$	Sparse ($\lambda \to 1$)	$O(\mu\sigma^2 + \frac{\mu^2\sigma^2}{1-\lambda})$	$O(\mu\sigma^2 + \frac{\mu^2\sigma^2}{1-\lambda})$	Similar performance
	Dense ($\lambda \to 0$)	$O(\mu\sigma^2)$	$O(\mu\sigma^2 + \mu^2\sigma^2)$	Diffusion
$b^2 \neq 0, \sigma^2 = 0$	Sparse ($\lambda \to 1$)	$O(\frac{\mu^2 b^2}{(1-\lambda)^2})$	0	Exact diffusion
	Dense ($\lambda \to 0$)	$O(\mu^2 b^2 \lambda^2)$	0	Exact diffusion†
$b^2 \neq 0, \sigma^2 \neq 0$	Sparse ($\lambda \to 1$)	$O\left(\mu\sigma^2 + \frac{\mu^2\sigma^2}{1-\lambda} + \frac{\mu^2 b^2}{(1-\lambda)^2}\right)$	$O\left(\mu\sigma^2 + \frac{\mu^2\sigma^2}{1-\lambda}\right)$	Exact diffusion
	Dense ($\lambda \to 0$)	$O\left(\mu\sigma^2\right)$	$O\left(\mu\sigma^2 + \frac{\mu^2\sigma^2}{1-\lambda}\right)$	Diffusion

Sufficiently small step-size μ				
Scenario	**Network**	**Diffusion**	**Exact Diffusion**	**Better Algorithm**
$b^2 \neq 0, \sigma^2 = 0$	Sparse or dense	$O(\mu^2 b^2 \lambda^2/(1-\lambda)^2)$	0	Exact diffusion†
All other scenarios	Sparse or dense	$O(\mu\sigma^2)$	$O(\mu\sigma^2)$	Similar performance

† Exact diffusion performs better unless $\lambda = 0$

Figure 4.1: Performance of Exact Diffusion and Diffusion under different scenarios

networks and moderatly-small step-sizes, diffusion will perform better than exact diffusion. The comparison between (4.4) and (4.5) provides guidelines on the proper choice of diffusion or exact diffusion in various application scenarios.

Third, the difference between exact diffusion and diffusion will vanish as the step-size μ approaches 0. This is because $O(\mu\sigma^2/K)$ will dominate the $O(\mu^2)$ terms when μ is sufficiently small. The "sufficiently" small μ can be roughly characterized as $\mu \leq c(1-\lambda)^{2+x}$, where x is any positive constant. This shows that diffusion and exact diffusion have the same upper bound for the steady-state performance (ignoring higher order step-sizes). However, sharing the same upper bound may not necessarily imply both algorithms perform the same. To more accurately characterize the steady-state performance of diffusion and exact diffusion when μ is sufficiently small, we shall establish the precise MSD expression defined as [1]

$$\text{MSD} = \mu \left(\lim_{\mu \to 0} \limsup_{i \to \infty} \frac{1}{\mu K} \sum_{k=1}^{K} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 \right) \tag{4.7}$$

for exact diffusion and find that it matches that of diffusion:

$$\text{MSD}_{ed} = \text{MSD}_{\text{d}} = \frac{\mu}{2K} \text{Tr}\left\{ \left(\sum_{k=1}^{K} H_k \right)^{-1} \left(\sum_{k=1}^{K} S_k \right) \right\}, \tag{4.8}$$

where $H_k = \nabla^2 J_k(w^\star)$ and S_k is the covariance matrix of gradient noise. Obviously, the MSD expression (4.7) is exact to first order in μ and ignores all higher-order terms. Equality

(4.8) states that when μ is sufficiently small, both diffusion and exact diffusion perform *exactly the same* during the steady-state stage. The main results derived in this chapter are summarized in Fig. 4.1 where we omit constant K for clarity.

4.1.2 Related work

In addition to exact diffusion, there exist some other useful bias-correction methods such as EXTRA [75, 91], gradient-tracking methods [93, 97, 98, 143, 144], Aug-DGM [95, 96] and NIDS [106]. All these methods converge linearly to the exact solution under the deterministic setting, but their performance (especially their advantage over diffusion or consensus) in the stochastic and adaptive settings remains unexplored and/or unclear. The recent work [2] studies the gradient-tracking method (referred to as DIGing in [93]) under the stochastic setting and shows that it can outperform the decentralized gradient descent (DGD) [5, 9] via numerical simulations. However, it does not analytically discuss *when* and *why* bias-correction methods can outperform consensus. Similarly, the work [3] also studies the stochastic gradient-tracking method [97, 98] and shows that it converges linearly around a neighborhood of the minimizer. No comparison with diffusion or consensus is presented in [3]. Another useful work is [145], which establishes the convergence property of exact diffusion with decaying step-sizes in the stochastic and non-convex setting. It proves exact diffusion is less sensitive to the data variance across the network than diffusion and is therefore endowed with a better convergence rate when the data variance is large. Different from [145], our bound in (4.5) shows that even small data variances (i.e., b^2) can be significantly amplified by a bad network connectivity – see the example graph topologies discussed in Sec. 4.4.2. This observation implies that the superiority of exact diffusion does not just rely on its robustness to data variance, but more importantly, on the network connectivity as well. In addition, different from the works [2, 145], which claim or suggest that the gradient-tracking method [2] or exact diffusion [145] always converges better than traditional DGD or diffusion, our current work disproves this statement and clarifies analytically that there are important scenarios where exact diffusion performs similarly or even worse than diffusion. Simulations also suggest that gradient tracking methods [2, 3] may also degrade the perfor-

mance of traditional diffusion, which was not explored prior to this work. Finally, we remark that work [146] showed that diffusion outperforms traditional primal-dual methods in the stochastic setting for $b^2 = 0$ and quadratic problems only, and is hence more restricted than our result. Our results recover this case (see Remark 4.2) and show that exact diffusion, which is also a primal-dual method, can outperform diffusion when $b^2 \neq 0$.

Notation. Throughout the paper we use $\mathrm{col}\{x_1, \cdots, x_K\}$ or $\mathrm{col}\{x_k\}_{k=1}^K$ and $\mathrm{diag}\{x_1, \cdots, x_K\}$ or $\mathrm{diag}\{x_k\}_{k=1}^K$ to denote a column vector and a diagonal matrix formed from $x_1, \cdots, x_K$. The notation $\mathbb{1}_K = \mathrm{col}\{1, \cdots, 1\} \in \mathbb{R}^K$ and $I_K \in \mathbb{R}^{K \times K}$ is an identity matrix. The Kronecker product is denoted by "$\otimes$". For two matrices X and Y, the notation $X \geq Y$ denotes $X - Y$ is nonnegative.

4.2 Exact Diffusion Strategy

The exact diffusion strategy from [15, 16] was originally proposed to solve deterministic optimization problems. We adapt it to solve stochastic optimization problems by replacing the gradient of the local cost $J_k(w)$ by the stochastic gradient of the corresponding loss function. That is, we now use:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu \nabla Q(\boldsymbol{w}_{k,i-1}; \boldsymbol{x}_{k,i}), \quad \text{(Adapt)} \tag{4.9}$$

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{\psi}_{k,i} + \boldsymbol{w}_{k,i-1} - \boldsymbol{\psi}_{k,i-1}, \quad \text{(Correct)} \tag{4.10}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} \bar{a}_{\ell k} \boldsymbol{\phi}_{\ell,i}. \quad \text{(Combine)} \tag{4.11}$$

For the initialization, we let $\boldsymbol{w}_{k,-1} = \boldsymbol{\psi}_{k,-1} = 0$. Observe that the fusion step (4.11) now employs the corrected iterates from (4.10) rather than the intermediate iterates from (4.9). Note that the weight $\bar{a}_{\ell k}$ is different from $a_{\ell k}$ used in the diffusion recursion (4.3). If we let $A = [a_{\ell k}] \in \mathbb{R}^{K \times K}$ and $\bar{A} = [\bar{a}_{\ell k}] \in \mathbb{R}^{K \times K}$ denote the combination matrices used in diffusion and exact diffusion respectively, then the relation between them is $\bar{A} = (A + I_K)/2$. In the paper, we assume A (and, hence, $\bar{A}$) to be symmetric and doubly stochastic (see Assumption 4.2).

As explained in [15, 16], exact diffusion is essentially a primal-dual method. To rewrite

the update (4.9)–(4.11) in a compact primal-dual form, we collect the iterates and gradients from across the network into global vectors. Specifically, we introduce

$$\boldsymbol{\mathcal{w}}_i = col\{\boldsymbol{w}_{k,i}\}_{k=1}^K \in \mathbb{R}^{KM}, \tag{4.12}$$

$$\nabla\mathcal{Q}(\boldsymbol{\mathcal{w}}_{i-1};\boldsymbol{\mathcal{x}}) = col\{\nabla Q(\boldsymbol{w}_{k,i-1};\boldsymbol{x}_{k,i})\}_{k=1}^K \in \mathbb{R}^{KM}, \tag{4.13}$$

$\mathcal{A} = A \otimes I_M$, and $\overline{\mathcal{A}} = (\mathcal{A} + I_{KM})/2$. Since the combination matrix $\bar{A}$ is symmetric and doubly stochastic, it holds that $I - \bar{A}$ is positive semi-definite. By introducing the eigen-decomposition $I - \bar{A} = U\Sigma U^{\mathsf{T}}$ and defining $V = U\Sigma^{1/2}U^{\mathsf{T}} \in \mathbb{R}^{K\times K}$, where Σ is a non-negative diagonal matrix, we know that V is also positive semi-definite and $V^2 = I - \bar{A}$. We further let $\mathcal{V} = V \otimes I_M$, which implies $\mathcal{V}^2 = I_{KM} - \overline{\mathcal{A}}$. With these relations, it can be verified[1] that recursion (4.9)–(4.11) is equivalent to [15]

$$\begin{cases} \boldsymbol{\mathcal{w}}_i = \overline{\mathcal{A}}\big(\boldsymbol{\mathcal{w}}_{i-1} - \mu\nabla\mathcal{Q}(\boldsymbol{\mathcal{w}}_{i-1};\boldsymbol{\mathcal{x}}_i)\big) - \mathcal{V}\boldsymbol{\mathcal{y}}_{i-1}, \\ \boldsymbol{\mathcal{y}}_i = \boldsymbol{\mathcal{y}}_{i-1} + \mathcal{V}\boldsymbol{\mathcal{w}}_i, \end{cases} \tag{4.14}$$

for $i \geq 0$ with $\mathcal{y}_{-1} = 0$ where $\mathcal{y}_i \in \mathbb{R}^{KM}$ is a dual variable. The analysis in [15,16] explains how the correction term in (4.10) guarantees *exact* convergence to $w^\star$ by all agents in deterministic optimization problems where the true gradient $\nabla J_k(w)$ is available. In the following sections, we examine the convergence of exact diffusion in the stochastic setting.

4.3 Error Dynamics of Exact Diffusion

To establish the error dynamics of exact diffusion, we first introduce some standard assumptions. These assumptions are common in the literature (e.g, [1, 2]).

Assumption 4.1 (Conditions on cost functions) *Each $J_k(w)$ is ν-strongly convex and twice differentiable, and its Hessian matrix satisfies*

$$\nu I_M \leq \nabla^2 J_k(w) \leq \delta I_M, \quad \forall\ k \tag{4.15}$$

where $\delta \geq \nu > 0$. ■

[1]To verify it, one can substitute the second recursion in (4.14) into the first recursion to remove $\boldsymbol{\mathcal{y}}_i$ and arrive at (4.9)–(4.11).

We remark that the twice differentiability assumption is necessary to derive the MSD expression in Sec. 4.5.

Assumption 4.2 (Conditions on combination matrix) *The network is undirected and strongly connected, and the combination matrix A is symmetric and doubly stochastic, i.e., it satisfies $A = A^{\mathsf{T}},\ A\mathbb{1}_K = \mathbb{1}_K$.* ■

Assumption 4.2 implies that $\bar{A} = (I + A)/2$ is also symmetric and doubly-stochastic. Since the network is strongly connected, it holds that $1 = \lambda_1(\bar{A}) > \lambda_2(\bar{A}) \geq \cdots \geq \lambda_K(\bar{A}) > 0$.

To establish the optimality condition for problem (4.1), we introduce the following notation:

$$\mathcal{W} = \mathrm{col}\{w_1, \cdots, w_K\} \in \mathbb{R}^{KM}, \tag{4.16}$$

$$\nabla\mathcal{J}(\mathcal{W}) = \mathrm{col}\{\nabla J_1(w_1), \cdots, \nabla J_K(w_K)\} \in \mathbb{R}^{KM}, \tag{4.17}$$

where w_k in (4.16) is the k-th block entry of vector w. With the above notation, the following lemma from [16] states the optimality condition for problem (4.1).

Lemma 4.1 (Optimality Condition) *Under Assumption 4.1, if some block vectors $(\mathcal{W}^\star, \mathcal{Y}^\star)$ exist that satisfy:*

$$\mu\overline{\mathcal{A}}\nabla\mathcal{J}(\mathcal{W}^\star) + \mathcal{V}\mathcal{Y}^\star = 0, \tag{4.18}$$

$$\mathcal{V}\mathcal{W}^\star = 0. \tag{4.19}$$

then it holds that each block entries in $\mathcal{W}^\star$ satisfy:

$$w_1^\star = w_2^\star = \cdots = w_N^\star = w^\star \tag{4.20}$$

where $w^\star$ is the unique solution to problem (4.1). ■

4.3.1 Error Dynamics

We define the gradient noise at agent k as

$$\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1}) \ \triangleq\ \nabla Q(\boldsymbol{w}_{k,i-1}; \boldsymbol{x}_{k,i}) - \nabla J_k(\boldsymbol{w}_{k,i-1}) \tag{4.21}$$

and collect them into the network vector

$$\boldsymbol{s}_i(\boldsymbol{\mathcal{w}}_{i-1}) = \mathrm{col}\{\boldsymbol{s}_{1,i}(\boldsymbol{w}_{1,i-1}), \cdots, \boldsymbol{s}_{K,i}(\boldsymbol{w}_{K,i-1})\} \tag{4.22}$$

$$\nabla\mathcal{J}(\boldsymbol{\mathcal{w}}_{i-1}) = \mathrm{col}\{\nabla J_1(\boldsymbol{w}_{1,i-1}), \cdots, \nabla J_K(\boldsymbol{w}_{K,i-1})\} \tag{4.23}$$

It then follows that

$$\nabla\mathcal{Q}(\boldsymbol{\mathcal{w}}_{i-1}; \boldsymbol{\mathcal{x}}_i) = \nabla\mathcal{J}(\boldsymbol{\mathcal{w}}_{i-1}) + \boldsymbol{s}_i(\boldsymbol{\mathcal{w}}_{i-1}). \tag{4.24}$$

Next, we introduce the error vectors

$$\widetilde{\boldsymbol{\mathcal{w}}}_i = w^\star - \boldsymbol{\mathcal{w}}_i, \quad \widetilde{\boldsymbol{\mathcal{y}}}_i = \mathcal{y}^\star - \boldsymbol{\mathcal{y}}_i \tag{4.25}$$

where $(w^\star, \mathcal{y}^\star)$ are optimal solutions satisfying (3.1)–(3.2). By combining (4.14), (3.1), (3.2), (4.24) and (4.25), we reach

$$\begin{cases} \widetilde{\boldsymbol{\mathcal{w}}}_i = \overline{\mathcal{A}}\big[\widetilde{\boldsymbol{\mathcal{w}}}_{i-1} + \mu(\nabla\mathcal{J}(\boldsymbol{\mathcal{w}}_{i-1}) - \nabla\mathcal{J}(\boldsymbol{\mathcal{w}}^\star))\big] \\ \qquad\qquad\qquad\qquad -\mathcal{V}\widetilde{\boldsymbol{\mathcal{y}}}_{i-1} + \mu\overline{\mathcal{A}}\boldsymbol{s}_i(\boldsymbol{\mathcal{w}}_{i-1}), \\ \widetilde{\boldsymbol{\mathcal{y}}}_i = \widetilde{\boldsymbol{\mathcal{y}}}_{i-1} + \mathcal{V}\widetilde{\boldsymbol{\mathcal{w}}}_i. \end{cases} \tag{4.26}$$

Since each $J_k(w)$ is twice-differentiable (see Assumption 3.1), we can appeal to the mean-value theorem from Lemma D.1 in [1], which allows us to express each difference in (4.26) in terms of Hessian matrices for any $k = 1, 2, \ldots, N$:

$$\nabla J_k(\boldsymbol{w}_{k,i-1}) - \nabla J_k(w^\star) = -\boldsymbol{H}_{k,i-1}\widetilde{\boldsymbol{w}}_{k,i-1},$$

where $\boldsymbol{H}_{k,i-1} \triangleq \int_0^1 \nabla^2 J_k\big(w^\star - r\widetilde{\boldsymbol{w}}_{k,i-1}\big)dr \in \mathbb{R}^{M\times M}$. We introduce the block diagonal matrix

$$\boldsymbol{\mathcal{H}}_{i-1} \triangleq \mathrm{diag}\{\boldsymbol{H}_{1,i-1}, \boldsymbol{H}_{2,i-1}, \cdots, \boldsymbol{H}_{K,i-1}\} \tag{4.27}$$

so that

$$\nabla\mathcal{J}(\boldsymbol{\mathcal{w}}_{i-1}) - \nabla\mathcal{J}(w^\star) = -\boldsymbol{\mathcal{H}}_{i-1}\widetilde{\boldsymbol{\mathcal{w}}}_{i-1}. \tag{4.28}$$

Substituting (4.28) into the first recursion in (4.26), we reach

$$\begin{cases} \widetilde{\boldsymbol{\mathcal{w}}}_i = \overline{\mathcal{A}}(I_{KM} - \mu\boldsymbol{\mathcal{H}}_{i-1})\widetilde{\boldsymbol{\mathcal{w}}}_{i-1} - \mathcal{V}\widetilde{\boldsymbol{\mathcal{y}}}_{i-1} + \mu\overline{\mathcal{A}}\boldsymbol{s}_i(\boldsymbol{\mathcal{w}}_{i-1}), \\ \widetilde{\boldsymbol{\mathcal{y}}}_i = \widetilde{\boldsymbol{\mathcal{y}}}_{i-1} + \mathcal{V}\widetilde{\boldsymbol{\mathcal{w}}}_i. \end{cases} \tag{4.29}$$

Next, if we substitute the first recursion in (4.29) into the second one, and recall that $\mathcal{V}^2 = I_{KM} - \overline{\mathcal{A}}$, we reach the following error dynamics.

Lemma 4.2 (Error Dynamics) *Under Assumption 3.1, the error dynamics for the exact diffusion recursions* (4.9)–(4.11) *is as follows*

$$\begin{bmatrix} \widetilde{\boldsymbol{\mathcal{W}}}_i \\ \widetilde{\boldsymbol{\mathcal{Y}}}_i \end{bmatrix} = \underbrace{\begin{bmatrix} \overline{\mathcal{A}} & -\mathcal{V} \\ \mathcal{V}\overline{\mathcal{A}} & \overline{\mathcal{A}} \end{bmatrix}}_{\triangleq \mathcal{B}} \Big(I_{2KM} - \mu \underbrace{\begin{bmatrix} \boldsymbol{\mathcal{H}}_{i-1} & 0 \\ 0 & 0 \end{bmatrix}}_{\triangleq \boldsymbol{\mathcal{T}}_{i-1}} \Big) \begin{bmatrix} \widetilde{\boldsymbol{\mathcal{W}}}_{i-1} \\ \widetilde{\boldsymbol{\mathcal{Y}}}_{i-1} \end{bmatrix} + \mu \underbrace{\begin{bmatrix} \overline{\mathcal{A}} \\ \mathcal{V}\overline{\mathcal{A}} \end{bmatrix}}_{\triangleq \mathcal{B}_\ell} \boldsymbol{s}_i(\boldsymbol{\mathcal{W}}_{i-1}), \tag{4.30}$$

and $\boldsymbol{\mathcal{H}}_i$ *is defined in* (4.27). ■

4.3.2 Transformed Error Dynamics

The direct convergence analysis of recursion (4.30) is challenging. To facilitate the analysis, we identify a convenient change of basis and transform (4.30) into another equivalent form that is easier to handle. To this end, we introduce a fundamental decomposition from [16] here.

Lemma 4.3 (Fundamental Decomposition) *Under Assumptions 3.1 and 4.2, the matrix* $\mathcal{B}$ *defined in* (4.30) *can be decomposed as*

$$\mathcal{B} = \underbrace{\begin{bmatrix} \mathcal{R}_1 & \mathcal{R}_2 & c\mathcal{X}_R \end{bmatrix}}_{\mathcal{X}} \underbrace{\begin{bmatrix} I_M & 0 & 0 \\ 0 & I_M & 0 \\ 0 & 0 & \mathcal{D}_1 \end{bmatrix}}_{\mathcal{D}} \underbrace{\begin{bmatrix} \mathcal{L}_1^{\mathsf{T}} \\ \mathcal{L}_2^{\mathsf{T}} \\ \frac{1}{c}\mathcal{X}_L \end{bmatrix}}_{\mathcal{X}^{-1}} \tag{4.31}$$

where c *can be any positive constant, and* $\mathcal{D} \in \mathbb{R}^{2KM \times 2KM}$ *is a diagonal matrix. Moreover,*

we have

$$\mathcal{R}_1 = \begin{bmatrix} \mathcal{I} \\ 0 \end{bmatrix} \in \mathbb{R}^{2KM\times M}, \quad \mathcal{R}_2 = \begin{bmatrix} 0 \\ \mathcal{I} \end{bmatrix} \in \mathbb{R}^{2KM\times M}, \tag{4.32}$$

$$\mathcal{L}_1 = \begin{bmatrix} \frac{1}{K}\mathcal{I} \\ 0 \end{bmatrix} \in \mathbb{R}^{2KM\times M}, \ \mathcal{L}_2 = \begin{bmatrix} 0 \\ \frac{1}{K}\mathcal{I} \end{bmatrix} \in \mathbb{R}^{2KM\times M}, \tag{4.33}$$

$\mathcal{X}_R \in \mathbb{R}^{2KM\times 2(K-1)M}, \mathcal{X}_L \in \mathbb{R}^{2(K-1)M\times 2KM}$. *where* $\mathcal{I} = \mathbb{1}_K \otimes I_M \in \mathbb{R}^{KM\times M}$. *Also, the matrix* $\mathcal{D}_1$ *is a diagonal matrix with complex entries. The magnitudes of the diagonal entries in* $\mathcal{D}_1$ *are all strictly less than* 1. ■

By multiplying $\mathcal{X}^{-1}$ to both sides of the error dynamics (4.30) and simplifying we arrive at the following result.

Lemma 4.4 (Transformed Error Dynamics) *Under Assumption 3.1 and 4.2, the transformed error dynamics for exact diffusion recursions* (4.9)–(4.11) *is as follows*

$$\begin{bmatrix} \bar{\boldsymbol{z}}_i \\ \check{\boldsymbol{z}}_i \end{bmatrix} = \begin{bmatrix} I_M - \frac{\mu}{K}\sum_{k=1}^K \boldsymbol{H}_{k,i-1} & -\frac{c\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_{i-1}\mathcal{X}_{R,u} \\ -\frac{\mu}{c}\mathcal{D}_1\mathcal{X}_L\boldsymbol{\mathcal{T}}_{i-1}\mathcal{R}_1 & \mathcal{D}_1 - \mu\mathcal{D}_1\mathcal{X}_L\boldsymbol{\mathcal{T}}_{i-1}\mathcal{X}_R \end{bmatrix} \times \begin{bmatrix} \bar{\boldsymbol{z}}_{i-1} \\ \check{\boldsymbol{z}}_{i-1} \end{bmatrix} + \mu \begin{bmatrix} \frac{1}{K}\mathcal{I}^{\mathsf{T}} \\ \frac{1}{c}\mathcal{D}_1\mathcal{X}_L\mathcal{B}_\ell \end{bmatrix} \boldsymbol{s}_i(\boldsymbol{\mathcal{W}}_{i-1}). \tag{4.34}$$

where $\mathcal{X}_{R,u} \in \mathbb{R}^{KM\times 2(K-1)M}$ *is the upper part of matrix* $\mathcal{X}_R = [\mathcal{X}_{R,u}; \mathcal{X}_{R,d}]$. *The relation between the original and transformed error vectors are*

$$\begin{bmatrix} \widetilde{\boldsymbol{\mathcal{W}}}_i \\ \widetilde{\boldsymbol{\mathcal{Y}}}_i \end{bmatrix} = \begin{bmatrix} \mathcal{R}_1 & c\mathcal{X}_R \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{z}}_i \\ \check{\boldsymbol{z}}_i \end{bmatrix}. \tag{4.35}$$

■

4.4 Mean-square Convergence

Using the transformed error dynamics derived in (4.34), we can now analyze the mean-square convergence of exact diffusion (4.9)–(4.11) in the stochastic and adaptive setting. To begin

with, we introduce the filtration

$$\boldsymbol{\mathcal{F}}_{i-1} = \text{filtration}\{\boldsymbol{w}_{k,-1}, \boldsymbol{w}_{k,0}, \cdots, \boldsymbol{w}_{k,i-1}, \text{ all } k\}. \tag{4.36}$$

The following assumption is standard on the gradient noise process (see [1,2]) and is satisfied in many situations of interest such as linear and logistic regression problems.

Assumption 4.3 (Conditions on gradient noise) *It is assumed that the first and second-order conditional moments of the individual gradient noises for any k and i satisfy*

$$\mathbb{E}[\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})|\boldsymbol{\mathcal{F}}_{i-1}] = 0, \tag{4.37}$$

$$\mathbb{E}[\|\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\|^2|\boldsymbol{\mathcal{F}}_{i-1}] \leq \beta_k^2\|\widetilde{\boldsymbol{w}}_{k,i-1}\|^2+\sigma_k^2 \tag{4.38}$$

for some constants β_k and σ_k. Moreover, we assume the $\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})$ are independent of each other for any k, i given $\boldsymbol{\mathcal{F}}_{i-1}$. ■

With Assumption 4.3, it can be verified that

$$\mathbb{E}[\boldsymbol{s}_i(\boldsymbol{\mathcal{W}}_{i-1})|\boldsymbol{\mathcal{F}}_{i-1}] = 0, \quad \forall\ i, \tag{4.39}$$

$$\mathbb{E}\left[\left\|\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\right\|^2\Big|\boldsymbol{\mathcal{F}}_{i-1}\right] \leq \frac{\beta^2}{K}\|\widetilde{\boldsymbol{\mathcal{W}}}_{i-1}\|^2+\frac{\sigma^2}{K} \tag{4.40}$$

where $\beta^2 \triangleq \max_k\{\beta_k^2\}/K$ and $\sigma^2 \triangleq \sum_{k=1}^K \sigma_k^2/K$.

Theorem 4.1 (Mean-Square Convergence) *Under Assumptions 3.1–4.3, if the step-size μ satisfies*

$$\mu \leq \frac{(1-\lambda)\nu}{(32+16c_1c_2+8\sqrt{c_1c_2})(\delta^2+\beta_{max}^2)} = O\left(\frac{(1-\lambda)\nu}{\delta^2+\beta_{max}^2}\right) \tag{4.41}$$

where $\lambda = \max\{|\lambda_2(A)|, |\lambda_K(A)|\}$, $\beta_{max}^2 = \max_k\{\beta_k^2\}$, and c_1, c_2 are constants defined in (4.96)*, then the $\boldsymbol{w}_{k,i}$ generated by exact diffusion recursion* (4.14) *converges exponentially fast to a neighborhood around $w^\star$. The convergence rate is $\rho = 1 - O(\mu\nu)$, and the size of the neighborhood can be characterized as follows:*

$$\limsup_{i\to\infty}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 = O\left(\frac{\mu\sigma^2}{K\nu}+\frac{\delta^2}{\nu^2}\cdot\frac{\mu^2\sigma^2}{1-\lambda}\right) \tag{4.42}$$

Proof. See Appendix 4.A. ■

Theorem 4.1 indicates that when μ is smaller than a specified upper bound, the exact diffusion over adaptive networks is stable. The theorem also provides a bound on the size of the steady-state mean-square error. To compare exact diffusion with diffusion, we examine the mean-square convergence property of diffusion as well.

Lemma 4.5 (Mean-square stability of Diffusion) *Under Assumptions 3.1–4.3, if μ satisfies*

$$\mu \leq \frac{(1-\lambda)\nu}{(12+4e_1e_2+\sqrt{6e_1e_2})(\delta^2+\beta_{max}^2)} = O\left(\frac{(1-\lambda)\nu}{\delta^2+\beta_{max}^2}\right) \tag{4.43}$$

where $\lambda = \max\{|\lambda_2(A)|, |\lambda_K(A)|\}$, $\beta_{max}^2 = \max_k\{\beta_k^2\}$, e_1 and e_2 are constants that are independent of λ, δ, ν and β, then $\boldsymbol{w}_{k,i}$ generated by the diffusion recursions (4.2)–(4.3) converge exponentially fast to a neighborhood around $w^\star$. The convergence rate is $1-O(\mu\nu)$, and the size of the neighborhood can be characterized as follows

$$\begin{aligned}&\limsup_{i\to\infty} \frac{1}{K}\sum_{k=1}^{K} \|\widetilde{\boldsymbol{w}}_{k,i}\|^2 \\ &\quad = O\left(\frac{\mu\sigma^2}{K\nu} + \frac{\delta^2}{\nu^2}\cdot\frac{\mu^2\lambda^2\sigma^2}{1-\lambda} + \frac{\delta^2}{\nu^2}\cdot\frac{\mu^2\lambda^2 b^2}{(1-\lambda)^2}\right),\end{aligned} \tag{4.44}$$

where $b^2 = (1/K)\sum_{k=1}^{K}\|\nabla J_k(w^\star)\|^2$ is a bias term.

Proof. See Appendix 4.B for proof detail. ■

Comparing (4.42) and (4.44), it is observed that the expressions for both algorithms consist of two major terms – one $O(\mu)$ term and one $O(\mu^2)$ term. However, diffusion suffers from an additional bias term $O(\mu^2\lambda^2 b^2/(1-\lambda)^2)$.

Remark 4.1 (Deterministic case) *When $\sigma^2 = 0$, both diffusion and exact diffusion reduce to the deterministic scenario in which the real gradient $\nabla J_k(w)$ is available. In this scenario, it is observed from (4.42) and (4.44) that the error $\widetilde{\boldsymbol{w}}_{k,i}$ in exact diffusion converges to 0 while that in diffusion converges to $O(\mu^2 b^2)$, which is consistent with the results presented in [9, 14–16].* ■

Remark 4.2 (Zero bias) *When $b^2 = 0$, it holds that each local minimizer $w_k^\star$ coincides with the global minimizer $w^\star$, i.e., $w_k^\star = w^\star$ for any k. In this scenario, it is observed from* (4.44) *that diffusion has the steady-state error bound*

$$\limsup_{i\to\infty} \frac{1}{K}\sum_{k=1}^{K} \|\widetilde{\boldsymbol{w}}_{k,i}\|_{\mathrm{d}}^2 = O\left(\frac{\mu\sigma^2}{K\nu} + \frac{\delta^2}{\nu^2}\cdot\frac{\mu^2\lambda^2\sigma^2}{1-\lambda}\right) \tag{4.45}$$

which is smaller than the error bound (4.42) *for exact diffusion especially when λ approaches* 0. *This result is consistent with [146], which finds diffusion outperforms primal-dual distributed adaptive methods when $w_k^\star = w^\star$ in terms of steady-state performance.* ■

Remark 4.3 (Large bias) *When b^2 is sufficiently large so that the bias term (i.e., the third term) in* (4.44) *dominates the entire error bound, it is observed from* (4.42) *and* (4.44) *that exact diffusion performs better than diffusion since it removes the bias term completely. This result is consistent with [145], which claims exact diffusion is endowed with faster convergence rate when the data variance across the network is large.* ■

In the following subsections, we will focus on the scenario where $\sigma^2 > 0$ and the bias b^2 is a small positive constant. In this scenario, we will study how the step-size μ and topology λ influence the diffusion and exact diffusion algorithms.

4.4.1 Well-connected Network

When the network is well-connected, it holds that λ approaches 0. For example, the fully-connected network has $\lambda = 0$. In this scenario, the $O(\mu^2)$ terms inside diffusion's error bound will vanish and (4.44) becomes

$$\limsup_{i\to\infty} \frac{1}{K}\sum_{k=1}^{K} \|\widetilde{\boldsymbol{w}}_{k,i}\|_{\mathrm{d}}^2 = O\left(\frac{\mu\sigma^2}{K\nu}\right). \tag{4.46}$$

In comparison, the error bound (4.42) for exact diffusion is

$$\limsup_{i\to\infty} \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|_{ed}^2 = O\left(\frac{\mu\sigma^2}{K\nu} + \frac{\mu^2\delta^2\sigma^2}{\nu^2}\right) \tag{4.47}$$

as $\lambda \to 0$. When μ is moderately small such that the term $O(\mu^2\delta^2\sigma^2/\nu^2)$ is non-negligible, we conclude that diffusion works better than exact diffusion. To roughly characterize the "moderately" small step-size, we assume $O(\mu^2\delta^2\sigma^2/\nu^2)$ is non-negligible if $\mu^2\delta^2\sigma^2/\nu^2 \geq \mu\sigma^2/(K\nu)$, from which we get $\mu \geq \nu/(K\delta^2)$. Combining it with (4.41) we conclude that if μ satisfies (note that $\lambda \to 0$)

$$\frac{\nu}{K\delta^2} \leq \mu \leq \frac{d_1\nu}{\delta^2 + \beta_{max}^2} \tag{4.48}$$

where $d_1 = 1/(32 + 16c_1c_2 + 8\sqrt{c_1c_2})$, it holds that $O(\mu^2\delta^2\sigma^2/\nu^2)$ is non-trivial and diffusion has better steady-state performance than exact diffusion. To make the interval in (4.48) valid, it is enough to let K be sufficiently large.

However, if the step-size μ is chosen sufficiently small, then the second term in (4.47) is also negligible and hence both diffusion and exact diffusion will perform similarly. An example for "sufficiently" small step-size is when $\mu = \nu/(K^2\delta^2)$. By substituting $\mu = \nu/(K^2\delta^2)$ into (4.47), we reach $\limsup_{i\to\infty} \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|_{ed}^2 = O(\frac{\sigma^2}{K^3\delta^2} + \frac{\sigma^2}{K^4\delta^2}) = O(\frac{\sigma^2}{K^3\delta^2}) = O(\frac{\mu\sigma^2}{K\nu})$ in which the $O(\mu^2)$ term is negligible.

4.4.2 Sparsely-connected Network

When the network is sparsely-connected, it holds that λ approaches 1. In this scenario, even a trivial bias constant b^2 can be significantly amplified by the coefficient $1/(1-\lambda)^2$. When λ approaches 1, the first two terms in (4.44) will be the same as those in (4.42). As a result, when μ is moderately small and λ is close to 1 such that the bias term $O(\mu^2\delta^2\lambda^2b^2/(1-\lambda)^2\nu^2)$ is non-negligible, we conclude that exact diffusion works better than diffusion. Furthermore, the advantage of exact diffusion will be more evident if the bias gets more significant as $\lambda \to 1$. In the following example, we list several network topologies in which the bias $O(\mu^2b^2/(1-\lambda)^2)$ dominates (4.5) easily.

Example (Linear, Cyclic, and Grid networks). A linear or cyclic network with K agents is a network where each agent connects with its previous and next neighbors. On the other hand, a grid network with K agents is a network in which each node connects with its neighbors from left, right, top, and bottom. The grid and cycle networks are illustrated in

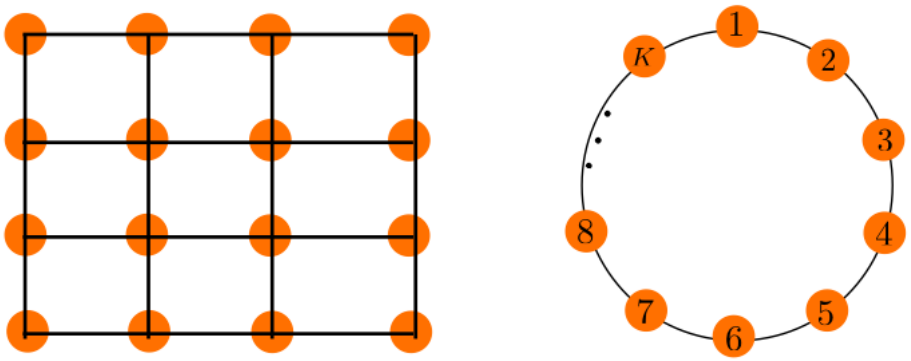

Figure 4.2: Illustration of the grid topology and cyclic topology.

Fig. 4.2. For these networks, it is shown in [110, 147] that

$$1-\lambda = O(1/K^2) \quad \text{(linear or cyclic network)} \tag{4.49}$$

$$1-\lambda = O(1/K) \quad \text{(grid network)} \tag{4.50}$$

and therefore, the bias term $O(\mu^2 b^2/(1-\lambda)^2)$ in diffusion over linear (or cyclic) graph and grid graph becomes $O(\mu^2 b^2 K^4)$ and $O(\mu^2 b^2 K^2)$ respectively, which increases rapidly with the size of the network. As a result, exact diffusion, by correcting the bias term, is evidently superior to diffusion over these network topologies. ■

To roughly characterize the "moderately" small step-size, we assume $O(\mu^2\delta^2\lambda^2 b^2/(1-\lambda)^2\nu^2)$ is non-trivial if

$$\frac{\delta^2}{\nu^2}\cdot\frac{\mu^2 b^2}{(1-\lambda)^2} \geq \frac{\mu\sigma^2}{K\nu} \tag{4.51}$$

from which we get $\mu \geq (1-\lambda)^2\sigma^2\nu/K\delta^2 b^2$. Combining it with (4.43), we conclude that if μ satisfies

$$\frac{(1-\lambda)^2\sigma^2\nu}{K\delta^2 b^2} \leq \mu \leq \frac{d_2(1-\lambda)\nu}{\delta^2+\beta_{max}^2}, \tag{4.52}$$

where $d_2 = 12 + 4e_1e_2 + \sqrt{6e_1e_2}$ is a constant, then the bias term in (4.44) is significant and exact diffusion is expected to have better performance than diffusion in steady-state. To make the interval in (4.52) valid, it is enough to let λ be sufficiently close to 1 and K be sufficiently large such that

$$\frac{(1-\lambda)^2\sigma^2\nu}{K\delta^2 b^2} < \frac{d_2(1-\lambda)\nu}{\delta^2+\beta^2} \iff \frac{b^2}{1-\lambda} > \frac{(\delta^2+\beta^2)}{d_2 K\delta^2}\sigma^2. \tag{4.53}$$

On the other hand, if μ is adjusted to be sufficiently small, the $O(\mu)$ term in both expressions (4.42) and (4.44) will eventually dominate for any fixed b^2 and λ. In such scenario, it holds that

$$\limsup_{i\to\infty} \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|_{\mathrm{ed}}^2 = O\Big(\frac{\mu\sigma^2}{K\nu}\Big), \tag{4.54}$$

$$\limsup_{i\to\infty} \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|_{\mathrm{d}}^2 = O\Big(\frac{\mu\sigma^2}{K\nu}\Big). \tag{4.55}$$

It is observed that both diffusion and exact diffusion will have the same mean-square error order, which implies that diffusion and exact diffusion will perform similarly in this scenario. Such "sufficiently" small step-size can be roughly characterized by the range

$$\mu \leq d_3(1-\lambda)^{2+x} \quad \text{where} \quad x > 0. \tag{4.56}$$

for some $d_3 > 0$. The comparison between exact diffusion and diffusion is listed in Fig. 4.1.

4.5 Mean-square Deviation Expression

In the last section, we showed that when μ is sufficiently small, the steady-state mean-square deviation of both diffusion and exact diffusion will be dominated by a term on the order of $O(\mu\sigma^2/\nu)$, as illustrated by (4.54)–(4.55). However, the hidden constants inside the big-O notation are still unclear. In this section, we show that, when μ is approaching 0, i.e., $\mu \to 0$, diffusion and exact diffusion will have exactly the same MSD expression in steady state. To this end, we recall the definition of mean-square deviation (MSD) from [1] as follows:

$$\mathrm{MSD} = \mu\Big(\lim_{\mu\to 0}\limsup_{i\to\infty} \frac{1}{\mu K}\sum_{k=1}^{K} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^2\Big). \tag{4.57}$$

Note that the MSD defined above is precise to the first-order in the step-size. All higher order terms are ignored.

4.5.1 Approximate Error Dynamics

It is generally difficult to derive the MSD performance of exact diffusion with the original transformed error dynamics developed in Lemma 4.4. We therefore propose an approximate

error dynamics and employ it to assess the MSD performance. To this end, we define

$$H_k=\nabla^2 J_k(w^\star),\ \mathcal{H}=\text{diag}\{H_1,\cdots,H_K\},\ \mathcal{T}=\begin{bmatrix}\mathcal{H} & 0\\ 0 & 0\end{bmatrix}. \tag{4.58}$$

Obviously, it holds that $\boldsymbol{H}_{k,i}\to H$, $\boldsymbol{\mathcal{H}}_i\to\mathcal{H}$ and $\boldsymbol{\mathcal{T}}_i\to\mathcal{T}$ if $\boldsymbol{\mathcal{W}}_i\to\mathcal{W}^\star$. Next, we consider the approximate error dynamic as follows.

$$\begin{aligned}\begin{bmatrix}\bar{\boldsymbol{z}}'_i\\ \check{\boldsymbol{z}}'_i\end{bmatrix}=&\begin{bmatrix}I_M-\frac{\mu}{K}\sum_{k=1}^K H_k & -\frac{c\mu}{K}\mathcal{I}^\mathsf{T}\mathcal{H}\mathcal{X}_{R,u}\\ -\frac{\mu}{c}\mathcal{D}_1\mathcal{X}_L\mathcal{T}\mathcal{R}_1 & \mathcal{D}_1-\mu\mathcal{D}_1\mathcal{X}_L\mathcal{T}\mathcal{X}_R\end{bmatrix}\begin{bmatrix}\bar{\boldsymbol{z}}'_{i-1}\\ \check{\boldsymbol{z}}'_{i-1}\end{bmatrix}\\ &+\mu\begin{bmatrix}\frac{1}{K}\mathcal{I}^\mathsf{T}\\ \frac{1}{c}\mathcal{D}_1\mathcal{X}_L\mathcal{B}_\ell\end{bmatrix}\boldsymbol{s}_i(\boldsymbol{\mathcal{W}}_{i-1}).\end{aligned} \tag{4.59}$$

Note that we replace $\boldsymbol{H}_{k,i-1}$, $\boldsymbol{\mathcal{H}}_{i-1}$ and $\boldsymbol{\mathcal{T}}_{i-1}$ in (4.34) with H_k, $\mathcal{H}$ and $\mathcal{T}$ in (4.59). We can show that the iterates $\bar{\boldsymbol{z}}'_i$ and $\check{\boldsymbol{z}}'_i$ generated through the approximate error dynamic (4.59) are close enough to $\bar{\boldsymbol{z}}_i$ and $\check{\boldsymbol{z}}_i$ generated from the original recursion (4.34) – see Lemma 4.6 below. This implies that we can employ recursion (4.59) rather than (4.34) to establish the MSD performance. To this end, we first introduce a few more assumptions on cost functions and the gradient noise. These assumptions are adapted from [1].

Assumption 4.4 (Smoothness condition in the limit) *For each cost function $J_k(w)$, it is assumed that*

$$\|\nabla^2 J_k(w^\star+\Delta w)-\nabla^2 J_k(w^\star)\|\le\kappa\|\Delta w\| \tag{4.60}$$

for small perturbations $\|\Delta w\|\le\epsilon$, where $\kappa>0$ is a constant.

Assumption 4.5 (Forth-Order Moment) *It is assumed for each k and i that*

$$\mathbb{E}[\|\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\|^4|\boldsymbol{\mathcal{F}}_{i-1}]\le\beta_{4,k}^4\|\widetilde{\boldsymbol{w}}_{k,i-1}\|^4+\sigma_{4,k}^4. \tag{4.61}$$

where $\beta_{4,k}$ and $\sigma_{4,k}$ are some positive constants. ■

By following the proof of Theorem 10.2 from [1], we can prove in the following lemma that difference between the original iterates (4.34) and the transformed iterates (4.59) is small.

Lemma 4.6 (Approximation Error) *Under Assumptions 3.1–4.5, it holds for sufficiently small step-sizes that*

$$\limsup_{i\to\infty} \mathbb{E} \left\| \begin{bmatrix} \bar{\boldsymbol{z}}_i \\ \check{\boldsymbol{z}}_i \end{bmatrix} - \begin{bmatrix} \bar{\boldsymbol{z}}'_i \\ \check{\boldsymbol{z}}'_i \end{bmatrix} \right\|^2 = O(\mu^2) \tag{4.62}$$

■

4.5.2 Deriving the MSD expression

Recall from (4.35) that

$$\widetilde{\boldsymbol{\mathcal{W}}}_i = \begin{bmatrix} \mathcal{I} & c\mathcal{X}_{R,u} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{z}}_i \\ \check{\boldsymbol{z}}_i \end{bmatrix}. \tag{4.63}$$

This together with $\mathcal{I}^{\mathsf{T}}\mathcal{X}_{R,u} = 0$[2] implies that

$$\|\widetilde{\boldsymbol{\mathcal{W}}}_i\|^2 = \begin{bmatrix} \bar{\boldsymbol{z}}_i \\ \check{\boldsymbol{z}}_i \end{bmatrix}^{\mathsf{T}} \underbrace{\begin{bmatrix} KI_{KM} & 0 \\ 0 & c^2\mathcal{X}_{R,u}^{\mathsf{T}}\mathcal{X}_{R,u} \end{bmatrix}}_{\triangleq\,\Gamma} \begin{bmatrix} \bar{\boldsymbol{z}}_i \\ \check{\boldsymbol{z}}_i \end{bmatrix} \tag{4.64}$$

For simplicity, in the following we let

$$\boldsymbol{z}_i = \begin{bmatrix} \bar{\boldsymbol{z}}_i \\ \check{\boldsymbol{z}}_i \end{bmatrix}, \quad \boldsymbol{z}'_i = \begin{bmatrix} \bar{\boldsymbol{z}}'_i \\ \check{\boldsymbol{z}}'_i \end{bmatrix}. \tag{4.65}$$

and it holds that $\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_i\|^2 = \mathbb{E}\|\boldsymbol{z}_i\|_\Gamma^2$. The following lemma shows that $\mathbb{E}\|\boldsymbol{z}_i\|_\Gamma^2$ is close to $\mathbb{E}\|\boldsymbol{z}'_i\|_\Gamma^2$.

Lemma 4.7 (Approximation Scaled Error) *Under Assumptions 3.1–4.5, it holds for sufficiently small step-sizes that*

$$\limsup_{i\to\infty}\ \mathbb{E}\|\boldsymbol{z}_i\|_\Gamma^2 - \mathbb{E}\|\boldsymbol{z}'_i\|_\Gamma^2 = O(\mu^{3/2}) \tag{4.66}$$

■

[2]Since $\mathcal{X}^{-1}\mathcal{X} = I$ with $\mathcal{X}$ and $\mathcal{X}^{-1}$ defined in (4.31), we have $c\mathcal{L}_1^{\mathsf{T}}\mathcal{X}_R = \frac{c}{K}\mathcal{I}^{\mathsf{T}}\mathcal{X}_{R,u} = 0$.

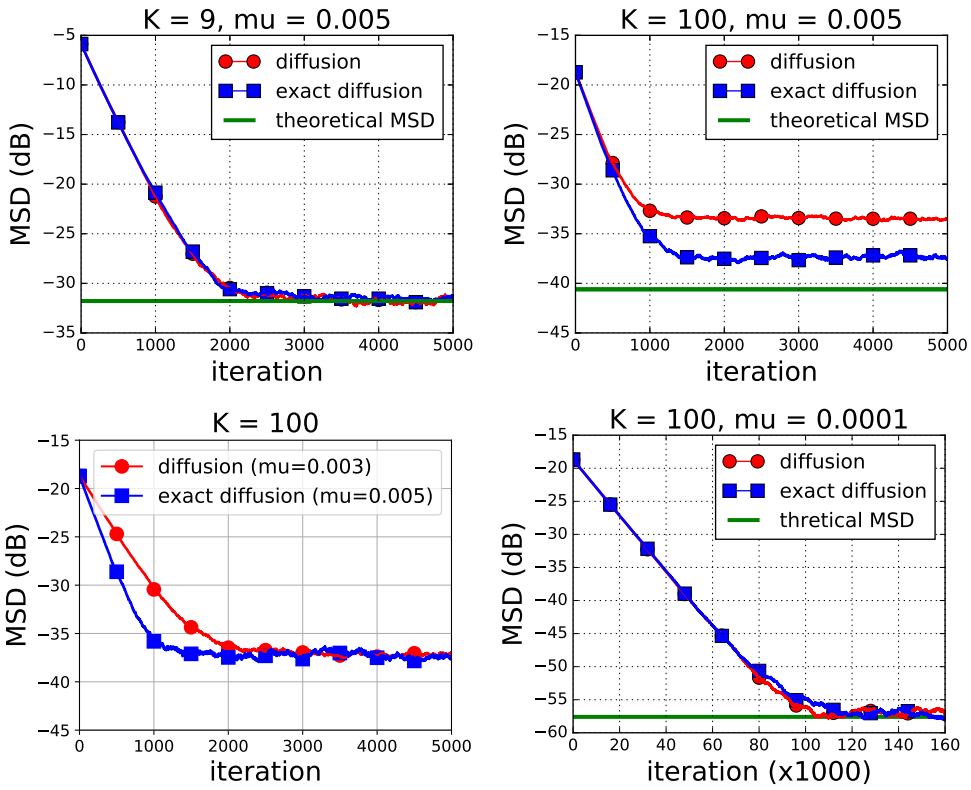

Figure 4.3: Diffusion v.s. exact diffusion over grid networks for problem (4.73).

Proof. It holds that

$$\begin{aligned}\mathbb{E}\|\boldsymbol{z}_i'\|_\Gamma^2 &= \mathbb{E}\|\boldsymbol{z}_i' - \boldsymbol{z}_i + \boldsymbol{z}_i\|_\Gamma^2 \\ &\leq \mathbb{E}\|\boldsymbol{z}_i' - \boldsymbol{z}_i\|_\Gamma^2 + \mathbb{E}\|\boldsymbol{z}_i\|_\Gamma^2 + 2\mathbb{E}[(\boldsymbol{z}_i' - \boldsymbol{z}_i)^{\mathsf{T}}\Gamma\boldsymbol{z}_i] \\ &\leq \mathbb{E}\|\boldsymbol{z}_i' - \boldsymbol{z}_i\|_\Gamma^2 + \mathbb{E}\|\boldsymbol{z}_i\|_\Gamma^2 + 2\sqrt{\mathbb{E}\|\boldsymbol{z}_i' - \boldsymbol{z}_i\|_\Gamma^2 \mathbb{E}\|\boldsymbol{z}_i\|_\Gamma^2},\end{aligned}$$

which implies that

$$\begin{aligned}&\mathbb{E}\|\boldsymbol{z}_i'\|_\Gamma^2 - \mathbb{E}\|\boldsymbol{z}_i\|_\Gamma^2 \\ \leq\ & \mathbb{E}\|\boldsymbol{z}_i' - \boldsymbol{z}_i\|_\Gamma^2 + 2\sqrt{\mathbb{E}\|\boldsymbol{z}_i' - \boldsymbol{z}_i\|_\Gamma^2 \mathbb{E}\|\boldsymbol{z}_i\|_\Gamma^2} \\ \leq\ & \lambda_{\max}(\Gamma)\mathbb{E}\|\boldsymbol{z}_i' - \boldsymbol{z}_i\|^2 + 2\lambda_{\max}(\Gamma)\sqrt{\mathbb{E}\|\boldsymbol{z}_i' - \boldsymbol{z}_i\|^2 \mathbb{E}\|\boldsymbol{z}_i\|^2}\end{aligned}$$

where $\lambda_{\max}(\Gamma)$ is the largest eigenvalue of Γ. From (4.65) we know it holds for sufficiently

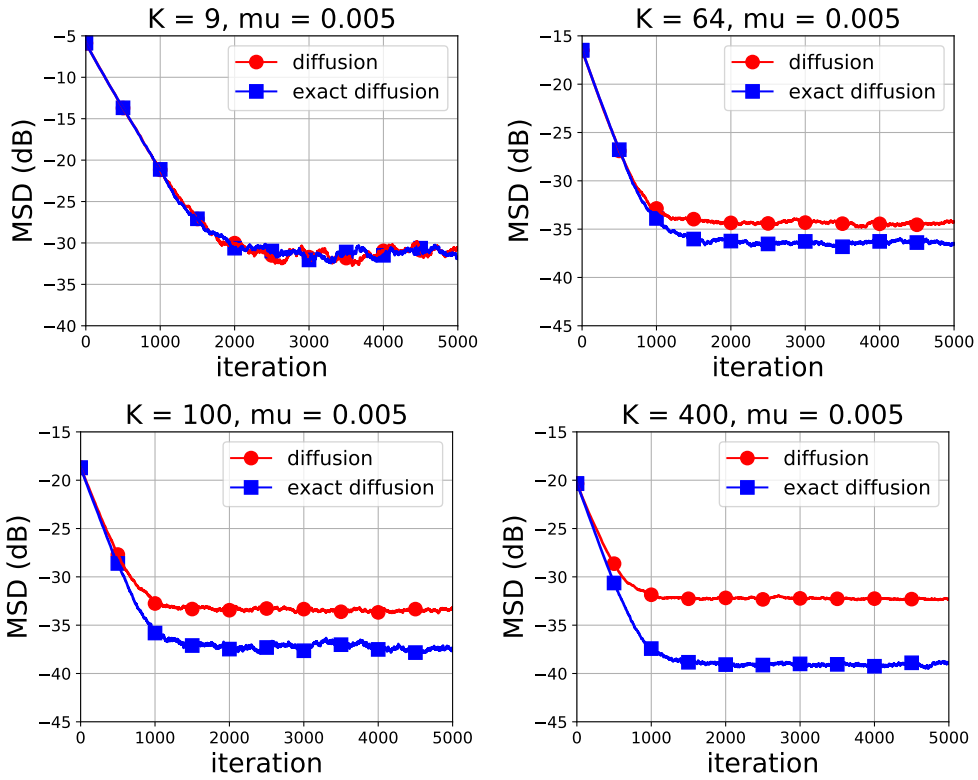

Figure 4.4: The superiority of exact diffusion is more evident as the grid network becomes larger when solving problem (4.73).

small μ that

$$\begin{aligned}\limsup_{i\to\infty}\mathbb{E}\|\boldsymbol{z}_i\|^2 &= \limsup_{i\to\infty}\mathbb{E}\|\bar{\boldsymbol{z}}_i\|^2 + \limsup_{i\to\infty}\mathbb{E}\|\check{\boldsymbol{z}}_i\|^2 \\ &\overset{(4.111)}{=} O(\mu)+O(\mu^2)=O(\mu). \end{aligned} \tag{4.67}$$

Also, from (4.62) we have $\limsup_{i\to\infty}\mathbb{E}\|\boldsymbol{z}_i'-\boldsymbol{z}_i\|^2=O(\mu^2)$. Since Γ is independent of μ, it therefore holds that

$$\limsup_{i\to\infty}\left(\mathbb{E}\|\boldsymbol{z}_i'\|_\Gamma^2-\mathbb{E}\|\boldsymbol{z}_i\|_\Gamma^2\right)=O(\mu^{3/2}). \tag{4.68}$$

■

Now we establish the MSD expression for exact diffusion. Since $\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_i\|^2=\mathbb{E}\|\boldsymbol{z}_i\|_\Gamma^2$ is close to $\mathbb{E}\|\boldsymbol{z}_i'\|_\Gamma^2$ as proved in Lemma 4.7, we will first derive the MSD expression for $\mathbb{E}\|\boldsymbol{z}_i'\|_\Gamma^2$ and use it to facilitate the derivation of the MSD for exact diffusion, i.e., $\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_i\|^2$. To proceed, we assume that, in the limit, the following covariance matrix evaluated at the global solution

$w^\star$ exists

$$S_k \triangleq \lim_{i\to\infty} \mathbb{E}[\boldsymbol{s}_{k,i}(w^\star)\boldsymbol{s}_{k,i}(w^\star)^{\mathsf{T}}]. \tag{4.69}$$

The following theorem establishes the MSD expression of the approximate error dynamics.

Theorem 4.2 (MSD expression) *Under Assumptions 3.1–4.5, it holds for exact diffusion that*

$$\mathrm{MSD}_{\mathrm{ed}} = \frac{\mu}{2K} Tr\left\{ \left(\sum_{k=1}^{K} H_k\right)^{-1} \left(\sum_{k=1}^{K} S_k\right) \right\}. \tag{4.70}$$

Proof. See Appendix 4.C. ■

Recall the MSD expression for standard diffusion is [1, Equation (11.140)]:

$$\mathrm{MSD}_{\mathrm{d}} = \frac{\mu}{2K} \mathrm{Tr}\left\{ \left(\sum_{k=1}^{K} H_k\right)^{-1} \left(\sum_{k=1}^{K} S_k\right) \right\}. \tag{4.71}$$

It is observed that the MSD expression for diffusion (4.71) is equal to that of exact diffusion (4.70). This implies that diffusion and exact diffusion will perform exactly the same in steady state for sufficiently small step-sizes.

4.6 Numerical Simulation

4.6.1 Mean-square-error Network

In this subsection we consider the scenario in which K agents observe streaming data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ that satisfy the regression model

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}^{\mathsf{T}} w_k^\star + \boldsymbol{v}_k(i) \tag{4.72}$$

where $w_k^\star$ is the local optimal solution at agent k, and the noise process, $\boldsymbol{v}_k(i)$, is independent of the regression data, $\boldsymbol{u}_{k,i}$. The cost over the mean-square-error (MSE) network is defined by

$$\min_{w\in\mathbb{R}^M} \quad \sum_{k=1}^{K} \mathbb{E}\big(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}^{\mathsf{T}} w\big)^2. \tag{4.73}$$

To generate $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$, we first generate the local optimal solution following a standard Gaussian distribution, i.e., $w_k^\star \sim \mathcal{N}(0, I_M)$. Next we generate $\boldsymbol{u}_{k,i} \sim \mathcal{N}(0, \Lambda_k)$ where Λ_k is a positive diagonal matrix and $\boldsymbol{v}_k(i) \sim \mathcal{N}(0, 0.1I_M)$. With $w_k^\star$, $\boldsymbol{u}_{k,i}$ and $\boldsymbol{v}_k(i)$, we generate $\boldsymbol{d}_k(i)$ according to (4.72). Also, we can verify that the global solution to (4.73) is given by

$$w^\star = \left(\sum_{k=1}^K \Lambda_k\right)^{-1} \sum_{k=1}^K \Lambda_k w_k^\star. \tag{4.74}$$

In all figures below, the y-axis indicates the MSD performance $\sum_{k=1}^K \mathbb{E}\|\boldsymbol{w}_{k,i} - w^\star\|^2/K$.

We first compare the performance of exact diffusion and diffusion over a grid topology — see the first plot in Fig.4.3. We first let $K = 9$ and $\mu = 0.005$ and compare exact diffusion and diffusion. With these two parameters, it is shown in the first plot in Fig.4.3 that both methods perform almost the same, and the steady-state MSD performance of both methods coincide with the derived MSD expression (4.70). In the second plot in Fig.4.3, we maintain $\mu = 0.005$ but increase the network size to 100 nodes. As we explained in Sec.4.4.2, a grid topology with larger network size has λ closer to 1, which amplifies the inherent bias $O(\mu^2 b^2/(1-\lambda)^2)$ suffered by diffusion. It is observed that exact diffusion has a clear advantage over diffusion during the steady-state stage. Note that in the second plot both diffusion and exact diffusion do not coincide with the derived theoretical MSD expression. This is because the theoretical MSD expression in (4.70) is only precise to first-order in μ. When λ approaches 1 as the grid network gets larger, the second-order term of μ is amplified by $1/(1-\lambda)$ and becomes non-negligible. In the third plot, we maintain $K = 100$ and $\mu_{ed} = 0.005$ for exact diffusion while decreasing the step-size of diffusion to ($\mu_{\rm d} = 0.003$) so that it has the same steady-state MSD performance as diffusion. It is observed that in this scenario exact diffusion converges faster than diffusion to reach the same steady-state performance, which implies that exact diffusion has faster adaptive and tracking abilities than diffusion over large grid graphs. In the fourth plot of Fig.4.3, we adjust $\mu = 0.0001$ for both methods while keeping $K = 100$. Since μ gets much smaller, the inherent bias in diffusion (4.44) becomes trivial and both methods perform similarly again, and they coincide with the derived MSD expression.

To further show how superior the exact diffusion can be compared to diffusion over the

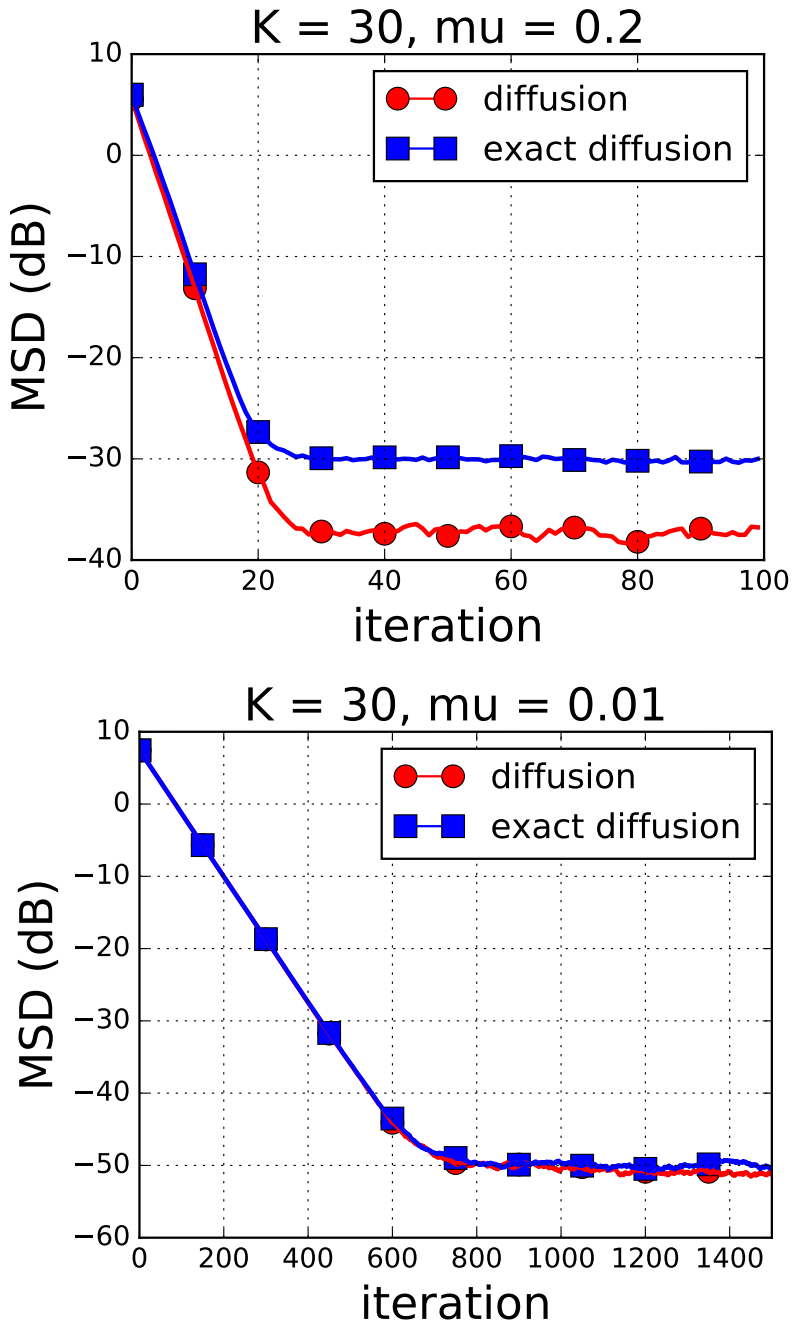

Figure 4.5: Diffusion v.s. exact diffusion over a fully connected network for problem (4.73).

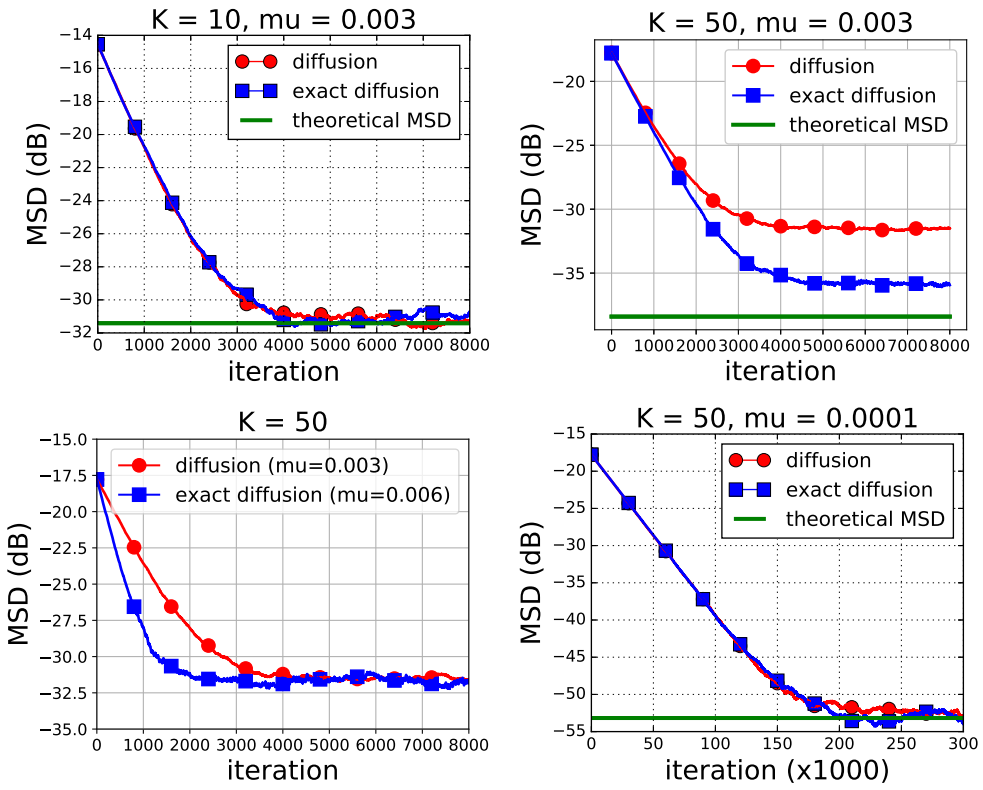

Figure 4.6: Diffusion v.s. exact diffusion over cyclic networks for problem 4.75.

grid network, we depict the performance of diffusion and exact diffusion for different network sizes in Fig.4.4. It is observed that the superiority of exact diffusion becomes more evident as the grid network gets larger, and exact diffusion performs much better than diffusion when $K = 400$.

In the third experiment, we compare diffusion with exact diffusion over a fully connected network with $K = 30$. Since $\lambda = 0$ for this scenario, it is expected diffusion has better steady-state performance than exact diffusion when μ is moderately small, see the discussion in Sec. 4.4.1. Also, the superiority of diffusion should vanish as the step-size becomes sufficiently small. The comparison results shown in Fig.4.5 are consistent with our discussion in 4.4.1.

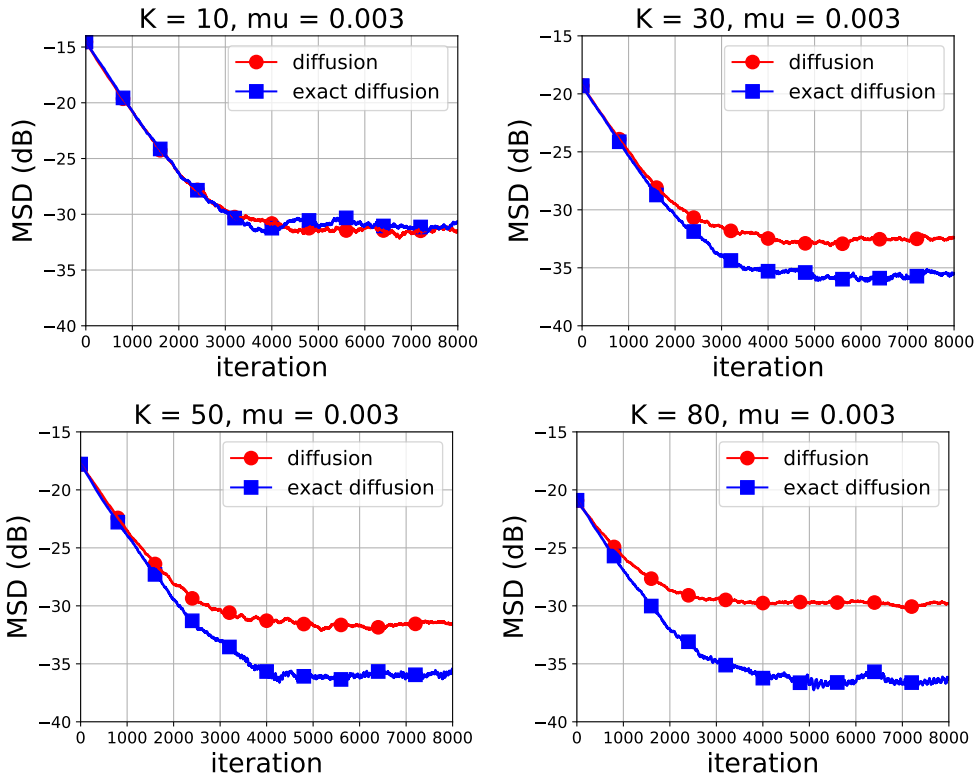

Figure 4.7: The superiority of exact diffusion gets more evident as the cyclic networks gets larger when solving problem 4.75.

4.6.2 Distributed Logistic Regression

In this subsection we compare the performance of exact diffusion and diffusion when solving a decentralized logistic regression problem of the form:

$$\min_{w\in\mathbb{R}^M} \sum_{k=1}^{K} \mathbb{E}\left\{\ln\left(1+e^{-\boldsymbol{\gamma}_k \boldsymbol{h}_k^{\mathsf{T}} w}\right)\right\} + \frac{\rho}{2}\|w\|^2, \tag{4.75}$$

where $(\boldsymbol{h}_k, \boldsymbol{\gamma}_k)$ represent the streaming data received by agent k. Variable $\boldsymbol{h}_k \in \mathbb{R}^M$ is the feature vector and $\boldsymbol{\gamma}_k \in \{-1, +1\}$ is the label scalar. In all experiments, we set $M = 20$ and $\rho = 0.001$. To make the $J_k(w)$'s have different minimizers, we first generate K different local minimizers $\{w_k^\star\}$. All $w_k^\star$ are normalized so that $\|w_k^\star\|^2 = 1$. At agent k, we generate each feature vector $\boldsymbol{h}_{k,i} \sim \mathcal{N}(0, I_{20})$. To generate the corresponding label $\boldsymbol{\gamma}_k(i)$, we generate a random variable $z_{k,i} \in \mathcal{U}(0,1)$. If $z_{k,i} \leq 1/(1+\exp(-\boldsymbol{h}_{k,i}^{\mathsf{T}} w_k^\star))$, we set $\boldsymbol{\gamma}_k(i) = 1$; otherwise $\boldsymbol{\gamma}_k(i) = -1$.

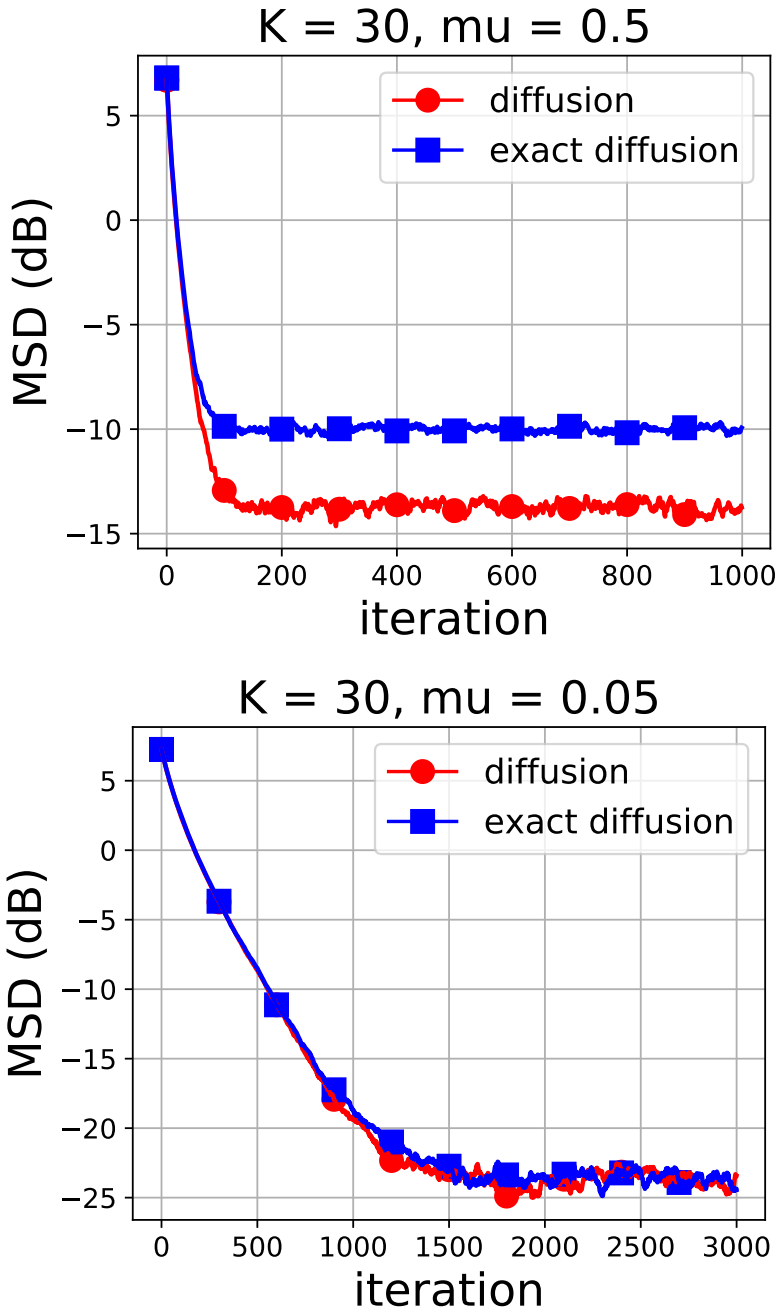

Figure 4.8: Diffusion v.s. exact diffusion over a fully connected network for problem (4.75).

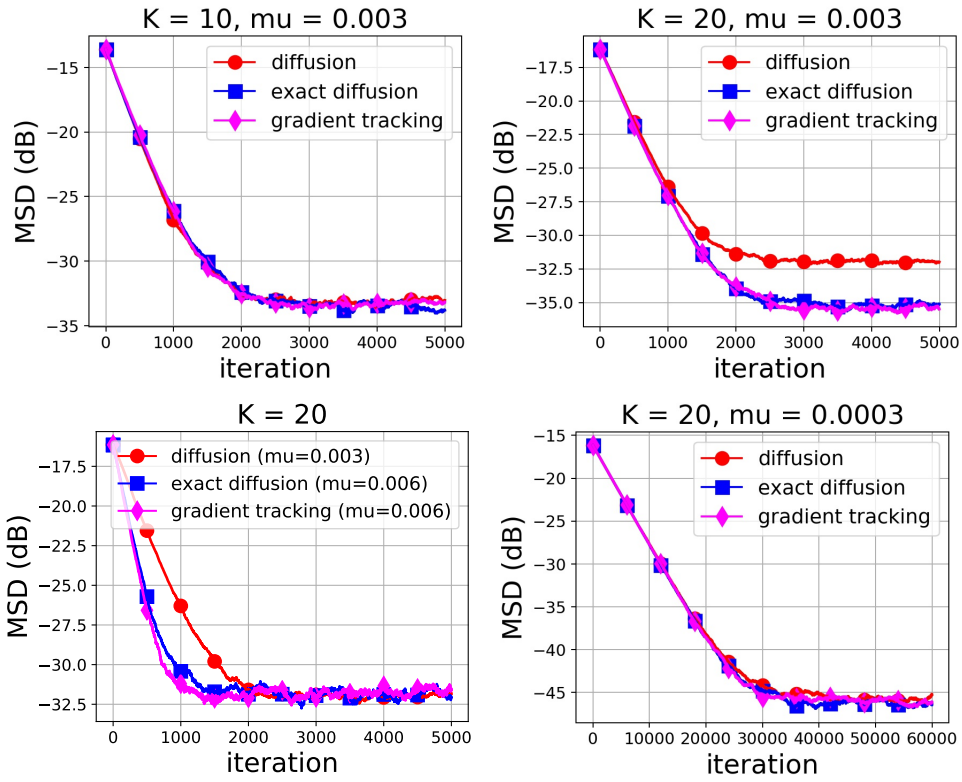

Figure 4.9: Comparison between diffusion [1], exact diffusion (proposed), and gradient tracking [2, 3] over cyclic networks for problem (4.73).

We first compare these two methods over a cyclic network, see the simulation in Figs. 4.6 and 4.7. Similar to Sec. VI.A, the simulation results shown in Figs. 4.6 and 4.7 are also consistent with our discussions in Sec.4.4.2. In the third plot in Fig.4.6, we set $\mu_{\rm d} = 0.003$ and $\mu_{ed} = 0.006$ so that both diffusion and exact diffusion have the same MSD performance. Next, we compare diffusion with exact diffusion over a fully connected network in Fig.4.8. It is observed that the results are consistent with the discussion in Sec.4.4.1.

4.6.3 Comparison with Gradient Tracking Methods

In this subsection we compare exact diffusion with the distributed stochastic gradient tracking method [2, 3]. While [2] shows stochastic gradient tracking has better steady-state MSD performance than decentralized gradient descent (DGD) via numerical simulations, it does not study *when* and *why* gradient tracking can be better DGD. In fact, since gradient tracking

can also be used to correct the bias suffered by diffusion, we can expect the gradient tracking method to have roughly a similar behavior to exact diffusion. In other words, gradient tracking will have better MSD performance than diffusion when the network is sparsely-connected and worse MSD performance when the network is well-connected. Moreover, the difference between diffusion and gradient tracking will diminish for small step-sizes. In this subsection, we verify this conclusion using simulations. We first consider the MSE-network (4.73) over a cyclic network (which is a sparsely-connected network). The results in Fig.4.9 show stochastic gradient tracking behaves as we expected, and it has almost the same performance as exact diffusion in all scenarios. Note though that the gradient tracking method [2,3] requires twice the amount of communication that is required by exact diffusion, which implies exact diffusion is more communication efficient. In the third plot in Fig.4.9, we set $\mu_{\rm d} = 0.003$ and $\mu_{ed} = \mu_{track} = 0.006$ to endow the algorithms with the same steady-state MSD performance.

We next compare diffusion, exact diffusion, and gradient tracking method over a fully-connected network (which is a well-connected network). It is observed in Fig.4.10 that diffusion has the best MSD performance compared to exact diffusion and gradient tracking, which confirms our conclusion. While reference [2] suggests that gradient tracking is superior to consensus, we observe from the analytical results in the current manuscript and from the simulations in Fig.4.10 that there are situations when gradient tracking cannot outperform the traditional diffusion; their performance measures match each other and sometimes gradient tracking can be worse.

4.7 Conclusion

This chapter studies the convergence property of exact diffusion under the stochastic and adaptive setting and compares it with traditional diffusion strategy, which illustrates the influence of bias-correction on distributed stochastic optimization. Conditions are established when exact diffusion can improve, match, or even degrade the performance of diffusion. In particular, it is analytically proven that the superiority of exact diffusion will be more evident over sparsely-connected network topologies. Future work includes improving the

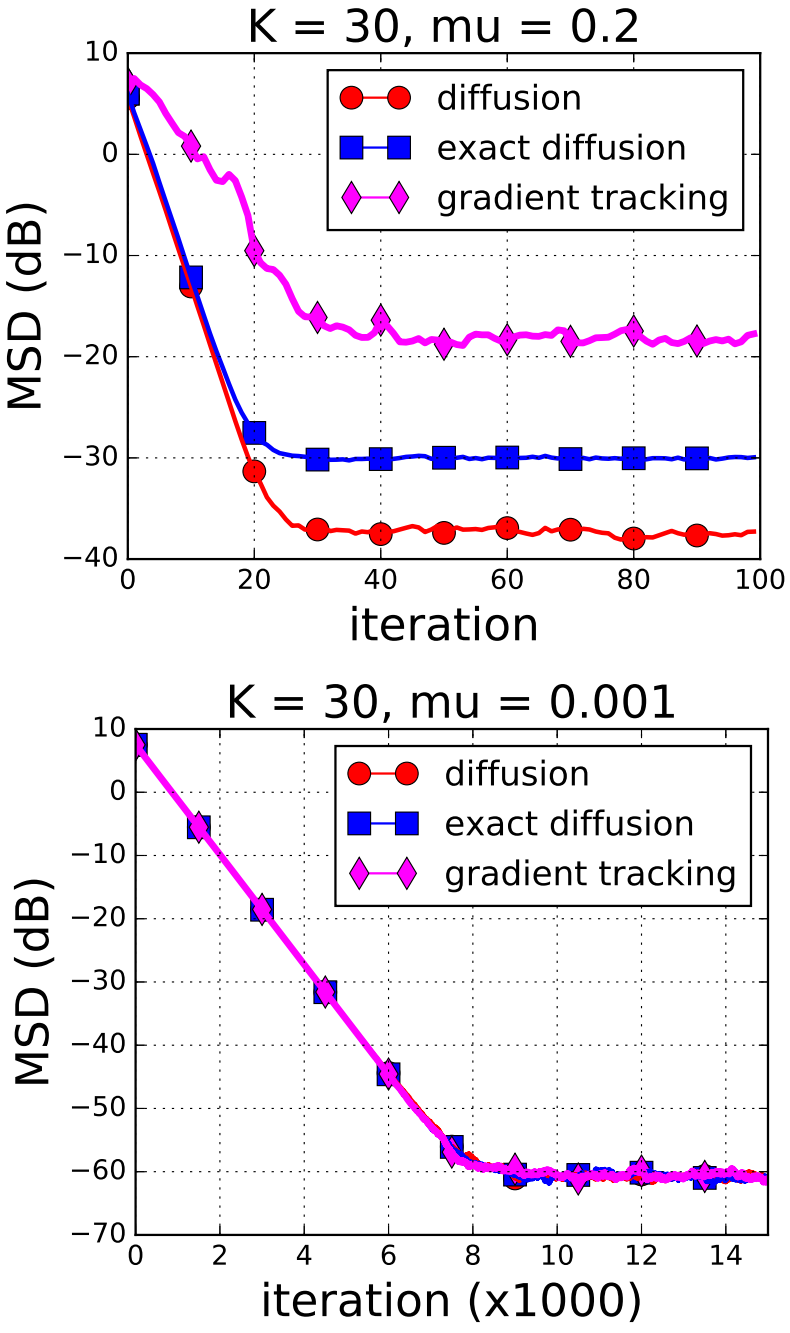

Figure 4.10: Comparison between diffusion [1], exact diffusion (proposed), and gradient tracking [2, 3] over a fully connected network when solving problem (4.73).

current exact diffusion structure so that it can match, or even outperform diffusion over well-connected networks.

4.A Proof of Theorem 4.1

From the first line in the transformed error dynamics (4.34), we know that

$$\bar{\boldsymbol{z}}_i = \Big(I_M - \frac{\mu}{K}\sum_{k=1}^{K}\boldsymbol{H}_{k,i-1}\Big)\bar{\boldsymbol{z}}_{i-1} - \frac{c\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_{i-1}\mathcal{X}_{R,u}\check{\boldsymbol{z}}_{i-1} + \frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{s}_i(\boldsymbol{\mathcal{w}}_{i-1}). \tag{4.76}$$

By squaring and taking conditional expectation of both sides of the recursion and recalling (4.37), we get

$$\begin{aligned}
&\mathbb{E}[\|\bar{\boldsymbol{z}}_i\|^2|\boldsymbol{\mathcal{F}}_{i-1}] = \\
&\Big\|\Big(I-\frac{\mu}{K}\sum_{k=1}^{K}\boldsymbol{H}_{k,i-1}\Big)\bar{\boldsymbol{z}}_{i-1}-\frac{c\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_{i-1}\mathcal{X}_{R,u}\check{\boldsymbol{z}}_{i-1}\Big\|^2 \\
&+\mu^2\mathbb{E}\Big[\Big\|\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i})\Big\|^2\Big|\boldsymbol{\mathcal{F}}_{i-1}\Big].
\end{aligned} \tag{4.77}$$

Next note that

$$\begin{aligned}
&\Big\|\Big(I-\frac{\mu}{K}\sum_{k=1}^{K}\boldsymbol{H}_{k,i-1}\Big)\bar{\boldsymbol{z}}_{i-1}-\frac{c\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_{i-1}\mathcal{X}_{R,u}\check{\boldsymbol{z}}_{i-1}\Big\|^2 \\
&\overset{(a)}{\leq} \frac{1}{1-t}\Big\|I-\frac{\mu}{K}\sum_{k=1}^{K}\boldsymbol{H}_{k,i-1}\Big\|^2\|\bar{\boldsymbol{z}}_{i-1}\|^2 \\
&\quad+\frac{c^2\mu^2}{K^2 t}\|\mathcal{I}\|^2\|\boldsymbol{\mathcal{H}}_{i-1}\|^2\|\mathcal{X}_{R,u}\|^2\|\check{\boldsymbol{z}}_{i-1}\|^2 \\
&\overset{(b)}{\leq} \frac{(1-\mu\nu)^2}{1-t}\|\bar{\boldsymbol{z}}_{i-1}\|^2+\frac{c^2\mu^2\delta^2\|\mathcal{X}_{R,u}\|^2}{Kt}\|\check{\boldsymbol{z}}_{i-1}\|^2 \\
&\overset{(c)}{=} (1-\mu\nu)\|\bar{\boldsymbol{z}}_{i-1}\|^2+\frac{\mu c^2\delta^2\|\mathcal{X}_{R,u}\|^2}{K\nu}\|\check{\boldsymbol{z}}_{i-1}\|^2,
\end{aligned} \tag{4.78}$$

where (a) holds for $t\in(0,1)$ because of Jensen's inequality, and (b) holds since $\nu^2 \leq \|\boldsymbol{\mathcal{H}}_{i-1}\|^2 \leq \delta^2$, $\|\mathcal{I}\|^2 = K$, and $\|I-\frac{\mu}{K}\sum_{k=1}^{K}\boldsymbol{H}_{k,i-1}\|^2 \leq (1-\mu\nu)^2$ when $\mu \leq 1/\delta$. Moreover,

equality (c) holds if we choose $t = \mu\nu$. In addition, recall from (4.40) that

$$\mathbb{E}\Big[\Big\|\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\Big\|^2\Big|\boldsymbol{\mathcal{F}}_{i-1}\Big] \le \frac{\beta^2}{K}\|\widetilde{\boldsymbol{\mathcal{W}}}_{i-1}\|^2+\frac{\sigma^2}{K} \tag{4.79}$$

Moreover, we can bound $\|\widetilde{\boldsymbol{\mathcal{W}}}_{i-1}\|^2$ as

$$\begin{aligned}\|\widetilde{\boldsymbol{\mathcal{W}}}_{i-1}\|^2 &\overset{(4.35)}{=} \|\mathcal{I}\bar{\boldsymbol{z}}_{i-1} + c\mathcal{X}_{R,u}\check{\boldsymbol{z}}_{i-1}\|^2\\ &\le 2\|\mathcal{I}\bar{\boldsymbol{z}}_{i-1}\|^2 + 2c^2\|\mathcal{X}_{R,u}\check{\boldsymbol{z}}_{i-1}\|^2\\ &\le 2K\|\bar{\boldsymbol{z}}_{i-1}\|^2 + 2c^2\|\mathcal{X}_{R,u}\|^2\|\check{\boldsymbol{z}}_{i-1}\|^2.\end{aligned} \tag{4.80}$$

Substituting (4.78), (4.79) and (4.80) into (4.77), we reach

$$\begin{aligned}&\mathbb{E}[\|\bar{\boldsymbol{z}}_i\|^2|\boldsymbol{\mathcal{F}}_{i-1}]\\ &\le (1-\mu\nu+2\mu^2\beta^2)\|\bar{\boldsymbol{z}}_{i-1}\|^2\\ &\quad+\Big(\frac{\mu c^2\delta^2}{K\nu}+\frac{2\mu^2c^2\beta^2}{K}\Big)\|\mathcal{X}_{R,u}\|^2\|\check{\boldsymbol{z}}_{i-1}\|^2+\frac{\mu^2\sigma^2}{K}\\ &\le (1-\mu\nu+2\mu^2\beta^2)\|\bar{\boldsymbol{z}}_{i-1}\|^2\\ &\quad+\Big(\frac{\mu c^2\delta^2}{K\nu}+\frac{2\mu^2c^2\beta^2}{K}\Big)\|\mathcal{X}_{R}\|^2\|\check{\boldsymbol{z}}_{i-1}\|^2+\frac{\mu^2\sigma^2}{K},\end{aligned} \tag{4.81}$$

where the last inequality holds since

$$\begin{aligned}\|\mathcal{X}_{R,u}\|^2 &= \|\begin{bmatrix} I_{KM} & 0\end{bmatrix}\mathcal{X}_R\|^2\\ &\le \|\begin{bmatrix} I_{KM} & 0\end{bmatrix}\|^2\|\mathcal{X}_R\|^2 = \|\mathcal{X}_R\|^2\end{aligned} \tag{4.82}$$

By taking expectation over the filtration, we get

$$\begin{aligned}\mathbb{E}\|\bar{\boldsymbol{z}}_i\|^2 &\le (1-\mu\nu+2\mu^2\beta^2)\mathbb{E}\|\bar{\boldsymbol{z}}_{i-1}\|^2\\ &\quad+\Big(\frac{\mu c^2\delta^2}{K\nu}+\frac{2\mu^2c^2\beta^2}{K}\Big)\|\mathcal{X}_R\|^2\mathbb{E}\|\check{\boldsymbol{z}}_{i-1}\|^2+\frac{\mu^2\sigma^2}{K}.\end{aligned} \tag{4.83}$$

On the other hand, from the second line in (4.34) we have

$$\begin{aligned}\check{\boldsymbol{z}}_i &= \mathcal{D}_1\check{\boldsymbol{z}}_{i-1} - \frac{\mu}{c}\mathcal{D}_1\mathcal{X}_L\boldsymbol{\mathcal{T}}_{i-1}(\mathcal{R}_1\bar{\boldsymbol{z}}_{i-1}+c\mathcal{X}_R\check{\boldsymbol{z}}_{i-1})\\ &\quad+\frac{\mu}{c}\mathcal{D}_1\mathcal{X}_L\mathcal{B}_\ell\boldsymbol{s}_i(\boldsymbol{\mathcal{W}}_{i-1}).\end{aligned} \tag{4.84}$$

By squaring and taking conditional expectation of both sides of the above recursion and recalling (4.37), we get

$$\begin{aligned}
&\mathbb{E}[\|\check{\boldsymbol{z}}_i\|^2|\boldsymbol{\mathcal{F}}_{i-1}] \\
&= \|\mathcal{D}_1\check{\boldsymbol{z}}_{i-1} - \frac{\mu}{c}\mathcal{D}_1\mathcal{X}_L\boldsymbol{\mathcal{T}}_{i-1}(\mathcal{R}_1\bar{\boldsymbol{z}}_{i-1} + c\mathcal{X}_R\check{\boldsymbol{z}}_{i-1})\|^2 \\
&\quad + \frac{\mu^2\|\mathcal{D}_1\|^2}{c^2}\mathbb{E}[\|\mathcal{X}_L\mathcal{B}_\ell\boldsymbol{s}_i(\boldsymbol{\mathcal{W}}_{i-1})\|^2|\boldsymbol{\mathcal{F}}_{i-1}].
\end{aligned} \tag{4.85}$$

Note that

$$\begin{aligned}
&\|\mathcal{D}_1\check{\boldsymbol{z}}_{i-1} - (\mu/c)\mathcal{X}_L\boldsymbol{\mathcal{T}}_{i-1}(\mathcal{R}_1\bar{\boldsymbol{z}}_{i-1} + c\mathcal{X}_R\check{\boldsymbol{z}}_{i-1})\|^2 \\
&\le \frac{1}{t}\|\mathcal{D}_1\check{\boldsymbol{z}}_{i-1}\|^2 + \frac{\mu^2\|\mathcal{D}_1\|^2}{c^2(1-t)}\|\mathcal{X}_L\boldsymbol{\mathcal{T}}_{i-1}(\mathcal{R}_1\bar{\boldsymbol{z}}_{i-1} + c\mathcal{X}_R\check{\boldsymbol{z}}_{i-1})\|^2 \\
&\le \frac{1}{t}\|\mathcal{D}_1\|^2\|\check{\boldsymbol{z}}_{i-1}\|^2 + \frac{2\mu^2\|\mathcal{D}_1\|^2}{c^2(1-t)}\|\mathcal{X}_L\|^2\|\boldsymbol{\mathcal{T}}_{i-1}\|^2\|\mathcal{R}_1\|^2\|\bar{\boldsymbol{z}}_{i-1}\|^2 \\
&\quad + \frac{2\mu^2\|\mathcal{D}_1\|^2}{1-t}\|\mathcal{X}_L\|^2\|\boldsymbol{\mathcal{T}}_{i-1}\|^2\|\mathcal{X}_R\|^2\|\check{\boldsymbol{z}}_{i-1}\|^2,
\end{aligned} \tag{4.86}$$

where $t \in (0,1)$. To simplify the above inequality, we denote

$$\lambda_2 \triangleq \lambda_2(A), \quad \lambda' \triangleq \lambda_2(\bar{A}), \tag{4.87}$$

$$\lambda \triangleq \max\{|\lambda_2(A)|, |\lambda_K(A)|\}. \tag{4.88}$$

Since $\bar{A} = (A + I_K)/2$ and A is doubly-stochastic, we have

$$\lambda' = (1 + \lambda_2)/2 \in (0,1). \tag{4.89}$$

From Lemma 4 in [16] we know that

$$\|\mathcal{D}_1\| = \sqrt{\lambda'} \in (0,1). \tag{4.90}$$

Also, from the definition of $\boldsymbol{\mathcal{T}}_i$ in (4.30), we have

$$\|\boldsymbol{\mathcal{T}}_i\|^2 = \left\|\begin{bmatrix} \boldsymbol{\mathcal{H}}_i & 0 \\ 0 & 0 \end{bmatrix}\right\|^2 \le \delta^2. \tag{4.91}$$

By substituting (4.91) into (4.86), setting $t = \sqrt{\lambda'}$ and recalling $\|\mathcal{R}_1\|^2 = \|\mathcal{I}\|^2 = K$, we get

$$\begin{aligned}
&\|\mathcal{D}_1\check{\boldsymbol{z}}_{i-1} - \frac{\mu}{c}\mathcal{X}_L\boldsymbol{\mathcal{T}}_{i-1}(\mathcal{R}_1\bar{\boldsymbol{z}}_{i-1} + c\mathcal{X}_R\check{\boldsymbol{z}}_{i-1})\|^2 \\
\leq &\Big(\sqrt{\lambda'} + \frac{2\mu^2\delta^2\lambda'\|\mathcal{X}_L\|^2\|\mathcal{X}_R\|^2}{1-\sqrt{\lambda'}}\Big)\|\check{\boldsymbol{z}}_{i-1}\|^2 \\
&+\frac{2K\mu^2\delta^2\|\mathcal{X}_L\|^2\lambda'}{c^2(1-\sqrt{\lambda'})}\|\bar{\boldsymbol{z}}_{i-1}\|^2
\end{aligned} \tag{4.92}$$

In addition, it also holds that

$$\begin{aligned}
&\mathbb{E}[\|\mathcal{X}_L\mathcal{B}_\ell\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2|\boldsymbol{\mathcal{F}}_{i-1}] \\
\leq\ &\|\mathcal{X}_L\|^2\|\mathcal{B}_\ell\|^2\mathbb{E}[\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2|\boldsymbol{\mathcal{F}}_{i-1}] \\
\overset{(a)}{\leq}\ &K\|\mathcal{X}_L\|^2\beta^2\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + K\|\mathcal{X}_L\|^2\sigma^2 \\
\overset{(4.80)}{\leq}\ &2K^2\|\mathcal{X}_L\|^2\beta^2\|\bar{\boldsymbol{z}}_{i-1}\|^2 + 2Kc^2\|\mathcal{X}_{R,u}\|^2\|\mathcal{X}_L\|^2\beta^2\|\check{\boldsymbol{z}}_{i-1}\|^2 \\
&+K\|\mathcal{X}_L\|^2\sigma^2 \\
\overset{(b)}{\leq}\ &\frac{2K^2\|\mathcal{X}_L\|^2\beta^2\|\bar{\boldsymbol{z}}_{i-1}\|^2}{1-\sqrt{\lambda'}} + \frac{2c^2K\|\mathcal{X}_R\|^2\|\mathcal{X}_L\|^2\beta^2\|\check{\boldsymbol{z}}_{i-1}\|^2}{1-\sqrt{\lambda'}} \\
&+K\|\mathcal{X}_L\|^2\sigma^2
\end{aligned} \tag{4.93}$$

where (a) holds because of inequality (4.40) and the fact

$$\|\mathcal{B}_\ell\|^2 = \left\|\mathcal{B}\begin{bmatrix} I_{KM} \\ 0 \end{bmatrix}\right\|^2 \leq \|\mathcal{B}\|^2 = 1$$

in which the last equality holds because of Lemma 4.3. The inequality (b) holds since $1-\sqrt{\lambda'} \in (0,1)$ and inequality (4.82). By substituting (4.92) and (4.93) into (4.85), we have

$$\begin{aligned}
&\mathbb{E}[\|\check{\boldsymbol{z}}_i\|^2|\boldsymbol{\mathcal{F}}_{i-1}] \\
\leq &\Big(\sqrt{\lambda'} + \frac{2\lambda'\mu^2(\delta^2+K\beta^2)\|\mathcal{X}_L\|^2\|\mathcal{X}_R\|^2}{1-\sqrt{\lambda'}}\Big)\|\check{\boldsymbol{z}}_{i-1}\|^2 \\
&+\frac{2\lambda'K\mu^2(\delta^2+K\beta^2)\|\mathcal{X}_L\|^2}{(1-\sqrt{\lambda'})c^2}\|\bar{\boldsymbol{z}}_{i-1}\|^2 \\
&+\frac{\mu^2\lambda'K\|\mathcal{X}_L\|^2\sigma^2}{c^2}
\end{aligned} \tag{4.94}$$

By taking expectation over the filtration, we get

$$\begin{aligned}
&\mathbb{E}\|\check{\boldsymbol{z}}_i\|^2 \\
&\leq \Big(\sqrt{\lambda'} + \frac{2\lambda'\mu^2(\delta^2+K\beta^2)\|\mathcal{X}_L\|^2\|\mathcal{X}_R\|^2}{1-\sqrt{\lambda'}}\Big)\mathbb{E}\|\check{\boldsymbol{z}}_{i-1}\|^2 \\
&\quad + \frac{2K\lambda'\mu^2(\delta^2+K\beta^2)\|\mathcal{X}_L\|^2}{(1-\sqrt{\lambda'})c^2}\mathbb{E}\|\bar{\boldsymbol{z}}_{i-1}\|^2 \\
&\quad + \frac{\lambda'\mu^2 K}{c^2}\|\mathcal{X}_L\|^2\sigma^2
\end{aligned} \tag{4.95}$$

To simplify notation, we introduce the constants

$$c_1 = \|\mathcal{X}_L\|^2, \quad c_2 = \|\mathcal{X}_R\|^2. \tag{4.96}$$

Combining (4.83) and (4.95), we have

$$\begin{aligned}
\begin{bmatrix} \mathbb{E}\|\bar{\boldsymbol{z}}_i\|^2 \\ \mathbb{E}\|\check{\boldsymbol{z}}_i\|^2 \end{bmatrix} \leq &\begin{bmatrix} 1-\mu\nu+2\mu^2\beta^2 & \Big(\frac{\mu c^2\delta^2}{K\nu}+\frac{2\mu^2c^2\beta^2}{K}\Big)c_2 \\ \frac{2K\lambda'\mu^2(\delta^2+K\beta^2)c_1}{(1-\sqrt{\lambda'})c^2} & \sqrt{\lambda'}+\frac{2\mu^2\lambda'(\delta^2+K\beta^2)c_1c_2}{1-\sqrt{\lambda'}} \end{bmatrix} \\
&\times \begin{bmatrix} \mathbb{E}\|\bar{\boldsymbol{z}}_{i-1}\|^2 \\ \mathbb{E}\|\check{\boldsymbol{z}}_{i-1}\|^2 \end{bmatrix} + \begin{bmatrix} \frac{1}{K}\mu^2\sigma^2 \\ \frac{K\lambda'c_1}{c^2}\mu^2\sigma^2 \end{bmatrix}.
\end{aligned} \tag{4.97}$$

Note that c is a parameter that can be set to any positive value. If we let $c^2 = Kc_1$, then the above inequality becomes

$$\begin{aligned}
\begin{bmatrix} \mathbb{E}\|\bar{\boldsymbol{z}}_i\|^2 \\ \mathbb{E}\|\check{\boldsymbol{z}}_i\|^2 \end{bmatrix} \leq &\begin{bmatrix} 1-\mu\nu+2\mu^2\beta^2 & \Big(\frac{\mu\delta^2}{\nu}+2\mu^2\beta^2\Big)c_1c_2 \\ \frac{2\lambda'\mu^2(\delta^2+K\beta^2)}{1-\sqrt{\lambda'}} & \sqrt{\lambda'}+\frac{2\lambda'\mu^2(\delta^2+K\beta^2)c_1c_2}{1-\sqrt{\lambda'}} \end{bmatrix} \\
&\times \begin{bmatrix} \mathbb{E}\|\bar{\boldsymbol{z}}_{i-1}\|^2 \\ \mathbb{E}\|\check{\boldsymbol{z}}_{i-1}\|^2 \end{bmatrix} + \begin{bmatrix} \frac{1}{K}\mu^2\sigma^2 \\ \lambda'\mu^2\sigma^2 \end{bmatrix}.
\end{aligned} \tag{4.98}$$

If we choose μ sufficiently small such that

$$1-\mu\nu+2\mu^2\beta^2 \leq 1-\frac{1}{2}\mu\nu, \tag{4.99}$$

$$\Big(\frac{\mu\delta^2}{\nu}+2\mu^2\beta^2\Big)c_1c_2 \leq \frac{2\mu\delta^2c_1c_2}{\nu}, \tag{4.100}$$

$$\frac{2\lambda'\mu^2(\delta^2+K\beta^2)}{1-\sqrt{\lambda'}} \leq \frac{1}{4}\lambda'\mu\nu, \tag{4.101}$$

$$\sqrt{\lambda'}+\frac{2\lambda'\mu^2(\delta^2+K\beta^2)c_1c_2}{1-\sqrt{\lambda'}} \leq \frac{1+\sqrt{\lambda'}}{2}, \tag{4.102}$$

then inequality (4.98) becomes

$$\begin{bmatrix} \mathbb{E}\|\bar{\boldsymbol{z}}_i\|^2 \\ \mathbb{E}\|\check{\boldsymbol{z}}_i\|^2 \end{bmatrix} \leq \underbrace{\begin{bmatrix} 1-\frac{1}{2}\mu\nu & \frac{2\mu\delta^2 c_1 c_2}{\nu} \\ \frac{1}{4}\lambda'\mu\nu & \frac{1+\sqrt{\lambda'}}{2} \end{bmatrix}}_{\triangleq\, \mathcal{C}} \begin{bmatrix} \mathbb{E}\|\bar{\boldsymbol{z}}_{i-1}\|^2 \\ \mathbb{E}\|\check{\boldsymbol{z}}_{i-1}\|^2 \end{bmatrix} + \begin{bmatrix} \frac{1}{K}\mu^2\sigma^2 \\ \lambda'\mu^2\sigma^2 \end{bmatrix}. \tag{4.103}$$

To satisfy (4.99)–(4.102), it is enough to let μ satisfy

$$\mu \leq \frac{(1-\sqrt{\lambda'})\nu}{(8+4c_1c_2+\sqrt{4c_1c_2})(\delta^2+K\beta^2)}, \tag{4.104}$$

Also, note that $1-\sqrt{\lambda'} = (1-\lambda')/(1+\sqrt{\lambda'})$. Since $0<\lambda'<1$, we have $(1-\lambda')/2 < 1-\sqrt{\lambda'} < 1-\lambda'$. Moreover, since $\lambda' = (1+\lambda_2)/2$ (see (4.89)), we have

$$\frac{1-\lambda_2}{4} < 1-\sqrt{\lambda'} < \frac{1-\lambda_2}{2}. \tag{4.105}$$

From (4.88) we have $|\lambda_2| \leq \lambda$, which further implies $-\lambda \leq \lambda_2 \leq \lambda$. This together with (4.105) leads to

$$\frac{1-\lambda}{4} < 1-\sqrt{\lambda'} < \frac{1+\lambda}{2}. \tag{4.106}$$

With relation (4.106), we know that if μ satisfies

$$\mu \leq \frac{(1-\lambda)\nu}{(32+16c_1c_2+8\sqrt{c_1c_2})(\delta^2+K\beta^2)}, \tag{4.107}$$

then μ must also satisfy (4.104). Recall that $\beta^2 = \frac{\max_k\{\beta_k^2\}}{K}$, we have $K\beta^2 = \beta_{max}^2 = \max_k\{\beta_k^2\}$.

Next we examine the spectral radius of the matrix $\mathcal{C}$. Note that $\lambda' \in (0,1)$, it is easy to verify that

$$\begin{aligned} \rho(\mathcal{C}) \leq \|\mathcal{C}\|_1 &= \max\left\{1-\frac{\mu\nu}{2}+\frac{\lambda'\mu\nu}{4},\ \frac{1+\sqrt{\lambda'}}{2}+\frac{2\mu\delta^2c_1c_2}{\nu}\right\} \\ &\leq \max\left\{1-\frac{\mu\nu}{4},\ \frac{1+\sqrt{\lambda'}}{2}+\frac{2\mu\delta^2c_1c_2}{\nu}\right\} \\ &\overset{(4.104)}{\leq} 1-\frac{1}{4}\mu\nu < 1, \end{aligned} \tag{4.108}$$

and therefore $\mathcal{C}$ is a stable matrix, and $\rho(C) = 1 - O(\mu\nu)$ is the convergence rate of $\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_i\|^2$. Next we examine:

$$
\begin{aligned}
&(I-\mathcal{C})^{-1} \\
&\quad= \begin{bmatrix} \frac{\mu\nu}{2} & -\frac{2\mu\delta^2 c_1 c_2}{\nu} \\ -\frac{\lambda'\mu\nu}{4} & \frac{1-\sqrt{\lambda'}}{2} \end{bmatrix}^{-1} \\
&\quad= \frac{4}{(1-\sqrt{\lambda'})\mu\nu - 2\lambda'\mu^2\delta^2 c_1 c_2} \begin{bmatrix} \frac{1-\sqrt{\lambda'}}{2} & \frac{2\mu\delta^2 c_1 c_2}{\nu} \\ \frac{\mu\nu\lambda'}{4} & \frac{\mu\nu}{2} \end{bmatrix} \\
&\quad\overset{(a)}{\leq} \frac{8}{\mu\nu(1-\sqrt{\lambda'})} \begin{bmatrix} \frac{1-\sqrt{\lambda'}}{2} & \frac{2\mu\delta^2 c_1 c_2}{\nu} \\ \frac{\mu\nu\lambda'}{4} & \frac{\mu\nu}{2} \end{bmatrix} \\
&\quad= \begin{bmatrix} \frac{4}{\mu\nu} & \frac{16\delta^2 c_1 c_2}{\nu^2(1-\sqrt{\lambda'})} \\ \frac{2\lambda'}{1-\sqrt{\lambda'}} & \frac{4}{1-\sqrt{\lambda'}} \end{bmatrix},
\end{aligned} \tag{4.109}
$$

where inequality (a) holds since

$$
(1-\sqrt{\lambda'})\mu\nu - 2\lambda'\mu^2\delta^2 c_1 c_2 \geq \frac{(1-\sqrt{\lambda'})\mu\nu}{2} \tag{4.110}
$$

when μ satisfies (4.104). By iterating (4.103), we conclude that

$$
\begin{aligned}
\limsup_{i\to\infty} \begin{bmatrix} \mathbb{E}\|\bar{\boldsymbol{z}}_i\|^2 \\ \mathbb{E}\|\check{\boldsymbol{z}}_i\|^2 \end{bmatrix} &\leq (I-\mathcal{C})^{-1} \begin{bmatrix} \frac{1}{K}\mu^2\sigma^2 \\ \lambda'\mu^2\sigma^2 \end{bmatrix} \\
&\overset{(4.109)}{=} \begin{bmatrix} \frac{4\mu\sigma^2}{K\nu} + \frac{16\lambda'\delta^2 c_1 c_2 \mu^2\sigma^2}{\nu^2(1-\sqrt{\lambda'})} \\ \frac{2\lambda'\mu^2\sigma^2}{K(1-\sqrt{\lambda'})} + \frac{4\lambda'\mu^2\sigma^2}{1-\sqrt{\lambda'}} \end{bmatrix}.
\end{aligned} \tag{4.111}
$$

As a result, we obtain

$$\begin{aligned}
&\limsup_{i\to\infty}\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2\\
&\overset{(4.80)}{\leq}\limsup_{i\to\infty}\big(2K\mathbb{E}\|\bar{\boldsymbol{z}}_i\|^2+2Kc_1c_2\mathbb{E}\|\check{\boldsymbol{z}}_i\|^2\big)\\
&\overset{(4.111)}{\leq}\frac{8\mu\sigma^2}{\nu}+\frac{(32K\delta^2+4\nu^2+8K\nu^2)\lambda' c_1c_2\mu^2\sigma^2}{\nu^2(1-\sqrt{\lambda'})}\\
&\leq\frac{8\mu\sigma^2}{\nu}+\frac{44K\delta^2\lambda' c_1c_2\mu^2\sigma^2}{\nu^2(1-\sqrt{\lambda'})}\\
&\overset{(4.106)}{\leq}\frac{8\mu\sigma^2}{\nu}+\frac{176K\lambda' c_1c_2\delta^2\mu^2\sigma^2}{\nu^2(1-\lambda)}\\
&\overset{(a)}{\leq}\frac{8\mu\sigma^2}{\nu}+\frac{88K(1+\lambda)c_1c_2\delta^2\mu^2\sigma^2}{\nu^2(1-\lambda)}\\
&\overset{(b)}{=}O\left(\frac{\mu\sigma^2}{\nu}+\frac{K\delta^2}{\nu^2}\cdot\frac{\mu^2\sigma^2}{1-\lambda}\right)
\end{aligned}\tag{4.112}$$

where (a) holds because $\lambda'=(1+\lambda_2(A))/2\leq(1+\lambda)/2$ and (b) holds because $\lambda<1$. Result (4.112) leads to (4.42) by dividing K to both sides of (4.112).

4.B Proof of Lemma 4.5

This section establishes the mean-square convergence of diffusion. With definition (4.12), we can rewrite diffusion recursions (4.2)–(4.3) as

$$\boldsymbol{\mathcal{w}}_i=\mathcal{A}\big(\boldsymbol{\mathcal{w}}_{i-1}-\mu\nabla\mathcal{Q}(\boldsymbol{\mathcal{w}}_{i-1};\boldsymbol{\mathcal{x}}_i)\big).\tag{4.113}$$

With relation (4.24), the above recursion becomes

$$\boldsymbol{\mathcal{w}}_i=\mathcal{A}\big(\boldsymbol{\mathcal{w}}_{i-1}-\mu\nabla\mathcal{J}(\boldsymbol{\mathcal{w}}_{i-1})-\mu\boldsymbol{\mathcal{s}}_i(\boldsymbol{\mathcal{w}}_{i-1})\big),\tag{4.114}$$

which also leads to

$$\begin{aligned}
\widetilde{\boldsymbol{\mathcal{w}}}_i&=\mathcal{A}\big(\widetilde{\boldsymbol{\mathcal{w}}}_{i-1}+\mu\nabla\mathcal{J}(\boldsymbol{\mathcal{w}}_{i-1})+\mu\boldsymbol{\mathcal{s}}_i(\boldsymbol{\mathcal{w}}_{i-1})\big)\\
&=\mathcal{A}\big(\widetilde{\boldsymbol{\mathcal{w}}}_{i-1}+\mu\nabla\mathcal{J}(\boldsymbol{\mathcal{w}}_{i-1})-\mu\nabla\mathcal{J}(w^\star)\big)\\
&\quad+\mu\mathcal{A}\nabla\mathcal{J}(w^\star)+\mu\mathcal{A}\boldsymbol{\mathcal{s}}_i(\boldsymbol{\mathcal{w}}_{i-1})\\
&\overset{(4.28)}{=}\mathcal{A}\bigg((I-\mu\boldsymbol{\mathcal{H}}_{i-1})\widetilde{\boldsymbol{\mathcal{w}}}_{i-1}+\mu h+\mu\boldsymbol{\mathcal{s}}_i(\boldsymbol{\mathcal{w}}_{i-1})\bigg),
\end{aligned}\tag{4.115}$$

where $\widetilde{\boldsymbol{w}}_i = w^\star - \boldsymbol{w}_i$ and $h \triangleq \nabla \mathcal{J}(w^\star)$. Note that $\mathcal{A} = A \otimes I_M$ is symmetric and doubly stochastic, it holds that

$$\mathcal{A} = \underbrace{\begin{bmatrix} \mathcal{I} & c\mathcal{X}_R \end{bmatrix}}_{\mathcal{X}} \begin{bmatrix} I_M & 0 \\ 0 & \Lambda \end{bmatrix} \underbrace{\begin{bmatrix} \frac{1}{K}\mathcal{I}^\mathsf{T} \\ \frac{1}{c}\mathcal{X}_L \end{bmatrix}}_{\mathcal{X}^{-1}}, \tag{4.116}$$

where $\mathcal{I} = \mathbb{1}_K \otimes I_M$ and $\lambda \triangleq \|\Lambda\| = \max\{|\lambda_2(A)|, |\lambda_K(A)|\} < 1$. Note that $\mathcal{X}_R$ and $\mathcal{X}_L$ are different matrices from the ones defined in (4.31). Now we define

$$\begin{bmatrix} \bar{\boldsymbol{w}}_i \\ \check{\boldsymbol{w}}_i \end{bmatrix} \triangleq \mathcal{X}^{-1}\widetilde{\boldsymbol{w}}_i \tag{4.117}$$

and multiply $\mathcal{X}^{-1}$ to both sides of (4.115), it holds that

$$\begin{aligned} \begin{bmatrix} \bar{\boldsymbol{w}}_i \\ \check{\boldsymbol{w}}_i \end{bmatrix} = &\begin{bmatrix} I_M - \frac{\mu}{K}\sum_{k=1}^K \boldsymbol{H}_{k,i-1} & -\frac{c\mu}{K}\mathcal{I}^\mathsf{T}\boldsymbol{\mathcal{H}}_{i-1}\mathcal{X}_R \\ -\frac{\mu}{c}\Lambda\mathcal{X}_L\boldsymbol{\mathcal{H}}_{i-1}\mathcal{I} & \Lambda - \mu\Lambda\mathcal{X}_L\boldsymbol{\mathcal{H}}_{i-1}\mathcal{X}_R \end{bmatrix} \\ &\times \begin{bmatrix} \bar{\boldsymbol{w}}_{i-1} \\ \check{\boldsymbol{w}}_{i-1} \end{bmatrix} + \begin{bmatrix} \frac{\mu}{K}\mathcal{I}^\mathsf{T}h \\ \frac{\mu}{c}\Lambda\mathcal{X}_L h \end{bmatrix} + \begin{bmatrix} \frac{\mu}{K}\mathcal{I}^\mathsf{T} \\ \frac{\mu}{c}\Lambda\mathcal{X}_L \end{bmatrix} \boldsymbol{s}_i(\boldsymbol{w}_{i-1}). \end{aligned} \tag{4.118}$$

For notational simplicity, we further define

$$\check{h} \triangleq \frac{1}{c}\Lambda\mathcal{X}_L h, \tag{4.119}$$

$$\bar{\boldsymbol{s}}_i \triangleq \frac{1}{K}\mathcal{I}^\mathsf{T}\boldsymbol{s}_i(\widetilde{\boldsymbol{w}}_{i-1}), \tag{4.120}$$

$$\check{\boldsymbol{s}}_i \triangleq \frac{1}{c}\Lambda\mathcal{X}_L\boldsymbol{s}_i(\widetilde{\boldsymbol{w}}_{i-1}), \tag{4.121}$$

Recalling that $h = \nabla\mathcal{J}(w^\star)$ and, thus, $\mathcal{I}^\mathsf{T}h = \sum_{k=1}^K \nabla J_k(w^\star) = 0$. Therefore, recursion (4.118) becomes

$$\begin{aligned} \begin{bmatrix} \bar{\boldsymbol{w}}_i \\ \check{\boldsymbol{w}}_i \end{bmatrix} = &\begin{bmatrix} I_M - \frac{\mu}{K}\sum_{k=1}^K \boldsymbol{H}_{k,i-1} & -\frac{c\mu}{K}\mathcal{I}^\mathsf{T}\boldsymbol{\mathcal{H}}_{i-1}\mathcal{X}_R \\ -\frac{\mu}{c}\Lambda\mathcal{X}_L\boldsymbol{\mathcal{H}}_{i-1}\mathcal{I} & \Lambda - \mu\Lambda\mathcal{X}_L\boldsymbol{\mathcal{H}}_{i-1}\mathcal{X}_R \end{bmatrix} \\ &\times \begin{bmatrix} \bar{\boldsymbol{w}}_{i-1} \\ \check{\boldsymbol{w}}_{i-1} \end{bmatrix} + \begin{bmatrix} 0 \\ \mu\check{h} \end{bmatrix} + \begin{bmatrix} \mu\bar{\boldsymbol{s}}_i \\ \mu\check{\boldsymbol{s}}_i \end{bmatrix}. \end{aligned} \tag{4.122}$$

In the first line of the above transformed recursion, we have

$$\begin{aligned}\bar{\boldsymbol{w}}_i =&\big(I_M - \frac{\mu}{K}\sum_{k=1}^{K}\boldsymbol{H}_{k,i-1}\big)\bar{\boldsymbol{w}}_{i-1}\\ &- \frac{c\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_{i-1}\mathcal{X}_R\check{\boldsymbol{w}}_{i-1} + \mu\bar{\boldsymbol{s}}_i.\end{aligned} \tag{4.123}$$

By following arguments in (4.76)–(4.83), we reach

$$\begin{aligned}\mathbb{E}\|\bar{\boldsymbol{w}}_i\|^2 \le& (1-\mu\nu+2\mu^2\beta^2)\mathbb{E}\|\bar{\boldsymbol{w}}_{i-1}\|^2\\ &+ \big(\frac{c^2\delta^2\mu}{K\nu} + \frac{2c^2\beta^2\mu^2}{K}\big)\|\mathcal{X}_R\|^2\mathbb{E}\|\check{\boldsymbol{w}}_{i-1}\|^2 + \frac{\mu^2\sigma^2}{K}.\end{aligned} \tag{4.124}$$

In the second line of (4.122), we have

$$\begin{aligned}\check{\boldsymbol{w}}_i =&(\Lambda - \mu\Lambda\mathcal{X}_L\boldsymbol{\mathcal{H}}_{i-1}\mathcal{X}_R)\check{\boldsymbol{w}}_{i-1}\\ &- \frac{\mu}{c}\Lambda\mathcal{X}_L\boldsymbol{\mathcal{H}}_{i-1}\mathcal{I}\bar{\boldsymbol{w}}_{i-1} + \mu\check{h} + \mu\check{\boldsymbol{s}}_i.\end{aligned} \tag{4.125}$$

By following arguments similar to the ones in (4.84)–(4.95), we have

$$\begin{aligned}&\mathbb{E}\|\check{\boldsymbol{w}}_i\|^2\\ \le& \Big(\lambda + \frac{3\mu^2\lambda^2(\delta^2+K\beta^2)\|\mathcal{X}_L\|^2\|\mathcal{X}_R\|^2}{1-\lambda}\Big)\mathbb{E}\|\check{\boldsymbol{w}}_{i-1}\|^2\\ &+ \frac{3K\mu^2\lambda^2(\delta^2+K\beta^2)\|\mathcal{X}_L\|^2}{(1-\lambda)c^2}\mathbb{E}\|\bar{\boldsymbol{w}}_{i-1}\|^2\\ &+ \frac{3\mu^2\lambda^2\|\mathcal{X}_L\|^2\|h\|^2}{(1-\lambda)c^2} + \frac{K\mu^2\lambda^2}{c^2}\|\mathcal{X}_L\|^2\sigma^2\end{aligned} \tag{4.126}$$

To simplify notation, we introduce the constants

$$e_1 = \|\mathcal{X}_L\|^2, \quad e_2 = \|\mathcal{X}_R\|^2, \quad b^2 = \|h\|^2/K. \tag{4.127}$$

Meanwhile, we also set $c^2 = e_1K$ in (4.124) and (4.126). With these notations and operations, we combine (4.124) and (4.126) to get

$$\begin{aligned}\begin{bmatrix}\mathbb{E}\|\bar{\boldsymbol{w}}_i\|^2\\ \mathbb{E}\|\check{\boldsymbol{w}}_i\|^2\end{bmatrix} \le& \begin{bmatrix}1-\mu\nu+2\mu^2\beta^2 & \Big(\frac{\mu\delta^2}{\nu}+2\mu^2\beta^2\Big)e_1e_2\\ \frac{3\mu^2\lambda^2(\delta^2+K\beta^2)}{1-\lambda} & \lambda + \frac{3\mu^2\lambda^2(\delta^2+K\beta^2)e_1e_2}{1-\lambda}\end{bmatrix}\\ &\times\begin{bmatrix}\mathbb{E}\|\bar{\boldsymbol{w}}_{i-1}\|^2\\ \mathbb{E}\|\check{\boldsymbol{w}}_{i-1}\|^2\end{bmatrix} + \begin{bmatrix}\frac{1}{K}\mu^2\sigma^2\\ \mu^2\lambda^2\sigma^2 + \frac{3\mu^2\lambda^2b^2}{1-\lambda}\end{bmatrix}.\end{aligned} \tag{4.128}$$

If we choose sufficiently small μ such that

$$1-\mu\nu+2\mu^2\beta^2 \le 1-\frac{1}{2}\mu\nu, \tag{4.129}$$

$$\Big(\frac{\mu\delta^2}{\nu}+2\mu^2\beta^2\Big)e_1e_2 \le \frac{2\mu\delta^2 e_1e_2}{\nu}, \tag{4.130}$$

$$\frac{3\mu^2\lambda^2(\delta^2+K\beta^2)}{1-\lambda} \le \frac{1}{4}\mu\lambda^2\nu, \tag{4.131}$$

$$\lambda+\frac{3\mu^2\lambda^2(\delta^2+K\beta^2)e_1e_2}{1-\lambda} \le \frac{1+\lambda}{2}, \tag{4.132}$$

then inequality (4.128) becomes

$$\begin{bmatrix} \mathbb{E}\|\bar{\boldsymbol{\mathcal{W}}}_i\|^2 \\ \mathbb{E}\|\check{\boldsymbol{\mathcal{W}}}_i\|^2 \end{bmatrix} \le \underbrace{\begin{bmatrix} 1-\frac{\mu\nu}{2} & \frac{2\mu\delta^2e_1e_2}{\nu} \\ \frac{\mu\lambda^2\nu}{4} & \frac{1+\lambda}{2} \end{bmatrix}}_{\triangleq\,\mathcal{C}} \begin{bmatrix} \mathbb{E}\|\bar{\boldsymbol{\mathcal{W}}}_{i-1}\|^2 \\ \mathbb{E}\|\check{\boldsymbol{\mathcal{W}}}_{i-1}\|^2 \end{bmatrix} + \begin{bmatrix} \frac{1}{K}\mu^2\sigma^2 \\ \mu^2\lambda^2\sigma^2+\frac{3\mu^2\lambda^2b^2}{1-\lambda} \end{bmatrix}. \tag{4.133}$$

To make inequalities (4.129)–(4.132) hold, it is enough to set

$$\mu \le \frac{(1-\lambda)\nu}{(12+4e_1e_2+\lambda\sqrt{6e_1e_2})(\delta^2+K\beta^2)} = O\left(\frac{(1-\lambda)\nu}{\delta^2+K\beta^2}\right). \tag{4.134}$$

Note that $K\beta^2=\beta_{max}^2$. Similar to (4.108), it can be easily verified that when μ satisfies (4.134), we have that $\rho(\mathcal{C})<1$. Moreover, we also have

$$\begin{aligned}(I-\mathcal{C})^{-1} &= \begin{bmatrix} \frac{\mu\nu}{2} & -\frac{2\mu\delta^2e_1e_2}{\nu} \\ -\frac{\mu\lambda^2\nu}{4} & \frac{1-\lambda}{2} \end{bmatrix}^{-1} \\ &= \frac{1}{\frac{\mu\nu(1-\lambda)}{4}-\frac{\mu^2\delta^2\lambda^2e_1e_2}{2}} \begin{bmatrix} \frac{1-\lambda}{2} & \frac{2\mu\delta^2e_1e_2}{\nu} \\ \frac{\mu\lambda^2\nu}{4} & \frac{\mu\nu}{2} \end{bmatrix} \\ &\overset{(a)}{\le} \frac{8}{\mu\nu(1-\lambda)} \begin{bmatrix} \frac{1-\lambda}{2} & \frac{2\mu\delta^2e_1e_2}{\nu} \\ \frac{\mu\lambda^2\nu}{4} & \frac{\mu\nu}{2} \end{bmatrix} \\ &= \begin{bmatrix} \frac{4}{\mu\nu} & \frac{16\delta^2e_1e_2}{\nu^2(1-\lambda)} \\ \frac{2\lambda^2}{1-\lambda} & \frac{4}{1-\lambda} \end{bmatrix}. \end{aligned} \tag{4.135}$$

where step (a) denotes entry-wise inequality, which holds because

$$\frac{\mu\nu(1-\lambda)}{4}-\frac{\mu^2\delta^2\lambda^2e_1e_2}{2} \ge \frac{\mu\nu(1-\lambda)}{8} \tag{4.136}$$

when μ satisfies (4.134). By iterating (4.133), we get

$$\begin{aligned}
&\limsup_{i\to\infty} \begin{bmatrix} \mathbb{E}\|\bar{\boldsymbol{\mathcal{W}}}_i\|^2 \\ \mathbb{E}\|\check{\boldsymbol{\mathcal{W}}}_i\|^2 \end{bmatrix} \\
&\quad = (I-\mathcal{C})^{-1} \begin{bmatrix} \frac{1}{K}\mu^2\sigma^2 \\ \mu^2\sigma^2 + \frac{3\mu^2 b^2}{1-\lambda} \end{bmatrix} \\
&\quad = \begin{bmatrix} \frac{4}{\mu\nu} & \frac{16\delta^2 e_1 e_2}{\nu^2(1-\lambda)} \\ \frac{2\lambda^2}{1-\lambda} & \frac{4}{1-\lambda} \end{bmatrix} \begin{bmatrix} \frac{1}{K}\mu^2\sigma^2 \\ \mu^2\lambda^2\sigma^2 + \frac{3\mu^2\lambda^2 b^2}{1-\lambda} \end{bmatrix} \\
&\quad = \begin{bmatrix} \frac{4\mu\sigma^2}{K\nu} + \frac{16\delta^2 e_1 e_2 \mu^2\lambda^2\sigma^2}{\nu^2(1-\lambda)} + \frac{48\delta^2 e_1 e_2\mu^2\lambda^2 b^2}{\nu^2(1-\lambda)^2} \\ \frac{2\mu^2\lambda^2\sigma^2}{K(1-\lambda)} + \frac{4\mu^2\lambda^2\sigma^2}{1-\lambda} + \frac{12\mu^2\lambda^2 b^2}{(1-\lambda)^2} \end{bmatrix}
\end{aligned} \tag{4.137}$$

Therefore,

$$\begin{aligned}
&\limsup_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_i\|^2 \\
&\overset{(4.117)}{\leq} \limsup_{i\to\infty} \left(2K\mathbb{E}\|\bar{\boldsymbol{\mathcal{W}}}_i\|^2 + 2Ke_1e_2\mathbb{E}\|\check{\boldsymbol{\mathcal{W}}}_i\|^2\right) \\
&\quad = \frac{8\mu\sigma^2}{\nu} + \frac{(4\nu^2 + 32K\delta^2 + 8K\nu^2)e_1e_2\mu^2\lambda^2\sigma^2}{\nu^2(1-\lambda)} \\
&\qquad + \frac{(96\delta^2 + 24\nu^2)Ke_1e_2\mu^2\lambda^2 b^2}{\nu^2(1-\lambda)^2} \\
&\quad \leq \frac{8\mu\sigma^2}{\nu} + \frac{44Ke_1e_2\delta^2\mu^2\lambda^2\sigma^2}{\nu^2(1-\lambda)} + \frac{120Ke_1e_2\delta^2\mu^2\lambda^2 b^2}{\nu^2(1-\lambda)^2} \\
&\quad = O\left(\frac{\mu\sigma^2}{\nu} + \frac{\delta^2}{\nu^2}\cdot\frac{K\mu^2\lambda^2\sigma^2}{(1-\lambda)} + \frac{\delta^2}{\nu^2}\cdot\frac{K\mu^2\lambda^2 b^2}{(1-\lambda)^2}\right).
\end{aligned} \tag{4.138}$$

This leads to (4.43) by dividing K to both sides of (4.138).

4.C Proof of Theorem 4.2

The derivation of the MSD expression adjusts the arguments from [1, Ch. 11] to our case. We start by introducing

$$\mathcal{C} \triangleq \begin{bmatrix} I_M - \frac{\mu}{K}\sum_{k=1}^{K} H_k & -\frac{c\mu}{K}\mathcal{I}^{\mathsf{T}}\mathcal{H}\mathcal{X}_{R,u} \\ -\frac{\mu}{c}\mathcal{D}_1\mathcal{X}_L\mathcal{T}\mathcal{R}_1 & \mathcal{D}_1 - \mu\mathcal{D}_1\mathcal{X}_L\mathcal{T}\mathcal{X}_R \end{bmatrix}, \tag{4.139}$$

$$\mathcal{G} \triangleq \begin{bmatrix} \frac{1}{K}\mathcal{I}^{\mathsf{T}} \\ \frac{1}{c}\mathcal{D}_1\mathcal{X}_L\mathcal{B}_\ell \end{bmatrix}, \qquad \boldsymbol{s}_i \triangleq \boldsymbol{s}_i(\boldsymbol{\mathcal{W}}_{i-1}), \tag{4.140}$$

With these definitions, we can rewrite the approximate error dynamics (4.59) as $\boldsymbol{\mathcal{Z}}'_i = \mathcal{C}\boldsymbol{\mathcal{Z}}'_{i-1} + \mu\mathcal{G}\boldsymbol{s}_i$. By squaring and taking conditional expectation over the filtration $\boldsymbol{\mathcal{F}}_{i-1}$, we have

$$\mathbb{E}[\|\boldsymbol{\mathcal{Z}}'_i\|^2_\Sigma|\boldsymbol{\mathcal{F}}_{i-1}] = \|\boldsymbol{\mathcal{Z}}'_{i-1}\|^2_{\mathcal{C}^{\mathsf{T}}\Sigma\mathcal{C}} + \mu^2\mathbb{E}[\|\boldsymbol{s}_i\|^2_{\mathcal{G}^{\mathsf{T}}\Sigma\mathcal{G}}|\boldsymbol{\mathcal{F}}_{i-1}]. \tag{4.141}$$

where Σ is any positive semi-definite matrix to be decided later. By taking expectation again, we have

$$\mathbb{E}\|\boldsymbol{\mathcal{Z}}'_i\|^2_\Sigma = \mathbb{E}\|\boldsymbol{\mathcal{Z}}'_{i-1}\|^2_{\mathcal{C}^{\mathsf{T}}\Sigma\mathcal{C}} + \mu^2\mathbb{E}\|\boldsymbol{s}_i\|^2_{\mathcal{G}^{\mathsf{T}}\Sigma\mathcal{G}}. \tag{4.142}$$

Now we analyze the gradient noise term. To do that, we introduce the network noise quantity

$$\mathcal{S} \triangleq \mathrm{diag}\{S_1, S_2, \cdots, S_K\}. \tag{4.143}$$

where S_k is defined in (4.69). Note that $\mu^2\mathbb{E}\|\boldsymbol{s}_i\|^2_{\mathcal{G}^{\mathsf{T}}\Sigma\mathcal{G}} = \mu^2\mathrm{Tr}\Big(\Sigma\mathcal{G}\mathbb{E}[\boldsymbol{s}_i\boldsymbol{s}_i^{\mathsf{T}}]\mathcal{G}^{\mathsf{T}}\Big)$. By following [1, (11.72) – (11.76)], it holds that $\mu^2\mathbb{E}\|\boldsymbol{s}_i\|^2_{\mathcal{G}^{\mathsf{T}}\Sigma\mathcal{G}}$ can be well approximated by $\mu^2\mathrm{Tr}(\Sigma\mathcal{G}\mathcal{S}\mathcal{G}^{\mathsf{T}})$. To be more precise, we have

$$\limsup_{i\to\infty} \mu^2\mathbb{E}\|\boldsymbol{s}_i\|^2_{\mathcal{G}^{\mathsf{T}}\Sigma\mathcal{G}} = \mu^2\mathrm{Tr}(\Sigma\mathcal{Y}) + \mathrm{Tr}(\Sigma)\cdot o(\mu^2), \tag{4.144}$$

where $\mathcal{Y} \triangleq \mathcal{G}\mathcal{S}\mathcal{G}^{\mathsf{T}}$ and $o(\mu^2) = O(\mu^{2+\epsilon})$ with $\epsilon > 0$. By substituting (4.144) into (4.142) and taking the limit, we have

$$\begin{aligned} \limsup_{i\to\infty} \mathbb{E}\|\boldsymbol{\mathcal{Z}}'_i\|^2_{\Sigma-\mathcal{C}^{\mathsf{T}}\Sigma\mathcal{C}} &= \mu^2\mathbb{E}\|\boldsymbol{s}_i\|^2_{\mathcal{G}^{\mathsf{T}}\Sigma\mathcal{G}} \\ &= \mu^2\mathrm{Tr}(\Sigma\mathcal{Y}) + \mathrm{Tr}(\Sigma)\cdot o(\mu^2). \end{aligned} \tag{4.145}$$

Note that from (4.64), we are interested in $\limsup_{i\to\infty}\|\widetilde{\boldsymbol{\mathcal{W}}}_i\|^2 = \mathbb{E}\|\boldsymbol{\mathcal{Z}}_i'\|_\Gamma^2$. Thus, we need

$$\Sigma - \mathcal{C}^{\mathsf{T}}\Sigma\mathcal{C} = \Gamma. \tag{4.146}$$

We now recall two block Kronecker product properties that are useful in the following derivations [1, Appendix F]:

$$\mathrm{bvec}(\mathcal{A}\mathcal{C}\mathcal{B}) = (\mathcal{B}^{\mathsf{T}} \otimes_b \mathcal{A})\mathrm{bvec}(\mathcal{C}) \tag{4.147a}$$

$$\mathrm{Tr}(\mathcal{A}\mathcal{B}) = [\mathrm{bvec}(\mathcal{B}^{\mathsf{T}})]^{\mathsf{T}}\mathrm{bvec}(\mathcal{A}) \tag{4.147b}$$

for any $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ of appropriate dimensions. To solve for Σ in (4.146), we apply property (4.147a) to both sides of (4.146) and reach

$$\mathrm{bvec}(\Sigma) - (\mathcal{C}^{\mathsf{T}} \otimes_b \mathcal{C}^{\mathsf{T}})\mathrm{bvec}(\Sigma) = \mathrm{bvec}(\Gamma), \tag{4.148}$$

where $\otimes_b$ is block Kronecker operation. Now we define $\mathcal{F} = \mathcal{C}^{\mathsf{T}} \otimes_b \mathcal{C}^{\mathsf{T}} \in \mathbb{R}^{(2K-1)^2M^2\times(2K-1)^2M^2}$. Since $\mathcal{C}$ is stable for sufficiently small step-sizes, we know $\mathcal{F}$ is also stable and hence $I-\mathcal{F}$ is invertible. Therefore, it holds that

$$\mathrm{bvec}(\Sigma) = (I-\mathcal{F})^{-1}\mathrm{bvec}(\Gamma). \tag{4.149}$$

Next we evaluate the right-hand side in (4.145). From property (4.147b), we have

$$\begin{aligned}\mu^2\mathrm{Tr}(\Sigma\mathcal{Y}) &= \mu^2[\mathrm{bvec}(\mathcal{Y}^{\mathsf{T}})]^{\mathsf{T}}\mathrm{bvec}(\Sigma) \\ &\overset{(4.149)}{=} \mu^2[\mathrm{bvec}(\mathcal{Y}^{\mathsf{T}})]^{\mathsf{T}}(I-\mathcal{F})^{-1}\mathrm{bvec}(\Gamma).\end{aligned} \tag{4.150}$$

To examine the above quantity, we have to evaluate $(I-\mathcal{F})^{-1}$ first. We recall from (4.139) that

$$\mathcal{C}^{\mathsf{T}} = \begin{bmatrix} I_M - \frac{\mu}{K}\sum_{k=1}^K H_k & -\frac{\mu}{c}\mathcal{R}_1^{\mathsf{T}}\mathcal{T}^{\mathsf{T}}\mathcal{X}_L^{\mathsf{T}}\mathcal{D}_1 \\ -\frac{c\mu}{K}\mathcal{X}_{R,u}^{\mathsf{T}}\mathcal{H}\mathcal{I} & \mathcal{D}_1 - \mu\mathcal{X}_R^{\mathsf{T}}\mathcal{T}^{\mathsf{T}}\mathcal{X}_L^{\mathsf{T}}\mathcal{D}_1 \end{bmatrix}. \tag{4.151}$$

With definition $\mathcal{F} = \mathcal{C}^{\mathsf{T}} \otimes_b \mathcal{C}^{\mathsf{T}}$, we partition $\mathcal{F}$ into four blocks

$$\mathcal{F} = \begin{bmatrix} \mathcal{F}_{11} & \mathcal{F}_{12} \\ \mathcal{F}_{21} & \mathcal{F}_{22} \end{bmatrix} \tag{4.152}$$

where

$$\mathcal{F}_{11} = \left(I_M - \frac{\mu}{K}\sum_{k=1}^{K} H_k\right) \otimes \left(I_M - \frac{\mu}{K}\sum_{k=1}^{K} H_k\right) \tag{4.153}$$

It can be verified that

$$\begin{aligned}
&(I-\mathcal{F})^{-1} \\
&= \begin{bmatrix} (I_M \otimes \frac{\mu}{K}\sum_{k=1}^{K} H_k + \frac{\mu}{K}\sum_{k=1}^{K} H_k \otimes I_M)^{-1} & 0 \\ 0 & 0 \end{bmatrix} + O(1) \\
&= \begin{bmatrix} I_{M^2} \\ 0 \end{bmatrix} Z^{-1} \begin{bmatrix} I_{M^2} & 0 \end{bmatrix} + O(1)
\end{aligned} \tag{4.154}$$

where $Z = \sum_{k=1}^{K} \frac{\mu}{K}\left(I_M \otimes H_k + H_k \otimes I_M\right)$. With (4.154), we have

$$\begin{aligned}
&(I-\mathcal{F})^{-1}\mathrm{bvec}(\Gamma) \\
&= \begin{bmatrix} I_{M^2} \\ 0 \end{bmatrix} Z^{-1} \begin{bmatrix} I_{M^2} & 0 \end{bmatrix} \mathrm{bvec}(\Gamma) + O(1).
\end{aligned} \tag{4.155}$$

By substituting

$$\begin{aligned}
&\begin{bmatrix} I_{M^2} & 0 \end{bmatrix} \mathrm{bvec}(\Gamma) \\
&= \left(\begin{bmatrix} I_M & 0 \end{bmatrix} \otimes_b \begin{bmatrix} I_M & 0 \end{bmatrix}\right) \mathrm{bvec}(\Gamma) \\
&= \mathrm{bvec}\left(\begin{bmatrix} I_M & 0 \end{bmatrix} \Gamma \begin{bmatrix} I_M \\ 0 \end{bmatrix}\right) \\
&= K\mathrm{bvec}(I_M) = K\mathrm{vec}(I_M)
\end{aligned} \tag{4.156}$$

into (4.155), we have

$$(I-\mathcal{F})^{-1}\mathrm{bvec}(\Gamma) = K \begin{bmatrix} I_{M^2} \\ 0 \end{bmatrix} Z^{-1}\mathrm{vec}(I_M) + O(1). \tag{4.157}$$

Next we let

$$P \triangleq \mathrm{unvec}\left(Z^{-1}\mathrm{vec}(I_M)\right) = \frac{1}{2}\left(\frac{\mu}{K}\sum_{k=1}^{K} H_k\right)^{-1}. \tag{4.158}$$

where the last equality can be verified by following similar arguments to [1, Equations (11.123)–(11.129)]. Substituting (4.158) into (4.157), we have

$$(I-\mathcal{F})^{-1}\mathrm{bvec}(\Gamma) = K \begin{bmatrix} I_{M^2} \\ 0 \end{bmatrix} \mathrm{b}vec(P) + O(1). \tag{4.159}$$

Substituting (4.159) into (4.150), we have

$$\mu^2\mathrm{Tr}(\Sigma\mathcal{Y}) = \underbrace{\mu^2 K[\mathrm{bvec}(\mathcal{Y}^{\mathsf{T}})]^{\mathsf{T}} \begin{bmatrix} I_{M^2} \\ 0 \end{bmatrix} \mathrm{bvec}(P)}_{\triangleq a} + O(\mu^2) \tag{4.160}$$

To examine $\mu^2\mathrm{Tr}(\Sigma\mathcal{Y})$ in the previous expression, we need to evaluate $\mathcal{Y}$. Since $\mathcal{Y} = \mathcal{G}\mathcal{S}\mathcal{G}^{\mathsf{T}}$, we have

$$\begin{aligned} \mathcal{Y} &= \begin{bmatrix} \frac{1}{K}\mathcal{I}^{\mathsf{T}} \\ \frac{1}{c}\mathcal{X}_L\mathcal{B}_\ell \end{bmatrix} \mathcal{S} \begin{bmatrix} \frac{1}{K}\mathcal{I} & \frac{1}{c}\mathcal{B}_\ell^{\mathsf{T}}\mathcal{X}_L^{\mathsf{T}} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{K^2}\mathcal{I}^{\mathsf{T}}\mathcal{S}\mathcal{I} & \frac{1}{K}\mathcal{I}^{\mathsf{T}}\mathcal{S}\mathcal{B}_\ell^{\mathsf{T}}\mathcal{X}_L^{\mathsf{T}} \\ \frac{1}{K}\mathcal{X}_L\mathcal{B}_\ell\mathcal{S}\mathcal{I} & \frac{1}{c^2}\mathcal{X}_L\mathcal{B}_\ell\mathcal{S}\mathcal{B}_\ell^{\mathsf{T}}\mathcal{X}_L^{\mathsf{T}} \end{bmatrix}. \end{aligned} \tag{4.161}$$

Note that from (4.143), we have $\mathcal{I}^{\mathsf{T}}\mathcal{S}\mathcal{I} = \sum_{k=1}^{K} S_k$. With the expression of $\mathcal{Y}$ in (4.161), we have

$$\begin{aligned} a &= \mu^2 K\mathrm{Tr}\left[\mathrm{unbvec}\left\{\begin{bmatrix} I_{M^2} \\ 0 \end{bmatrix} \mathrm{bvec}(P)\right\}\mathcal{Y}\right] \\ &\overset{(a)}{=} \mu^2 K\mathrm{Tr}[\begin{bmatrix} I_M \\ 0 \end{bmatrix} P \begin{bmatrix} I_M & 0 \end{bmatrix}\mathcal{Y}] \\ &= \frac{\mu}{2}\mathrm{Tr}\left\{\left(\sum_{k=1}^{K} H_k\right)^{-1}\left(\sum_{k=1}^{K} S_k\right)\right\}. \end{aligned} \tag{4.162}$$

where step (a) follows from property (4.147a) and in the last step we used (4.158) and (4.161). With the same technique as above, we can also derive that

$$\mathrm{Tr}(\Sigma)\cdot o(\mu^2) = o(\mu). \tag{4.163}$$

Substituting (4.160)–(4.163) into (4.145), we have

$$\limsup_{i\to\infty}\mathbb{E}\|\boldsymbol{\mathcal{Z}}_i'\|_{\Gamma}^2=\frac{\mu}{2}\mathrm{Tr}\left\{\left(\sum_{k=1}^{K}H_k\right)^{-1}\left(\sum_{k=1}^{K}S_k\right)\right\}+o(\mu). \tag{4.164}$$

With relation (4.66) in Lemma 4.7, we also have

$$\begin{aligned}&\limsup_{i\to\infty}\mathbb{E}\|\boldsymbol{\mathcal{Z}}_i\|_{\Gamma}^2\\&\quad=\frac{\mu}{2}\mathrm{Tr}\left\{\left(\sum_{k=1}^{K}H_k\right)^{-1}\left(\sum_{k=1}^{K}S_k\right)\right\}+o(\mu).\end{aligned} \tag{4.165}$$

Recalling the facts that $\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_i\|^2=\sum_{k=1}^{K}\mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$ and $\lim_{\mu\to 0}o(\mu)/\mu=0$, we therefore derive the MSD expression of exact diffusion as follows

$$\begin{aligned}\mathrm{M}SD&=\mu\left(\lim_{\mu\to 0}\limsup_{i\to\infty}\frac{1}{\mu K}\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_i\|^2\right)\\&\overset{(4.64)}{=}\mu\left(\lim_{\mu\to 0}\limsup_{i\to\infty}\frac{1}{\mu K}\mathbb{E}\|\boldsymbol{\mathcal{Z}}_i\|_{\Gamma}^2\right)\\&=\frac{\mu}{2K}\mathrm{Tr}\left\{\left(\sum_{k=1}^{K}H_k\right)^{-1}\left(\sum_{k=1}^{K}S_k\right)\right\}.\end{aligned} \tag{4.166}$$

CHAPTER 5

Exact Diffusion for Distributed Empirical Learning

5.1 Context and Background

This chapter considers empirical risk minimization under the decentralized network setting. For most traditional machine learning tasks, the training data are usually stored at a single computing unit [148–151]. This unit can access the entire data set and can carry out training procedures in a centralized fashion. However, to enhance performance and accelerate convergence speed, there have also been extensive studies on replacing this centralized mode of operation by distributed mechanisms [54,63,152–154]. In these schemes, the data may either be artificially distributed onto a collection of computing nodes (also known as *workers*), or it may already be physically collected by dispersed nodes or devices. These nodes can be smart phones or tablets, wireless sensors, wearables, drones, robots or self-driving automobiles. Each node is usually assigned a local computation task and the objective is to enable the nodes to converge towards the global minimizer of a central learning model. Nevertheless, in most of these distributed implementations, there continues to exist a central node, referred to as the *master*, whose purpose is to regularly collect intermediate iterates from the local workers, conduct global update operations, and distribute the updated information back to all workers.

Clearly, this mode of operation is not fully decentralized because it involves coordination with a central node. Such architectures are not ideal for on-device intelligence settings [63,155] for various reasons. First, the transmission of local information to the central node, and back from the central node to the dispersed devices, can be expensive especially when communication is conducted via multi-hop relays or when the devices are moving and the

network topology is changing. Second, there are privacy and secrecy considerations where individual nodes may be reluctant to share information with remote centers. Third, there is a critical point of failure in centralized architectures: when the central node fails, the operation comes to a halt. Moreover, the master/worker structure requires each node to complete its local computation before aggregating them at the master node, and the efficiency of the algorithms will therefore be dependent on the slowest worker.

Motivated by these considerations, in this chapter we develop a fully decentralized solution for multi-agent network situations where nodes process the data locally and are allowed to communicate only with their immediate *neighbors*. We shall assume that the dispersed nodes are connected through a network topology and that information exchanges are only allowed among neighboring devices. By "neighbors" we mean nodes that can communicate directly to each other as allowed by the graph topology. For example, in wireless sensor networks, neighboring nodes can be devices that are within the range of radio broadcasting. Likewise, in smart phone networks, the neighbors can be devices that are within the same local area network. In the proposed algorithm, there will be no need for a central or master unit and the objective is to enable each dispersed node to learn *exactly* the global model despite their limited localized interactions.

5.1.1 Problem Formulation

In a connected network with K nodes, if node k stores local data samples $\{x_{k,n}\}_{n=1}^{N_k}$, where N_k is the size of the local samples, then the data stored by the entire network is:

$$\{x_n\}_{n=1}^{N} \triangleq \Big\{\{x_{1,n}\}_{n=1}^{N_1}, \{x_{2,n}\}_{n=1}^{N_2}, \cdots, \{x_{K,n}\}_{n=1}^{N_K}\Big\}, \tag{5.1}$$

where $N = \sum_{k=1}^{K} N_k$. We consider minimizing an empirical risk function, $J(w)$, which is defined as the sample average of loss values over *all* observed data samples in the network:

$$\begin{aligned} w^\star \;\triangleq\; \arg\min_{w\in M} \;\; J(w) \;&\triangleq\; \frac{1}{N}\sum_{n=1}^{N} Q(w; x_n) \\ &= \frac{1}{N}\sum_{k=1}^{K}\sum_{n=1}^{N_k} Q(w; x_{k,n}). \end{aligned} \tag{5.2}$$

Here, the notation $Q(w;x_n)$ denotes the loss value evaluated at w and the n-th sample, x_n. We also introduce the local empirical risk function, $J_k(w)$, which is defined as the sample average of loss values over the *local* data samples stored at node k, i.e., over $\{x_{k,n}\}_{n=1}^{N_k}$:

$$J_k(w) \triangleq \frac{1}{N_k}\sum_{n=1}^{N_k} Q(w;x_{k,n}). \tag{5.3}$$

Using the local empirical risk functions, $\{J_k(w)\}$, it can be verified that the original global optimization problem (5.2) can be reformulated as the equivalent problem of minimizing the weighted aggregation of K local empirical risk functions:

$$w^\star \triangleq \arg\min_{w\in M} \; J(w) \triangleq \sum_{k=1}^{K} q_k J_k(w). \tag{5.4}$$

where $q_k \triangleq N_k/N$. The following assumptions are standard in the distributed optimization literature, and they are automatically satisfied by many loss functions of interest in the machine learning literature (such as quadratic losses, logistic losses — see, e.g., [1, 4]). For simplicity in this article, we assume the loss functions are smooth, although the arguments can be extended to deal with non-smooth losses, as we have done in [156, 157].

Assumption 5.1 *The loss function, $Q(w;x_n)$, is convex, twice-differentiable, and has a δ-Lipschitz continuous gradient, i.e., for any $w_1, w_2 \in \mathbb{R}^M$ and $1 \le n \le N$:*

$$\|\nabla_w Q(w_1;x_n) - \nabla_w Q(w_2;x_n)\| \le \delta\|w_1 - w_2\| \tag{5.5}$$

where $\delta > 0$. Moreover, there exists at least one loss function $Q(w;x_{n_o})$ that is strongly convex, i.e.,

$$\nabla_w^2 Q(w;x_{n_o}) \ge \nu I_M > 0, \;\; \text{for some } n_o. \tag{5.6}$$

■

5.1.2 Related Work

There exists an extensive body of research on solving optimization problems of the form (5.4) in a fully decentralized manner. Some recent works include techniques such as ADMM [74,85], DLM [87], EXTRA [75], ESOM [158], DIGing [93], Aug-DGM [95] and exact diffusion [15, 16]. These methods provide linear convergence rates and are proven to converge to the *exact* minimizer, $w^\star$. The exact diffusion method, in particular, has been shown to have a

wider stability range than EXTRA implementations (i.e., it is stable for a wider range of step-sizes, μ), and is also more efficient in terms of communications than DIGing. However, all these methods require the evaluation of the true gradient vector of each $J_k(w)$ at each iteration. It is seen from the definition (5.3), and depending on the size N_k, that this computation can be prohibitive for large-data scenarios.

One can resort to replacing the true gradient by a stochastic gradient approximation, as is commonplace in traditional diffusion or consensus algorithms [1, 4–6, 8, 9, 12, 142]. In these implementations, each node k approximates the true gradient vector $\nabla J_k(w)$ by using one random sample gradient, $\nabla Q(w; x_{k,\boldsymbol{n}})$, where $\boldsymbol{n} \in \{1, 2, \cdots, N_k\}$ is a uniformly-distributed random index number. While this mode of operation is efficient, it has been proven to converge linearly only to a small $O(\mu)-$neighborhood around the exact solution $w^\star$ [36] where μ is the constant step-size. If convergence to the exact solution is desired, then one can employ decaying step-sizes instead of constant step-sizes; in this case, however, the convergence rate will be slowed down appreciably. An alternative is to employ variance-reduced techniques to enable convergence to the exact minimizer while employing a stochastic gradient approximation. One proposal along these lines is the DSA method [77], which is based on the variance-reduced SAGA method [149, 151]. However, similar to SAGA, the DSA method suffers from the same huge memory requirement since each node k will need to store an estimate for each possible gradient $\{\nabla Q(w; x_{k,n})\}_{n=1}^{N_k}$. This requirement is a burden when N_k is large, as happens in applications involving large data sets.

5.1.3 Contribution

This chapter derives a new fully-decentralized variance-reduced stochastic-gradient algorithm with linear convergence guarantees and with significantly reduced memory requirements. We refer to the technique as the diffusion-AVRG method (where AVRG stands for the "amortized variance-reduced gradient" technique proposed in the related work [76] for single-agent learning). Unlike DSA and SAGA, this method does not require extra memory to store gradient estimates. The method is also different from the well-known alternative to SAGA known as SVRG [150, 159]. The SVRG method has an inner loop to perform stochastic

variance-reduced gradient descent and an outer loop to calculate the true local gradient. These two loops introduce *imbalances* into the gradient calculation and complicate decentralized implementations. In comparison, the AVRG construction involves balanced gradient calculations and is amenable to fully distributed solutions, especially when the size of the data is unevenly distributed across the nodes. More comparisons between diffusion-AVRG and diffusion-SVRG are discussed in Section 5.4.1. This paper also proposes to use the mini-batch technique to save communications in diffusion-AVRG.

Notation Throughout this paper we use $\mathrm{diag}\{x_1, \cdots, x_N\}$ to denote a diagonal matrix consisting of diagonal entries $x_1, \cdots, x_N$, and use $\mathrm{col}\{x_1, \cdots, x_N\}$ to denote a column vector formed by stacking $x_1, \cdots, x_N$. For symmetric matrices X and Y, the notation $X \leq Y$ or $Y \geq X$ denotes $Y - X$ is positive semi-definite. For a vector x, the notation $x \succeq 0$ denotes that each element of x is non-negative. For a matrix X, we let $\|X\|$ denote its 2-induced norm (maximum singular value), and $\lambda(X)$ denote its eigenvalues. The notation $\mathbb{1}_K = \mathrm{col}\{1, \cdots, 1\} \in \mathbb{R}^K$, and $0_K = \mathrm{col}\{0, \cdots, 0\} \in \mathbb{R}^K$. For a nonnegative diagonal matrix $\Lambda = \mathrm{diag}\{\lambda_1, \cdots, \lambda_K\}$, we let $\Lambda^{1/2} = \mathrm{diag}\{\lambda_1^{1/2}, \cdots, \lambda_K^{1/2}\}$.

5.2 Two Key Components

In this section we review two useful techniques that will be blended together to yield the diffusion-AVRG scheme. The first technique is the exact diffusion algorithm from [15, 16], which is able to converge to the *exact* minimizers of the decentralized optimization problem (5.4). The second technique is the amortized variance-reduced (AVRG) algorithm proposed in our earlier work [76, 160], which has balanced computations per iteration and was shown there to converge linearly under random reshuffling. Neither of the methods alone is sufficient to solve the multi-agent optimization problem (5.4) in a decentralized and efficient manner. This is because exact diffusion is decentralized but not efficient for the current problem, while AVRG is efficient but not decentralized.

Algorithm 5.1 (Exact diffusion strategy for each node k)

Let $\overline{A} = (I_N + A)/2$ and $\overline{a}_{\ell k} = [\,\overline{A}\,]_{\ell k}$. Initialize $w_{k,0}$ arbitrarily, and let $\psi_{k,0} = w_{k,0}$.

Repeat iteration $i = 1, 2, 3 \cdots$

$$\psi_{k,i+1} = w_{k,i} - \mu\, q_k \nabla J_k(w_{k,i}), \qquad \text{(adaptation)} \tag{5.8}$$

$$\phi_{k,i+1} = \psi_{k,i+1} + w_{k,i} - \psi_{k,i}, \qquad \text{(correction)} \tag{5.9}$$

$$w_{k,i+1} = \sum_{\ell \in \mathcal{N}_k} \overline{a}_{\ell k} \phi_{\ell,i+1}. \qquad \text{(combination)} \tag{5.10}$$

End

5.2.1 Exact Diffusion Algorithm

Thus, consider again the aggregate optimization problem (5.4) over a strongly-connected network with K nodes, where the $\{q_k\}$ are positive scalars. Each local risk $J_k(w)$ is a differentiable and convex cost function, and the global risk $J(w)$ is strongly convex. To implement the exact diffusion algorithm, we need to associate a combination matrix $A = [a_{\ell k}]_{\ell,k=1}^{K}$ with the network graph, where a positive weight $a_{\ell k}$ is used to scale data that flows from node ℓ to k if both nodes happen to be neighbors; if nodes ℓ and k are not neighbors, then we set $a_{\ell k} = 0$. In this paper we assume A is symmetric and doubly stochastic, i.e.,

$$a_{\ell k} = a_{k\ell}, \;\; A = A^{\mathsf{T}} \text{ and } \; A\mathbb{1}_K = \mathbb{1}_K \tag{5.7}$$

where $\mathbb{1}$ is a vector with all unit entries. Such combination matrices can be easily generated in a decentralized manner through the Laplacian rule, maximum-degree rule, Metropolis rule or other rules (see, e.g., Table 14.1 in [1]). We further introduce μ as the step-size parameter for all nodes, and let $\mathcal{N}_k$ denote the set of neighbors of node k (including node k itself).

The exact diffusion algorithm [15] is listed in (5.8)–(5.10). The subscript k refers to the node while the subscript i refers to the iteration. It is observed that there is no central node that performs global updates. Each node performs a local update (see equation (5.8)) and then combines its iterate with information collected from the neighbors (see equation (5.10)).

The correction step (5.9) is necessary to guarantee exact convergence. Indeed, it is proved in [16] that the local variables $w_{k,i}$ converge to the exact minimizer of problem (5.4), $w^\star$, at a linear convergence rate under relatively mild conditions. However, note from (5.3) that it is expensive to calculate the gradient $\nabla J_k(w)$ in step (5.8), especially when N_k is large. In the proposed algorithm derived later, we will replace the true gradient $\nabla J_k(w)$ in (5.8) by an amortized variance-reduced gradient, denoted by $\widehat{\nabla J_k}(\boldsymbol{w}_{k,i-1})$.

5.2.2 Amortized Variance-Reduced Gradient (AVRG) Algorithm

The AVRG construction [76] is a centralized solution to optimization problem (5.2). It belongs to the class of variance-reduced methods. There are mainly two families of variance-reduced stochastic algorithms to solve problems like (5.2): SVRG [150, 159] and SAGA [149, 151]. The SVRG solution employs two loops — the true gradient is calculated in the outer loop and the variance-reduced stochastic gradient descent is performed within the inner loop. For this method, one disadvantage is that the inner loop can start only after the calculation of the true gradient is completed in the outer loop. This leads to an *unbalanced* gradient calculation. For large data sets, the calculation of the true gradient can be time-consuming leading to significant idle time, which is not well-suited for decentralized solutions. More details are provided later in Sec. 5.4. In comparison, the SAGA solution has a single loop. However, it requires significant storage to estimate the true gradient, which is again prohibitive for effective decentralization on nodes or devices with limited memory.

These observations are the key drivers behind the introduction of the amortized variance-reduced gradient (AVRG) algorithm in [76]: it avoids the disadvantages of both SVRG and SAGA for decentralization, and has been shown to converge at a linear rate to the true minimizer. AVRG is based on the idea of removing the outer loop from SVRG and amortizing the calculation of the true gradient within the inner loop evenly. To guarantee convergence, random reshuffling is employed in each epoch. Under random reshuffling, the algorithm is run multiple times over the data where each run is indexed by t and is referred to as an epoch. For each epoch t, a uniform random permutation function $\boldsymbol{\sigma}^t$ is generated and data are

Algorithm 5.2 (AVRG strategy)

Initialize $\boldsymbol{w}_0^0$ arbitrarily; let $\boldsymbol{g}^0 = 0$, $\nabla Q(\boldsymbol{w}_0^0; x_n) \leftarrow 0$ for $n \in \{1, 2, \cdots, N\}$.

Repeat epoch $t = 0, 1, 2, \cdots$:

generate a random permutation function $\boldsymbol{\sigma}^t$ and set $\boldsymbol{g}^{t+1} = 0$;

Repeat iteration $i = 0, 1, \cdots, N-1$:

$$\boldsymbol{n}_i^t = \boldsymbol{\sigma}^t(i+1) \tag{5.11}$$

$$\boldsymbol{w}_{i+1}^t = \boldsymbol{w}_i^t - \mu\Big(\nabla Q(\boldsymbol{w}_i^t; x_{\boldsymbol{n}_i^t}) - \nabla Q(\boldsymbol{w}_0^t; x_{\boldsymbol{n}_i^t}) + \boldsymbol{g}^t\Big) \tag{5.12}$$

$$\boldsymbol{g}^{t+1} \leftarrow \boldsymbol{g}^{t+1} + \frac{1}{N}\nabla Q(\boldsymbol{w}_i^t; x_{\boldsymbol{n}_i}) \tag{5.13}$$

End

set $\boldsymbol{w}_0^{t+1} = \boldsymbol{w}_N^t$;

End

sampled according to it. AVRG is listed in Algorithm 2, which has balanced computation costs per iteration with the calculation of two gradients $\nabla Q(\boldsymbol{w}_i^t; x_{\boldsymbol{n}_i})$ and $\nabla Q(\boldsymbol{w}_0^t; x_{\boldsymbol{n}_i})$. Different from SVRG and SAGA, the stochastic gradient estimate $\widehat{\nabla J}(\boldsymbol{w}_i^t) = \nabla Q(\boldsymbol{w}_i^t; x_{\boldsymbol{n}_i}) - \nabla Q(\boldsymbol{w}_0^t; x_{\boldsymbol{n}_i}) + \boldsymbol{g}^t$ is biased. However, it is explained in [76] that $\mathbb{E}\|\widehat{\nabla J}(\boldsymbol{w}_i^t) - \nabla J(\boldsymbol{w}_i^t)\|^2$ will approach 0 as epoch t tends to infinity, which implies that AVRG is an asymptotic unbiased variance-reduced method.

5.3 Diffusion–AVRG Algorithm for Balanced Data Distributions

We now design a fully-decentralized algorithm to solve (5.4) by combining the exact diffusion strategy (5.8)–(5.10) and the AVRG mechanism (5.11)–(5.13). We consider first the case in which all nodes store the same amount of local data, i.e., $N_1 = \cdots = N_K = \bar{N} = N/K$. For this case, the cost function weights $\{q_k\}$ in problem (5.4) are equal, $q_1 = \cdots = q_K = 1/K$, and it makes no difference whether we keep these scaling weights or remove them

from the aggregate cost. The proposed diffusion-AVRG algorithm to solve (5.4) is listed in Algorithm 3 under Eqs. (5.14)–(5.19). Since each node has the same amount of local data samples, Algorithm 3 can be described in a convenient format involving epochs t and an inner iterations index i within each epoch. For each epoch or run t over the data, the original data is randomly reshuffled so that the sample of index $i+1$ at agent k becomes the sample of index $\boldsymbol{n}_{k,i}^t = \boldsymbol{\sigma}_k^t(i+1)$ in that run. Subsequently, at each inner iteration i, each node k will first generate an amortized variance-reduced gradient $\widehat{\nabla J}_k(\boldsymbol{w}_{k,i}^t)$ via (5.14)–(5.16), and then apply it into exact diffusion (5.17)–(5.19) to update $\boldsymbol{w}_{k,i+1}^t$. Here, the notation $\boldsymbol{w}_{k,i}^t$ represents the estimate that agent k has for $w^\star$ at iteration i within epoch t. With each node combining information from neighbors, there is no central node in this algorithm. Moreover, unlike DSA [77], this algorithm does not require extra memory to store gradient estimates. The linear convergence of diffusion-AVRG is established in the following theorem.

Theorem 5.1 (Linear Convergence) *Under Assumption 5.1, if the step-size μ satisfies*

$$\mu \leq C\left(\frac{\nu(1-\lambda)}{\delta^2 \bar{N}}\right), \tag{5.20}$$

then, for any $k \in \{1, 2, \cdots, K\}$, it holds that

$$\mathbb{E}\|\boldsymbol{w}_{k,0}^{t+1} - w^\star\|^2 \leq D\rho^t, \tag{5.21}$$

where

$$\rho = \frac{1 - \frac{\overline{N}}{8} a\mu\nu}{1 - 8b\mu^3\delta^4\bar{N}^3/\nu} < 1. \tag{5.22}$$

The constants C, D, a, b are positive constants independent of $\bar{N}$, ν and δ; they are defined in the appendices. The constant $\lambda = \lambda_2(A) < 1$ is the second largest eigenvalue of the combination matrix A.

Proof: The derivation of this result is lengthy and is given in Appendix 5.A. The proof is by no means trivial for various reasons. One source of complication is the decentralized nature of the algorithm with nodes only allowed to interact locally. Moreover, due to the bias in the gradient estimate, current analyses used for SVRG [150], SAGA [151], or DSA [77] are not suitable; these analyses can only deal with uniform sampling and unbiased gradient constructions. In our setting, the gradient constructions are biased and sampling is random with reshuffling (rather than uniform). The detailed proof is given in the appendix. ■

Algorithm 5.3 (diffusion-AVRG at node k for balanced data)

Initialize $\boldsymbol{w}_{k,0}^0$ arbitrarily; let $\boldsymbol{\psi}_{k,0}^0 = \boldsymbol{w}_{k,0}^0$, $\boldsymbol{g}_k^0 = 0$, and $\nabla Q(\boldsymbol{w}_0^0; x_{k,n}) \leftarrow 0,\ \ 1 \le n \le \bar{N}$, where $\bar{N} = N/K$.

Repeat epoch $t = 0, 1, 2, \cdots$

generate a random permutation function $\boldsymbol{\sigma}_k^t$ and set $\boldsymbol{g}_k^{t+1} = 0$.

Repeat iteration $i = 0, 1, \cdots, \bar{N} - 1$:

$$\boldsymbol{n}_{k,i}^t = \boldsymbol{\sigma}_k^t(i+1), \tag{5.14}$$

$$\widehat{\nabla J}_k(\boldsymbol{w}_{k,i}^t) = \nabla Q(\boldsymbol{w}_{k,i}^t; x_{k,\boldsymbol{n}_{k,i}^t}) - \nabla Q(\boldsymbol{w}_{k,0}^t; x_{k,\boldsymbol{n}_{k,i}^t}) + \boldsymbol{g}_k^t, \tag{5.15}$$

$$\boldsymbol{g}_k^{t+1} \leftarrow \boldsymbol{g}_k^{t+1} + \frac{1}{\bar{N}} \nabla Q(\boldsymbol{w}_{k,i}^t; x_{k,\boldsymbol{n}_{k,i}^t}), \tag{5.16}$$

update $\boldsymbol{w}_{k,i+1}^t$ with exact diffusion:

$$\boldsymbol{\psi}_{k,i+1}^t = \boldsymbol{w}_{k,i}^t - \mu \widehat{\nabla J}_k(\boldsymbol{w}_{k,i}^t), \tag{5.17}$$

$$\boldsymbol{\phi}_{k,i+1}^t = \boldsymbol{\psi}_{k,i+1}^t + \boldsymbol{w}_{k,i}^t - \boldsymbol{\psi}_{k,i}^t, \tag{5.18}$$

$$\boldsymbol{w}_{k,i+1}^t = \sum_{\ell \in \mathcal{N}_k} \overline{a}_{\ell k} \boldsymbol{\phi}_{\ell,i+1}^t. \tag{5.19}$$

End

set $\boldsymbol{w}_{k,0}^{t+1} = \boldsymbol{w}_{k,\bar{N}}^t$ and $\boldsymbol{\psi}_{k,0}^{t+1} = \boldsymbol{\psi}_{k,\bar{N}}^t$

End

5.4 Diffusion–AVRG Algorithm for Unbalanced Data Distributions

When the size of the data collected at the nodes may vary drastically, some challenges arise. For example, assume we select $\widehat{N} = \max_k\{N_k\}$ as the epoch size for all nodes. When node k with a smaller N_k finishes its epoch, it will have to stop and wait for the other nodes to finish their epochs. Such an implementation is inefficient because nodes will be idle while they could be assisting in improving the convergence performance.

We instead assume that nodes will continue updating without any idle time. If a particular node k finishes running over all its data samples during an epoch, it will then continue

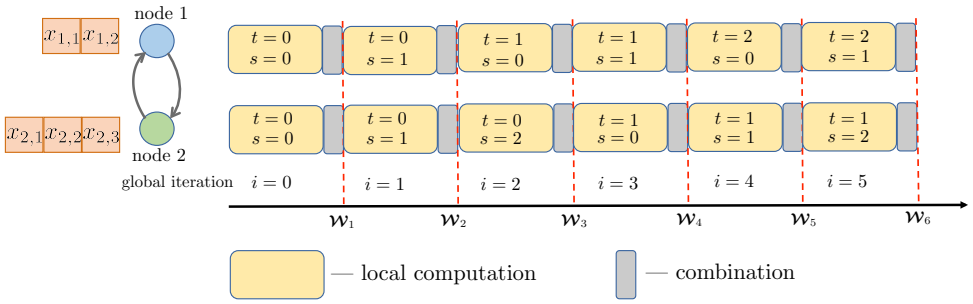

Figure 5.1: Illustration of the operation of diffusion-AVRG for a two-node network.

its next epoch right away. In this way, there is no need to introduce a uniform epoch. We list the method in Algorithm 4; this listing includes the case of balanced data as a special case. In other words, we have a single diffusion-AVRG algorithm. We are describing it in two formats (Algorithms 3 and 4) for ease of exposition so that readers can appreciate the simplifications that occur in the balanced data case.

In Algorithm 4, at each iteration i, each node k will update its $\boldsymbol{w}_{k,i}$ to $\boldsymbol{w}_{k,i+1}$ by exact diffusion (5.26)–(5.28) with stochastic gradient. Notice that q_k has to be used to scale the step-size in (5.26) because of the spatially unbalanced data distribution. To generate the local stochastic gradient $\widehat{\nabla J_k}(\boldsymbol{w}_{k,i})$, node k will transform the *global* iteration index i to its own *local* epoch index t and *local* inner iteration s. With t and s determined, node k is able to generate $\widehat{\nabla J_k}(\boldsymbol{w}_{k,i})$ with the AVRG recursions (5.23)–(5.25). Note that $t, s, \boldsymbol{\sigma}_k^t, \boldsymbol{\theta}_{k,0}^t, \boldsymbol{n}_s^t$ are all local variables hidden in node k to help generate the local stochastic gradient $\widehat{\nabla J_k}(\boldsymbol{w}_{k,i})$ and do not appear in exact diffusion (5.26)–(5.28). Steps (5.23)–(5.27) are all local update operations within each node while step (5.28) needs communication with neighbors. It is worth noting that the local update (5.23)–(5.27) for each node k at each iteration requires the same amount of computations no matter how different the sample sizes $\{N_k\}$ are. This balanced computation feature guarantees the efficiency of diffusion-AVRG and reduces waiting time. Figure 5.1 illustrates the operation of Algorithm 4 for a two-node network with $N_1 = 2$ and $N_2 = 3$. That is, the first node collects two samples while the second node collects three samples. For each iteration index i, the nodes will determine the local values for their indices t and s. These indices are used to generate the local variance-reduced gradients $\widehat{\nabla J_k}(\boldsymbol{w}_{k,i})$. Once node k finishes its own local epoch t, it will start its next

epoch $t+1$ right away. Observe that the local computations has similar widths because each node has a balanced computation cost per iteration. Note that $\boldsymbol{\mathcal{w}}_i = [\boldsymbol{w}_{1,i}; \boldsymbol{w}_{2,i}]$ in Figure 5.1.

5.4.1 Comparison with Decentralized SVRG

AVRG is not the only variance-reduced algorithm that can be combined with exact diffusion. In fact, SVRG is another alternative to save memory compared to SAGA. SVRG has two loops of calculation: it needs to complete the calculation of the true gradient before starting the inner loop. Such two-loop structures are not suitable for decentralized setting, especially when data can be distributed unevenly. To illustrate this fact assume, for the sake of argument, that we combine exact diffusion with SVRG to obtain a diffusion-SVRG variant, which we list in Algorithm 5. Similar to diffusion-AVRG, each node k will transform the global iteration index i into a local epoch index t and a local inner iteration s, which are then used to generate $\widehat{\nabla J}(\boldsymbol{w}_{k,i})$ through SVRG. At the very beginning of each local epoch t, a true local gradient has to be calculated in advance; this step causes a pause before the update of $\boldsymbol{\phi}_{k,i+1}$. Now since the neighbors of node k will be waiting for $\boldsymbol{\phi}_{k,i+1}$ in order to update their own $\boldsymbol{w}_{\ell,i+1}$, the pause by node k will cause all its neighbors to wait. These waits reduce the efficiency of this decentralized implementation, which explains why the earlier diffusion-AVRG algorithm is preferred. Fig. 5.2 illustrates the diffusion-SVRG strategy with $N_1 = 2$ and $N_2 = 3$. Comparing Figs. 5.1 and 5.2, the balanced calculation resulting from AVRG effectively reduces idle times and enhances the efficiency of the decentralized implementation.

5.5 Diffusion-AVRG with Mini-batch Strategy

Compared to exact diffusion [15, 16], diffusion-AVRG allows each agent to sample one gradient at each iteration instead of calculating the true gradient with N_k data. This property enables diffusion-AVRG to be more computation efficient than exact diffusion. It is observed in Figs. 5.6 and 5.7 from Section 5.6 that in order to reach the same accuracy, diffusion-

AVRG needs less gradient calculation than exact diffusion.

However, such computational advantage comes with extra communication costs. In the exact diffusion method listed in Algorithm 1, it is seen that agent k will communicate after calculating its true gradient $\nabla J(w) = \frac{1}{N_k}\sum_{n=1}^{N_k} Q(w; x_{k,n})$. But in the diffusion-AVRG listed in Algorithms 2 and 3, each agent will communicate after calculating only one stochastic gradient. Intuitively, in order to reach the same accuracy, diffusion-AVRG needs more iterations than exact diffusion, which results in more communications. The communication comparison for diffusion-AVRG and exact diffusion are also shown in Figs. 5.6 and 5.7 in Section 5.6.

In this section we introduce the mini-batch strategy to balance the computation and communication of diffusion-AVRG. For simplicity, we consider the situation where all local data size N_k are equal to $\bar{N}$, but the strategy can be extended to handle the spatially unbalanced data distribution case. Let the batch size be B, and the number of batches $L \triangleq \bar{N}/B$. The local data in agent k can be partitioned as

$$\{x_{k,n}\}_{n=1}^{\bar{N}} = \left\{\{x_{k,n}^{(1)}\}_{n=1}^{B}, \{x_{k,n}^{(2)}\}_{n=1}^{B}, \cdots, \{x_{k,n}^{(L)}\}_{n=1}^{B}\right\}, \tag{5.35}$$

where the superscript (ℓ) indicates the ℓ-th mini-batch. In addition, the local cost function $J_k(w)$ can be rewritten as

$$\begin{aligned} J_k(w) &= \frac{1}{\bar{N}}\sum_{n=1}^{\bar{N}} Q(w; x_{k,n}) = \frac{B}{\bar{N}}\sum_{\ell=1}^{L}\frac{1}{B}\sum_{n=1}^{B} Q(w; x_{k,n}^{(\ell)}) \\ &= \frac{1}{L}\sum_{\ell=1}^{L} Q_k^{(\ell)}(w), \end{aligned} \tag{5.36}$$

where the last equality holds because $L = \bar{N}/B$ and

$$Q_k^{(\ell)}(w) \triangleq \frac{1}{B}\sum_{n=1}^{B} Q(w; x_{k,n}^{(\ell)}) \tag{5.37}$$

is defined as the cost function over the ℓ-th batch in agent k. Note that the mini-batch formulations (5.36) and (5.37) are the generalization of cost function (5.3). When $B = 1$, formulations (5.36) and (5.37) will reduce to (5.3). Moreover, it is easy to prove that $\{Q_k^{\ell}(w)\}_{k=1,\ell=1}^{K,L}$ satisfy Assumption 5.1.

Since the mini-batch formulations (5.36) and (5.37) fall into the form of problem (5.3)

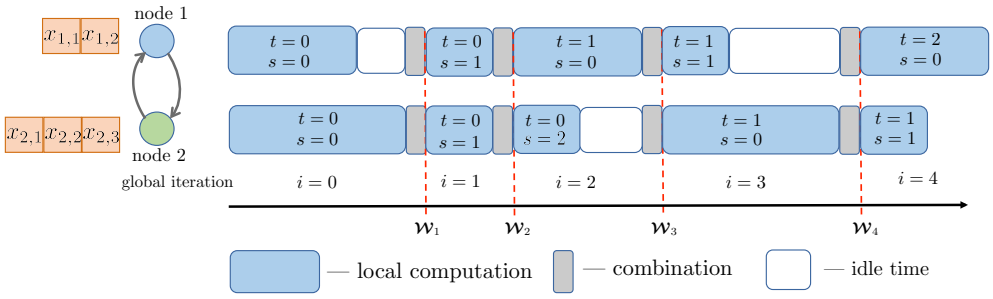

Figure 5.2: Illustration of what would go wrong if one attempts a diffusion-SVRG implementation for a two-node network, and why diffusion-AVRG is the recommended implementation.

and (5.4), we can directly extend Algorithm 3 to the mini-batch version with the convergence guarantee. The only difference is for each iteration, a batch, rather than a sample will be picked up, and then length of batches is L rather than $\bar{N}$. We also list the mini-batch algorithm in Algorithm 6.

Diffusion-AVRG with mini-batch stands in the middle point between standard diffusion-AVRG and exact diffusion. For each iteration, Algorithm 6 samples B gradients, rather than 1 gradient or $\bar{N}$ gradients, and then communicates. The size of B will determine the computation and communication efficiency, and there is a trade-off between computation and communication. When given the actual cost in real-world applications, we can determine the Pareto optimal for the batch-size. In our simulation shown in Section 5.6, when best batch-size is chosen, diffusion-AVRG with mini-batch can be much more computation efficient while maintaining almost the same communication efficiency with exact diffusion.

5.6 Simulation Results

In this section, we illustrate the convergence performance of diffusion-AVRG. We consider problem (5.4) in which $J_k(w)$ takes the form of regularized logistic regression loss function:

$$J_k(w) \triangleq \frac{1}{N_k}\sum_{n=1}^{N_k}\left(\frac{\rho}{2}\|w\|^2+\ln\left(1+\exp(-\gamma_k(n)h_{k,n}^{\mathsf{T}}w)\right)\right)$$

with $q_k = N_k/N$. The vector $h_{k,n}$ is the n-th feature vector kept by node k and $\gamma_k(n) \in \{\pm 1\}$ is the corresponding label. In all experiments, the factor ρ is set to $1/N$, and the solution $w^\star$

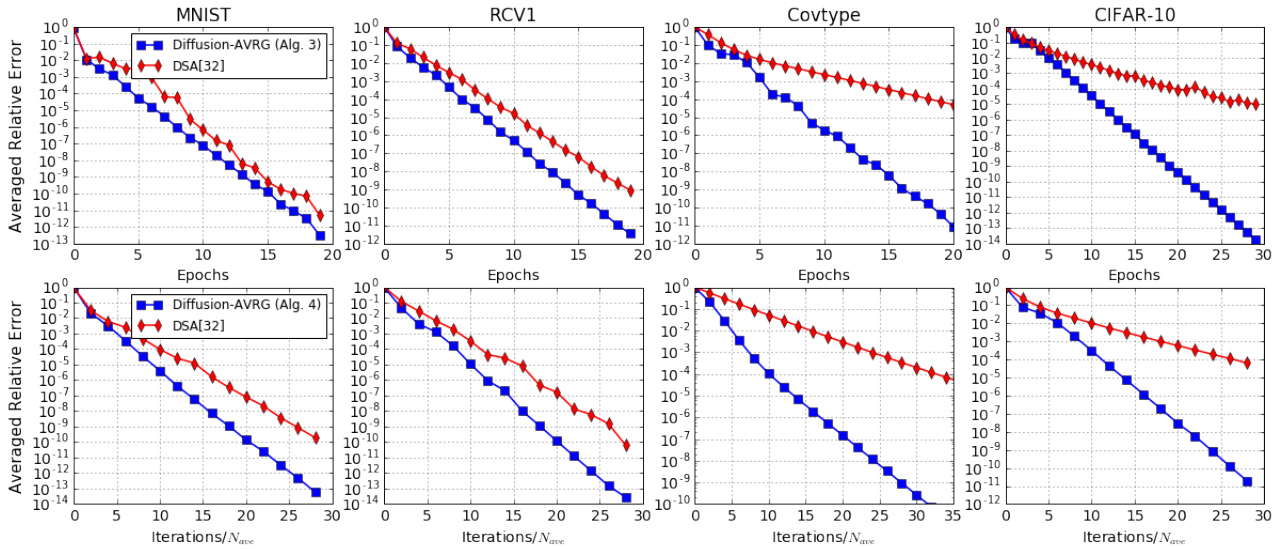

Figure 5.3: Comparison between diffusion-AVRG and DSA over various datasets. Top: data are evenly distributed over the nodes; Bottom: data are unevenly distributed over the nodes. The average sample size is $N_{\text{ave}} = \sum_{k=1}^{K} N_k/K$.

to (5.4) is computed by using the Scikit-Learn Package. All experiments are run over four datasets: covtype.binary[1], rcv1.binary[2], MNIST[3], and CIFAR-10[4]. The last two datasets have been transformed into binary classification problems by considering data with labels 2 and 4, i.e., digital two and four classes for MNIST, and cat and dog classes for CIFAR-10. All features have been preprocessed and normalized to the unit vector [159]. We also generate a randomly connected network with $K = 20$ nodes, which is shown in Fig. 5.4. The associated doubly-stochastic combination matrix A is generated by the Metropolis rule [1].

In our first experiment, we test the convergence performance of diffusion-AVRG (Algorithm 3) with even data distribution, i.e., $N_k = N/K$. We compare the proposed algorithm with DSA [77], which is based on SAGA [151] and hence has significant memory requirement. In comparison, the proposed diffusion-AVRG algorithm does not need to store the

[1] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

[2] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

[3] `http://yann.lecun.com/exdb/mnist/`

[4] `http://www.cs.toronto.edu/~kriz/cifar.html`

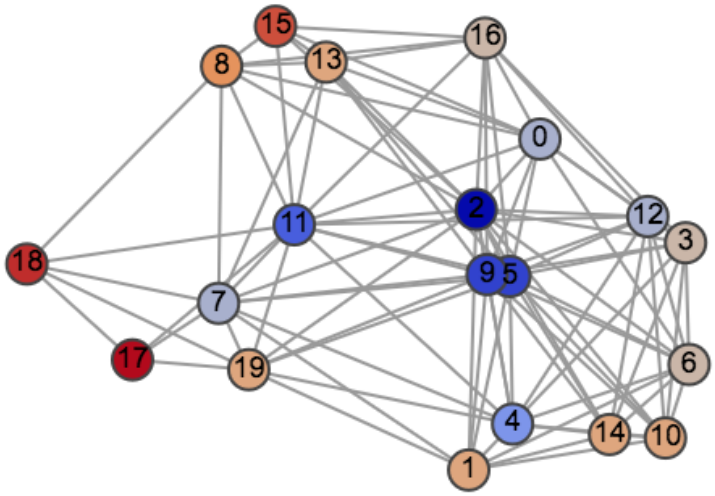

Figure 5.4: A random connected network with 20 nodes.

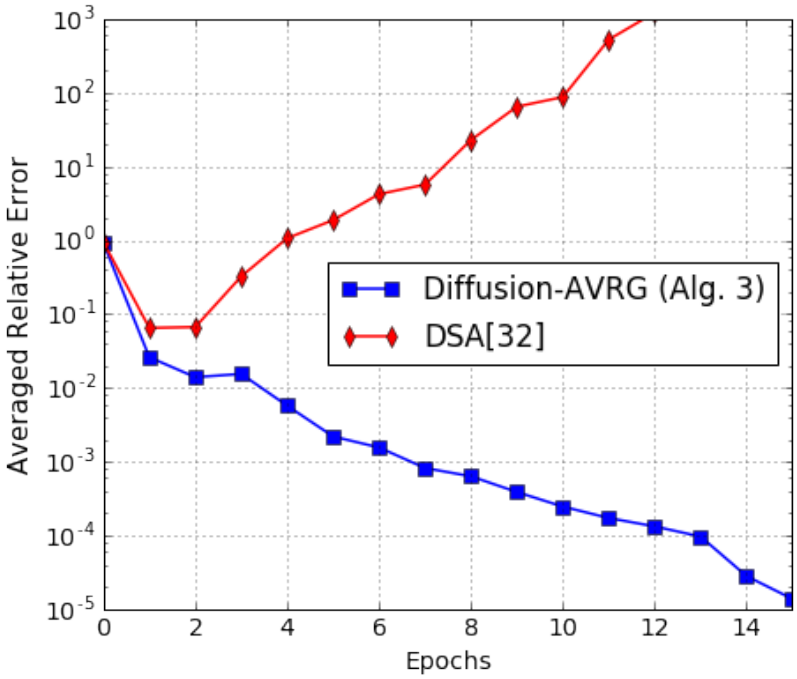

Figure 5.5: Diffusion-AVRG is more stable than DSA.

gradient estimates and is quite memory-efficient. The experimental results are shown in the top 4 plots of Fig. 5.3. To enable fair comparisons, we tune the step-size parameter of each algorithm for fastest convergence in each case. The plots are based on measuring the averaged relative square-error, $\frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{w}_{k,0}^{t}-\boldsymbol{w}^{\star}\|^2/\|\boldsymbol{w}^{\star}\|^2$. It is observed that both algorithms converge linearly to $w^{\star}$, while diffusion-AVRG converges faster (especially on Covtype and CIFAR-10).

In our second experiment, data are randomly assigned to each node, and the sample sizes at the nodes may vary drastically. We now compare diffusion-AVRG (Algorithm 3) with DSA. Since there is no epoch for this scenario, we compare the algorithms with respect to the iterations count. In the result shown in bottom 4 plots of Fig. 5.3, it is also observed that both algorithms converge linearly to $w^{\star}$, with diffusion-AVRG converging faster than DSA.

In our third experiment, we test the stability of DSA and diffusion-AVRG. For simplicity, this experiment is conducted in the context of solving a linear regression problem with synthetic data. Each feature-label pair $(\boldsymbol{h}_n, \boldsymbol{\gamma}(n))$ is drawn from a Gaussian distribution. We generate $N = 100,000$ data points, which are evenly distributed over the 20 nodes. We set the same step-size to both algorithms and check which one of them exhibits a wider step-size range for stability. For example, in Fig. 5.5, it is observed that DSA diverges while diffusion-AVRG still converges when $\mu = 0.13$. It has been observed during these experiments that diffusion-AVRG is more stable than DSA. This improved stability is inherited from the structure of the exact diffusion strategy [4, 15, 16]. The improved stability range also helps explain why diffusion-AVRG is faster than DSA in Fig. 5.3.

In our fourth experiment, we test how the mini-batch size B influences the computation and communication efficiency in diffusion-AVRG. The experiment is conducted on the MNIST and RCV1 datasets. For each batch size, we run the algorithm until the relative error reaches 10^{-10}. The step-size for each batch size is adjusted to be optimal. The communication is examined by counting the number of message passing rounds, and the computation is examined by counting the number of $\nabla Q(w; x_n)$ evaluations. The exact diffusion is also tested for comparison. In Fig. 5.6, we use "AVRG" to indicate the standard diffusion-

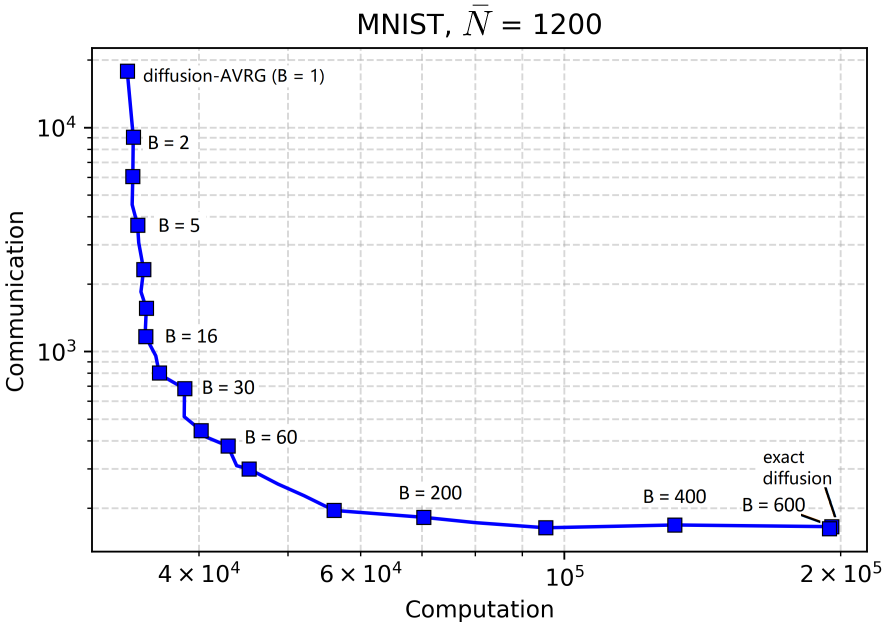

Figure 5.6: Performance of diffusion-AVRG with different batch sizes on MNIST dataset. Each agent holds $\bar{N} = 1200$ data.

AVRG method. It is observed that standard diffusion-AVRG is more computation efficient than exact diffusion. To reach 10^{-10} relative error, exact diffusion needs around 2×10^5 gradient evaluations while diffusion-AVRG just needs around 2×10^4 gradient evaluations. However, exact diffusion is much more communication efficient than diffusion-AVRG. To see that, exact diffusion requires around 200 communication rounds to reach 10^{-10} error while diffusion-AVRG requires 2×10^4 communication rounds. Similar observation also holds for RCV1 dataset, see Fig. 5.7.

It is also observed in Fig. 5.6 that mini-batch can balance the communication and computation for diffusion-AVRG. As batch size grows, the computation expense increases while the communication expense reduces. Diffusion-AVRG with appropriate batch-size is able to reach better performance than exact diffusion. For example, diffusion-AVRG with $B = 200$ will save around 60% computations while maintaining almost the same amount of communications. Similar observation also holds for RCV1 dataset, see Fig. 5.7.

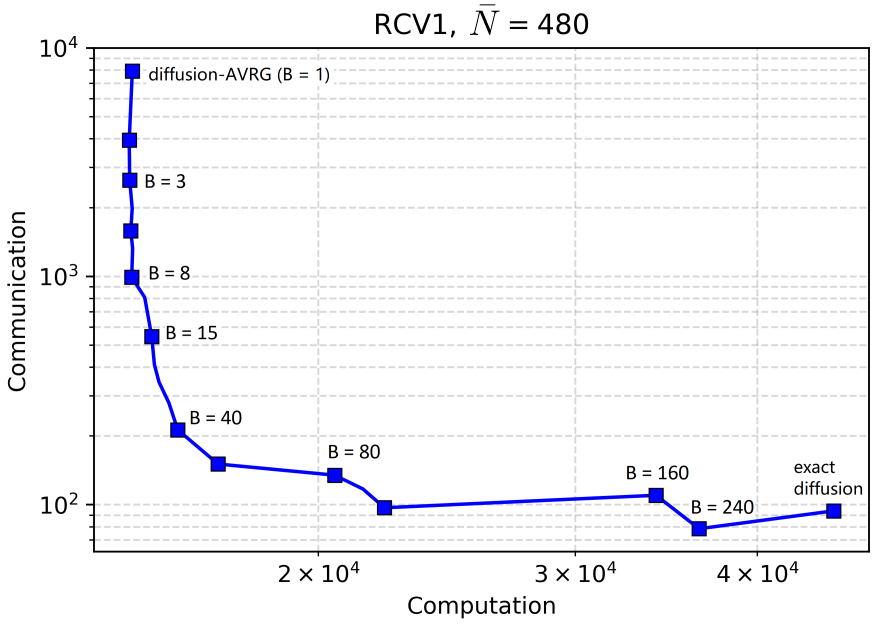

Figure 5.7: Performance of diffusion-AVRG with different batch sizes on RCV1 dataset. Each agent holds $\bar{N} = 480$ data.

5.A Proof of Theorem 5.1

In this section we establish the linear convergence property of diffusion-AVRG (Algorithm 2). We start by transforming the exact diffusion recursions into an equivalent linear error dynamics driven by perturbations due to gradient noise (see Lemma 2). By upper bounding the gradient noise (see Lemma 3), we derive a couple of useful inequalities for the size of the inner iterates (Lemma 4), epoch iterates (Lemma 5), and inner differences (Lemma 6). We finally introduce an energy function and show that it decays exponentially fast (Lemma 7). From this result we will conclude the convergence of $\mathbb{E}\|\boldsymbol{w}_{k,0}^t - w^\star\|^2$ (as stated in (5.21) in Theorem 1). Throughout this section we will consider the practical case where $\bar{N} \geq 2$. When $\bar{N} = 1$, diffusion-AVRG reduces to the exact diffusion algorithm whose convergence is already established in [16].

5.A.1 Extended Network Recursion

Recursions (5.17)–(5.19) of Algorithm 2 only involve local variables $\boldsymbol{w}_{k,i}^t$, $\boldsymbol{\phi}_{k,i}^t$ and $\boldsymbol{\psi}_{k,i}^t$. To analyze the convergence of all $\{\boldsymbol{w}_{k,i}^t\}_{k=1}^K$, we need to combine all iterates from across the network into extended vectors. To do so, we introduce

$$\boldsymbol{\mathcal{W}}_i^t = \mathrm{col}\{\boldsymbol{w}_{1,i}^t, \cdots, \boldsymbol{w}_{K,i}^t\} \tag{5.44}$$

$$\boldsymbol{\phi}_i^t = \mathrm{col}\{\boldsymbol{\phi}_{1,i}^t, \cdots, \boldsymbol{\phi}_{K,i}^t\} \tag{5.45}$$

$$\boldsymbol{\psi}_i^t = \mathrm{col}\{\boldsymbol{\psi}_{1,i}^t, \cdots, \boldsymbol{\psi}_{K,i}^t\} \tag{5.46}$$

$$\nabla\mathcal{J}(\boldsymbol{\mathcal{W}}_i^t) = \mathrm{col}\{\nabla J_1(\boldsymbol{w}_{1,i}^t), \cdots, \nabla J_K(\boldsymbol{w}_{K,i}^t)\} \tag{5.47}$$

$$\widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{W}}_i^t) = \mathrm{col}\{\widehat{\nabla J}_1(\boldsymbol{w}_{1,i}^t), \cdots, \widehat{\nabla J}_K(\boldsymbol{w}_{K,i}^t)\} \tag{5.48}$$

$$\overline{\mathcal{A}} = \overline{A} \otimes I_M \tag{5.49}$$

where $\otimes$ is the Kronecker product. With the above notation, for $0 \le i \le \bar{N} - 1$ and $t \ge 0$, recursions (5.17)–(5.19) of Algorithm 2 can be rewritten as

$$\begin{cases} \boldsymbol{\psi}_{i+1}^t = \boldsymbol{\mathcal{W}}_i^t - \mu\widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{W}}_i^t), \\ \boldsymbol{\phi}_{i+1}^t = \boldsymbol{\psi}_{i+1}^t + \boldsymbol{\mathcal{W}}_i^t - \boldsymbol{\psi}_i^t, \\ \boldsymbol{\mathcal{W}}_{i+1}^t = \overline{\mathcal{A}}\,\boldsymbol{\phi}_{i+1}^t, \end{cases} \tag{5.50}$$

and we let $\boldsymbol{\psi}_0^{t+1} = \boldsymbol{\psi}_{\bar{N}}^t$ and $\boldsymbol{\mathcal{W}}_0^{t+1} = \boldsymbol{\mathcal{W}}_{\bar{N}}^t$. In particular, since $\boldsymbol{\psi}_0^0$ is initialized to be equal to $\boldsymbol{\mathcal{W}}_0^0$, for $t = 0$ and $i = 0$, it holds that

$$\begin{cases} \boldsymbol{\psi}_1^0 = \boldsymbol{\mathcal{W}}_0^0 - \mu\widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{W}}_0^0), \\ \boldsymbol{\phi}_1^0 = \boldsymbol{\psi}_1^0, \\ \boldsymbol{\mathcal{W}}_1^0 = \overline{\mathcal{A}}\,\boldsymbol{\phi}_1^0, \end{cases} \tag{5.51}$$

Substituting the first and second equations of (5.50) into the third one, we have that for $1 \le i \le \bar{N}$ and $t \ge 0$:

$$\boldsymbol{\mathcal{W}}_{i+1}^t = \overline{\mathcal{A}}\left(2\boldsymbol{\mathcal{W}}_i^t - \boldsymbol{\mathcal{W}}_{i-1}^t - \mu[\widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{W}}_i^t) - \widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{W}}_{i-1}^t)]\right), \tag{5.52}$$

and we let $\boldsymbol{\mathcal{W}}_0^{t+1} = \boldsymbol{\mathcal{W}}_{\bar{N}}^t$ and $\boldsymbol{\mathcal{W}}_1^{t+1} = \boldsymbol{\mathcal{W}}_{\bar{N}+1}^t$ for each epoch t. Moreover, we can also rewrite (5.51) as

$$\boldsymbol{\mathcal{W}}_1^0 = \overline{\mathcal{A}}\left(\boldsymbol{\mathcal{W}}_0^0 - \mu\widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{W}}_0^0)\right). \tag{5.53}$$

It is observed that recursion (5.52) involves two consecutive variables $\boldsymbol{\mathcal{w}}_i^t$ and $\boldsymbol{\mathcal{w}}_{i-1}^t$, which complicates the analysis. To deal with this issue, we introduce an auxiliary variable $\boldsymbol{\mathcal{y}}_i^t$ to make the structure in (5.52) more tractable. For that purpose, we first introduce the eigen-decomposition:

$$\frac{1}{2K}(I_K - A) = U\Sigma U^{\mathsf{T}}, \tag{5.54}$$

where Σ is a nonnegative diagonal matrix (note that $I_K - A$ is positive semi-definite because A is doubly stochastic), and U is an orthonormal matrix. We also define

$$V \;\triangleq\; U\Sigma^{1/2}U^{\mathsf{T}}, \qquad \mathcal{V} \;\triangleq\; V \otimes I_M. \tag{5.55}$$

Note that V and $\mathcal{V}$ are symmetric matrices. It can be verified (see Appendix 5.B) that recursion (5.52) is equivalent to

$$\begin{cases} \boldsymbol{\mathcal{w}}_{i+1}^t = \overline{\mathcal{A}}\Big(\boldsymbol{\mathcal{w}}_i^t - \mu\widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{w}}_i^t)\Big) - K\mathcal{V}\boldsymbol{\mathcal{y}}_i^t \\ \boldsymbol{\mathcal{y}}_{i+1}^t = \boldsymbol{\mathcal{y}}_i^t + \mathcal{V}\boldsymbol{\mathcal{w}}_{i+1}^t \end{cases} \tag{5.56}$$

where $0 \le i \le \bar{N}-1$ and $t \ge 0$, $\boldsymbol{\mathcal{y}}_0^0$ is initialized at 0, and $\boldsymbol{\mathcal{w}}_0^{t+1} = \boldsymbol{\mathcal{w}}_{\bar{N}}^t$, $\boldsymbol{\mathcal{y}}_0^{t+1} = \boldsymbol{\mathcal{y}}_{\bar{N}}^t$ after epoch t. Note that recursion (5.56) is very close to recursion for exact diffusion (see equation (93) in [15]), except that $\widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{w}}_i^t)$ is a stochastic gradient generated by AVRG. We denote the gradient noise by

$$\boldsymbol{s}(\boldsymbol{\mathcal{w}}_i^t) = \widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{w}}_i^t) - \nabla\mathcal{J}(\boldsymbol{\mathcal{w}}_i^t). \tag{5.57}$$

Substituting into (5.56), we get

$$\begin{cases} \boldsymbol{\mathcal{w}}_{i+1}^t = \overline{\mathcal{A}}\Big(\boldsymbol{\mathcal{w}}_i^t - \mu\nabla\mathcal{J}(\boldsymbol{\mathcal{w}}_i^t)\Big) - K\mathcal{V}\boldsymbol{\mathcal{y}}_i^t - \mu\overline{\mathcal{A}}\,\boldsymbol{s}(\boldsymbol{\mathcal{w}}_i^t) \\ \boldsymbol{\mathcal{y}}_{i+1}^t = \boldsymbol{\mathcal{y}}_i^t + \mathcal{V}\boldsymbol{\mathcal{w}}_{i+1}^t \end{cases} \tag{5.58}$$

In summary, the exact diffusion recursions (5.17)–(5.19) of Algorithm 2 are equivalent to form (5.58).

5.A.2 Optimality Condition

It is proved in Lemma 4 of [16] that there exists a *unique* pair of variables $(w^\star, y_o^\star)$, with $y_o^\star$ lying in the range space of $\mathcal{V}$, such that

$$\mu\overline{\mathcal{A}}\nabla\mathcal{J}(w^\star) + K\mathcal{V}y_o^\star = 0 \quad \text{and} \quad \mathcal{V}w^\star = 0, \tag{5.59}$$

where we partition $w^\star$ into block entries of size $M\times 1$ each as follows: $w^\star = \mathrm{col}\{w_1^\star, w_2^\star, \cdots, w_K^\star\} \in \mathbb{R}^{KM}$. For such $(w^\star, y_o^\star)$, it further holds that the block entries of $w^\star$ are identical and coincide with the unique solution to problem (4), i.e.

$$w_1^\star = w_2^\star = \cdots = w_K^\star = w^\star. \tag{5.60}$$

In other words, equation (5.59) is the optimality condition characterizing the solution to problem (5.4).

5.A.3 Error Dynamics

Let $\widetilde{\boldsymbol{w}}_i^t = w^\star - \boldsymbol{w}_i^t$ and $\widetilde{\boldsymbol{y}}_i^t = y_o^\star - \boldsymbol{y}_i^t$ denote error vectors relative to the solution pair $(w^\star, y_o^\star)$. It is proved in Appendix 5.C that recursion (5.58), under Assumption 5.1, can be transformed into the following recursion driven by a gradient noise term:

$$\begin{bmatrix} \widetilde{\boldsymbol{w}}_{i+1}^t \\ \widetilde{\boldsymbol{y}}_{i+1}^t \end{bmatrix} = (\mathcal{B} - \mu\boldsymbol{\mathcal{T}}_i^t) \begin{bmatrix} \widetilde{\boldsymbol{w}}_i^t \\ \widetilde{\boldsymbol{y}}_i^t \end{bmatrix} + \mu\mathcal{B}_l \boldsymbol{s}(\boldsymbol{w}_i^t), \tag{5.61}$$

where $0 \le i \le \bar{N}-1$, $t \ge 0$, and $\widetilde{\boldsymbol{w}}_0^{t+1} = \widetilde{\boldsymbol{w}}_{\bar{N}}^t$, $\widetilde{\boldsymbol{y}}_0^{t+1} = \widetilde{\boldsymbol{y}}_{\bar{N}}^t$ after epoch t. Moreover, $\mathcal{B}, \mathcal{B}_l$ and $\boldsymbol{\mathcal{T}}_i^t$ are defined as

$$\mathcal{B} \triangleq \begin{bmatrix} \overline{\mathcal{A}} & -K\mathcal{V} \\ \mathcal{V}\overline{\mathcal{A}} & \overline{\mathcal{A}} \end{bmatrix}, \ \mathcal{B}_l \triangleq \begin{bmatrix} \overline{\mathcal{A}} \\ \mathcal{V}\overline{\mathcal{A}} \end{bmatrix}, \ \boldsymbol{\mathcal{T}}_i^t \triangleq \begin{bmatrix} \overline{\mathcal{A}}\boldsymbol{\mathcal{H}}_i^t & 0 \\ \mathcal{V}\overline{\mathcal{A}}\boldsymbol{\mathcal{H}}_i^t & 0 \end{bmatrix}, \tag{5.62}$$

where

$$\boldsymbol{\mathcal{H}}_i^t = \mathrm{diag}\{\boldsymbol{H}_{1,i}^t, \cdots, \boldsymbol{H}_{K,i}^t\} \in \mathbb{R}^{KM\times KM}, \tag{5.63}$$

$$\boldsymbol{H}_{k,i}^t \triangleq \int_0^1 \nabla^2 J_k\big(w^\star - r\widetilde{\boldsymbol{w}}_{k,i}^t\big)\, dr \in \mathbb{R}^{M\times M}. \tag{5.64}$$

To facilitate the convergence analysis of recursion (5.61), we diagonalize $\mathcal{B}$ and transform (5.61) into an equivalent error dynamics. From equations (64)–(67) in [16], we know that $\mathcal{B}$ admits an eigen-decomposition of the form

$$\mathcal{B} \ \triangleq \ \mathcal{X}\mathcal{D}\mathcal{X}^{-1}, \tag{5.65}$$

where $\mathcal{X}, \mathcal{D}$ and $\mathcal{X}^{-1}$ are KM by KM matrices defined as

$$\mathcal{D} \triangleq \begin{bmatrix} I_M & 0 & 0 \\ 0 & I_M & 0 \\ 0 & 0 & \mathcal{D}_1 \end{bmatrix} \in \mathbb{R}^{2KM\times 2KM}, \tag{5.66}$$

$$\mathcal{X} \triangleq \begin{bmatrix} \mathcal{R}_1 & \mathcal{R}_2 & \mathcal{X}_R \end{bmatrix} \in \mathbb{R}^{2KM\times 2KM}, \tag{5.67}$$

$$\mathcal{X}^{-1} \triangleq \begin{bmatrix} \mathcal{L}_1^{\mathsf{T}} \\ \mathcal{L}_2^{\mathsf{T}} \\ \mathcal{X}_L \end{bmatrix} \in \mathbb{R}^{2KM\times 2KM}. \tag{5.68}$$

In (5.66), matrix $\mathcal{D}_1 = D_1 \otimes I_M$ and $D_1 \in \mathbb{R}^{2(K-1)\times 2(K-1)}$ is a diagonal matrix with $\|D_1\| = \lambda_2(A) \triangleq \lambda < 1$. In (5.67) and (5.68), matrices $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{L}_1$ and $\mathcal{L}_2$ take the form

$$\mathcal{R}_1 = \begin{bmatrix} \mathbb{1}_K \\ 0_K \end{bmatrix} \otimes I_M, \quad \mathcal{R}_2 = \begin{bmatrix} 0_K \\ \mathbb{1}_K \end{bmatrix} \otimes I_M \tag{5.69}$$

$$\mathcal{L}_1 = \begin{bmatrix} \frac{1}{K}\mathbb{1}_K \\ 0_K \end{bmatrix} \otimes I_M, \quad \mathcal{L}_2 = \begin{bmatrix} 0_K \\ \frac{1}{K}\mathbb{1}_K \end{bmatrix} \otimes I_M \tag{5.70}$$

Moreover, $\mathcal{X}_R \in \mathbb{R}^{2KM\times 2(K-1)M}$ and $\mathcal{X}_L \in \mathbb{R}^{2(K-1)M\times 2KM}$ are some constant matrices. Since $\mathcal{B}$ is independent of $\bar{N}$, δ and ν, all matrices appearing in (5.65)–(5.68) are independent of these variables as well. By multiplying $\mathcal{X}^{-1}$ to both sides of recursion (5.61), we have

$$\begin{aligned} &\mathcal{X}^{-1}\begin{bmatrix} \widetilde{\boldsymbol{w}}_{i+1}^t \\ \widetilde{\boldsymbol{\mathcal{Y}}}_{i+1}^t \end{bmatrix} \\ =&[\mathcal{X}^{-1}(\mathcal{B}-\mu\boldsymbol{\mathcal{T}}_i^t)\mathcal{X}]\mathcal{X}^{-1}\begin{bmatrix} \widetilde{\boldsymbol{w}}_{i}^t \\ \widetilde{\boldsymbol{\mathcal{Y}}}_{i}^t \end{bmatrix} + \mu\mathcal{X}^{-1}\mathcal{B}_l\boldsymbol{s}(\boldsymbol{w}_i^t) \\ \overset{(3.42)}{=}&\Big(\mathcal{D}-\mu\mathcal{X}^{-1}\boldsymbol{\mathcal{T}}_i^t\mathcal{X}\Big)\left(\mathcal{X}^{-1}\begin{bmatrix} \widetilde{\boldsymbol{w}}_{i}^t \\ \widetilde{\boldsymbol{\mathcal{Y}}}_{i}^t \end{bmatrix}\right) + \mu\mathcal{X}^{-1}\mathcal{B}_l\boldsymbol{s}(\boldsymbol{w}_i^t). \end{aligned} \tag{5.71}$$

Now we define

$$\begin{bmatrix} \bar{\boldsymbol{x}}_i^t \\ \widehat{\boldsymbol{x}}_i^t \\ \check{\boldsymbol{x}}_i^t \end{bmatrix} \triangleq \mathcal{X}^{-1}\begin{bmatrix} \widetilde{\boldsymbol{w}}_{i}^t \\ \widetilde{\boldsymbol{\mathcal{Y}}}_{i}^t \end{bmatrix} \overset{(5.68)}{=} \begin{bmatrix} \mathcal{L}_1^{\mathsf{T}} \\ \mathcal{L}_2^{\mathsf{T}} \\ \mathcal{X}_L \end{bmatrix}\begin{bmatrix} \widetilde{\boldsymbol{w}}_{i}^t \\ \widetilde{\boldsymbol{\mathcal{Y}}}_{i}^t \end{bmatrix}, \tag{5.72}$$

as transformed errors. Moreover, we partition $\mathcal{X}_R$ as

$$\mathcal{X}_R = \begin{bmatrix} \mathcal{X}_{R,u} \\ \mathcal{X}_{R,d} \end{bmatrix}, \quad \text{where} \quad \mathcal{X}_{R,u} \in \mathbb{R}^{KM \times 2(K-1)M}. \tag{5.73}$$

With the help of recursion (5.71), we can establish the following lemma.

Lemma 5.1 (Useful Transformation) *When $\boldsymbol{\mathcal{Y}}_0^0$ is initialized at 0, recursion (5.61) can be transformed into*

$$\begin{bmatrix} \bar{\boldsymbol{x}}_{i+1}^t \\ \check{\boldsymbol{x}}_{i+1}^t \end{bmatrix} = \begin{bmatrix} I_M - \frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{I} & -\frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{X}_{R,u} \\ -\mu\mathcal{X}_L\boldsymbol{\mathcal{T}}_i^t\mathcal{R}_1 & \mathcal{D}_1 - \mu\mathcal{X}_L\boldsymbol{\mathcal{T}}_i^t\mathcal{X}_R \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{x}}_i^t \\ \check{\boldsymbol{x}}_i^t \end{bmatrix} + \mu \begin{bmatrix} \frac{1}{K}\mathcal{I}^{\mathsf{T}} \\ \mathcal{X}_L\mathcal{B}_l \end{bmatrix} \boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t) \tag{5.74}$$

where $\mathcal{I} = \mathbb{1}_K \otimes I_M$. Moreover, the relation between $\widetilde{\boldsymbol{\mathcal{W}}}_i^t, \widetilde{\boldsymbol{\mathcal{Y}}}_i^t$ and $\bar{\boldsymbol{x}}_i^t, \check{\boldsymbol{x}}_i^t$ in (5.71) reduces to

$$\begin{bmatrix} \widetilde{\boldsymbol{\mathcal{W}}}_i^t \\ \widetilde{\boldsymbol{\mathcal{Y}}}_i^t \end{bmatrix} = \mathcal{X} \begin{bmatrix} \bar{\boldsymbol{x}}_i^t \\ 0_M \\ \check{\boldsymbol{x}}_i^t \end{bmatrix}. \tag{5.75}$$

Notice that $\mathcal{X}_L$, $\mathcal{X}_R$, $\mathcal{X}_{R,u}$ and $\mathcal{X}$ are all constant matrices and independent of $\bar{N}, \delta$ and ν.

Proof. See Appendix 5.D. The proof is similar to the derivations in equations (68)–(82) from [16] except that we have an additional noise term in (5.61). ■

Starting from (5.74), we can derive the following recursions for the mean-square errors of the quantities $\bar{\boldsymbol{x}}_i^t$ and $\check{\boldsymbol{x}}_i^t$.

Lemma 5.2 (Mean-square-error Recursion) *Under Assum-ption (5.1), $\boldsymbol{\mathcal{Y}}_0^0 = 0$ and for step-size $\mu < 1/\delta$, it holds that*

$$\begin{bmatrix} \mathbb{E}\|\bar{\boldsymbol{x}}_{i+1}^t\|^2 \\ \mathbb{E}\|\check{\boldsymbol{x}}_{i+1}^t\|^2 \end{bmatrix} \preceq \begin{bmatrix} 1 - a_1\mu\nu & \frac{2a_2\mu\delta^2}{\nu} \\ a_4\mu^2\delta^2 & \lambda + a_3\mu^2\delta^2 \end{bmatrix} \begin{bmatrix} \mathbb{E}\|\bar{\boldsymbol{x}}_i^t\|^2 \\ \mathbb{E}\|\check{\boldsymbol{x}}_i^t\|^2 \end{bmatrix} + \begin{bmatrix} \frac{2\mu}{\nu}\mathbb{E}\|\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\|^2 \\ a_5\mu^2\mathbb{E}\|\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\|^2 \end{bmatrix}, \tag{5.76}$$

where the scalars $a_l, 1 \le l \le 5$ are defined in (5.152); they are positive constants that are independent of $\bar{N}$, δ and ν.

Proof. See Appendix 5.E. ■

It is observed that recursion (5.76) still mixes gradient noise $\mathbb{E}\|\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\|^2$ (which is correlated with $\boldsymbol{\mathcal{W}}_i^t$) with iterates $\bar{\boldsymbol{\mathcal{X}}}_i^t$ and $\check{\boldsymbol{\mathcal{X}}}_i^t$. To establish the convergence of $\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_i^t\|^2$ and $\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_i^t\|^2$, we need to upper bound $\mathbb{E}\|\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\|^2$ with terms related to $\bar{\boldsymbol{\mathcal{X}}}_i^t$ and $\check{\boldsymbol{\mathcal{X}}}_i^t$. In the following lemma we provide such an upper bound.

Lemma 5.3 (Gradient Noise) *Under Assumption 5.1, the second moment of the gradient noise term satisfies:*

$$\begin{aligned}
&\mathbb{E}\|\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\|^2 \\
&\leq 6b\delta^2\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_i^t - \bar{\boldsymbol{\mathcal{X}}}_0^t\|^2 + 12b\delta^2\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_i^t\|^2 + 18b\delta^2\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^t\|^2 \\
&\quad + \frac{3b\delta^2}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_j^{t-1} - \bar{\boldsymbol{\mathcal{X}}}_{\bar{N}}^{t-1}\|^2 + \frac{6b\delta^2}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_j^{t-1}\|^2,
\end{aligned} \tag{5.77}$$

where $b = \|\mathcal{X}\|^2$ is a positive constant that is independent of $\bar{N}$, ν and δ.

Proof. See Appendix 5.F. ■

In the following subsections, we will exploit the error dynamic (3.164) and the upper bound (5.77) to establish the convergence of $\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_i^t\|^2$ and $\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_i^t\|^2$, from which we will conclude later the convergence of $\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_i^t\|^2$.

5.A.4 Useful Inequalities

To simplify the notation, we define

$$\mathbf{A}^t \triangleq \frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_j^t - \bar{\boldsymbol{\mathcal{X}}}_0^t\|^2, \tag{5.78}$$

$$\mathbf{B}^t \triangleq \frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_j^t - \bar{\boldsymbol{\mathcal{X}}}_{\bar{N}}^t\|^2, \tag{5.79}$$

$$\mathbf{C}^t \triangleq \frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_j^t\|^2. \tag{5.80}$$

All these quantities appear in the upper bound on gradient noise in (5.77), and their recursions will be required to establish the final convergence theorem.

Lemma 5.4 ($\mathbb{E}\|\check{\boldsymbol{x}}_i^t\|^2$ Recursion) *Suppose Assumption 1 holds. If the step-size μ satisfies*

$$\mu \leq C_1 \sqrt{\frac{1-\lambda}{\delta^2 \bar{N}}}, \tag{5.81}$$

where $C_1 > 0$, which is defined in (5.178)*, is a constant independent of $\bar{N}$, ν and δ, it then holds that*

$$\begin{aligned}\mathbf{C}^t \leq\ & c_1\mu^2\delta^2\bar{N}\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 + \lambda_3\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2 + c_2\mu^2\delta^2\bar{N}\mathbf{A}^t \\ & + c_3\mu^2\delta^2\bar{N}\mathbf{B}^{t-1} + c_4\mu^2\delta^2\bar{N}\mathbf{C}^{t-1}, \end{aligned} \tag{5.82}$$

$$\begin{aligned}\mathbb{E}\|\check{\boldsymbol{x}}_0^{t+1}\|^2 \leq\ & c_1\mu^2\delta^2\bar{N}\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 + \lambda_2\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2 + c_2\mu^2\delta^2\bar{N}\mathbf{A}^t \\ & + c_3\mu^2\delta^2\bar{N}\mathbf{B}^{t-1} + c_4\mu^2\delta^2\bar{N}\mathbf{C}^{t-1}, \end{aligned} \tag{5.83}$$

where the constants $\lambda_2 < 1,\ \lambda_3 < 1$, and $\{c_l\}_{l=1}^4$, which are defined in Appendix 5.G, are all positive scalars that are independent of $\bar{N}$, ν and δ.

Proof. See Appendix 5.G. ■

Lemma 5.5 ($\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2$ Recursion) *Suppose Assumption 1 holds. If the step-size μ satisfies*

$$\mu \leq C_2 \left(\frac{\nu\sqrt{1-\lambda}}{\delta^2 \bar{N}} \right), \tag{5.84}$$

where $C_2 > 0$, which is defined in (5.190)*, is a constant independent of $\bar{N}$, ν and δ, it then holds that*

$$\begin{aligned}&\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t+1}\|^2 \\ &\leq \left(1 - \frac{\bar{N}}{3}a_1\mu\nu\right)\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 + \frac{d_1\mu\delta^2\bar{N}}{\nu}\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2 \\ &\quad + \frac{d_2\delta^2\mu\bar{N}}{\nu}\mathbf{A}^t + \frac{d_3\delta^2\mu\bar{N}}{\nu}\mathbf{B}^{t-1} + \frac{d_4\delta^2\mu\bar{N}}{\nu}\mathbf{C}^{t-1} \end{aligned} \tag{5.85}$$

where $\{d_l\}_{l=1}^4$, which are defined in (5.188)*, are positive constants that are independent of $\bar{N}$, ν and δ.*

Proof. See Appendix 5.H. ■

Lemma 5.6 (Inner Difference Recursion) *Suppose Assumption 1 holds. If the step-size μ satisfies*

$$\mu \leq C_3 \sqrt{\frac{1-\lambda}{\delta^2 \bar{N}}}, \tag{5.86}$$

where $C_3 > 0$, which is defined in (5.205)*, is a constant independent of $\bar{N}$, ν and δ, it then holds that*

$$\begin{aligned}\mathbf{A}^t \leq & 12\mu^2\delta^2\bar{N}^2\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 + e_6\mu^2\delta^2\bar{N}^2\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2 + 2e_3\mu^2\delta^2\bar{N}^2\mathbf{A}^t \\ & + 2e_4\mu^2\delta^2\bar{N}^2\mathbf{B}^{t-1} + 2e_5\mu^2\delta^2\bar{N}^2\mathbf{C}^{t-1}, \end{aligned} \tag{5.87}$$

$$\begin{aligned}\mathbf{B}^t \leq & 12\mu^2\delta^2\bar{N}^2\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 + e_6\mu^2\delta^2\bar{N}^2\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2 + 2e_3\mu^2\delta^2\bar{N}^2\mathbf{A}^t \\ & + 2e_4\mu^2\delta^2\bar{N}^2\mathbf{B}^{t-1} + 2e_5\mu^2\delta^2\bar{N}^2\mathbf{C}^{t-1} \end{aligned} \tag{5.88}$$

where $\{e_i\}_{i=3}^6$, which are defined in (5.198)*, are positive constants that are independent of $\bar{N}$, ν and δ.*

Proof. See Appendix 5.J. ■

5.A.5 Linear Convergence

With the above inequalities, we are ready to establish the linear convergence of the transformed diffusion-AVRG recursion (5.74).

Lemma 5.7 (Linear Convergence) *Under Assumption 5.1, if the step-size μ satisfies*

$$\mu \leq C\left(\frac{\nu(1-\lambda)}{\delta^2\bar{N}}\right), \tag{5.89}$$

where $C > 0$, which is defined in (5.246)*, is a constant independent of $\bar{N}$, ν and δ, and $\lambda = \lambda_2(A)$ is second largest eigenvalue of the combination matrix A, it then holds that*

$$\begin{aligned}&\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t+1}\|^2 + \mathbb{E}\|\check{\boldsymbol{x}}_0^{t+1}\|^2\right) + \frac{\gamma}{2}\left(\mathbf{A}^{t+1} + \mathbf{B}^t + \mathbf{C}^t\right) \\ \leq & \rho\left\{\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 + \mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2\right) + \frac{\gamma}{2}(\mathbf{A}^t + \mathbf{B}^{t-1} + \mathbf{C}^{t-1})\right\}\end{aligned} \tag{5.90}$$

where $\gamma = 8f_5\delta^2\mu\bar{N}/\nu > 0$ is a constant, and

$$\rho = \frac{1 - \frac{\bar{N}}{8}a_1\mu\nu}{1 - 8f_1f_5\mu^3\delta^4\bar{N}^3/\nu} < 1. \tag{5.91}$$

The positive constants a_1, f_1 and f_5 are independent of $\bar{N}$, ν and δ. Their definitions are in (5.152) *and* (5.214)*.*

Proof. See Appendix 5.K. ■

Using Lemma 5.7, we can now establish the earlier Theorem 5.1.

Proof of Theorem 5.1. From recursion (5.90), we conclude that

$$\begin{aligned}&\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t+1}\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^{t+1}\|^2\right)+\frac{\gamma}{2}\left(\mathbf{A}^{t+1}+\mathbf{B}^t+\mathbf{C}^t\right)\\&\leq\rho^t\left\{\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^1\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^1\|^2\right)+\frac{\gamma}{2}(\mathbf{A}^1+\mathbf{B}^0+\mathbf{C}^0)\right\}.\end{aligned}\tag{5.92}$$

Since $\gamma>0$, it also holds that

$$\begin{aligned}&\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t+1}\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^{t+1}\|^2\\&\leq\rho^t\left\{\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^1\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^1\|^2\right)+\frac{\gamma}{2}(\mathbf{A}^1+\mathbf{B}^0+\mathbf{C}^0)\right\}.\end{aligned}\tag{5.93}$$

On the other hand, from (5.75) we have

$$\|\widetilde{\boldsymbol{\mathcal{W}}}_0^{t+1}\|^2+\|\widetilde{\boldsymbol{\mathcal{Y}}}_0^{t+1}\|^2\leq\|\mathcal{X}\|^2\left(\|\bar{\boldsymbol{x}}_0^{t+1}\|^2+\|\check{\boldsymbol{x}}_0^{t+1}\|^2\right).\tag{5.94}$$

By taking expectation of both sides, we have

$$\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_0^{t+1}\|^2+\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{Y}}}_0^{t+1}\|^2\leq\|\mathcal{X}\|^2\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t+1}\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^{t+1}\|^2\right).\tag{5.95}$$

Combining (5.93) and (5.95), we have

$$\begin{aligned}&\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_0^{t+1}\|^2+\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{Y}}}_0^{t+1}\|^2\\&\leq\rho^t\underbrace{\left(\|\mathcal{X}\|^2\left\{\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^1\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^1\|^2\right)+\frac{\gamma}{2}(\mathbf{A}^1+\mathbf{B}^0+\mathbf{C}^0)\right\}\right)}_{\triangleq\,D}.\end{aligned}\tag{5.96}$$

Since $\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_0^{t+1}\|^2=\sum_{k=1}^K\mathbb{E}\|w^\star-\boldsymbol{w}_{k,0}^{t+1}\|^2\leq\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_0^{t+1}\|^2+\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{Y}}}_0^{t+1}\|^2$, we conclude (5.21). ■

5.B Proof of recursion (5.56)

Since $V=U\Sigma^{1/2}U^{\mathsf{T}}$, it holds that

$$V^2=U\Sigma U^{\mathsf{T}}\overset{(5.54)}{=}(I_K-A)/2K,\tag{5.97}$$

which implies that

$$\mathcal{V}^2=V^2\otimes I_M=(I_{KM}-\mathcal{A})/2K.\tag{5.98}$$

Moreover, since $A\mathbb{1}_K=\mathbb{1}_K$ we get

$$V^2\mathbb{1}_K=(I_{KM}-A)\mathbb{1}_K/2K=0.\tag{5.99}$$

By noting that $\|V\mathbb{1}_K\|^2=\mathbb{1}_K^{\mathsf{T}}V^2\mathbb{1}_K=0$, we conclude that

$$V\mathbb{1}_K=0,\quad\text{and}\quad\mathcal{V}\mathcal{I}=0,\tag{5.100}$$

where $\mathcal{I}\triangleq\mathbb{1}_K\otimes I_M$. Result (5.100) will be used in Appendix 5.D.

Now, for $t=0$ and $i=0$, substituting $\boldsymbol{\mathcal{Y}}_0^0=0$ into (5.56) we have

$$\begin{cases} \boldsymbol{\mathcal{W}}_1^0 = \overline{\mathcal{A}}\Big(\boldsymbol{\mathcal{W}}_0^0 - \mu\widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{W}}_0^0)\Big) \\ \boldsymbol{\mathcal{Y}}_1^0 = \mathcal{V}\boldsymbol{\mathcal{W}}_1^0 \end{cases} \tag{5.101}$$

The first expression in (5.101) is exactly the first expression in (5.52). For $t\geq 0$ and $1\leq i\leq \bar{N}$, from the first recursion in (5.56) we have

$$\begin{aligned} \boldsymbol{\mathcal{W}}_{i+1}^t - \boldsymbol{\mathcal{W}}_i^t =& \overline{\mathcal{A}}\left(\boldsymbol{\mathcal{W}}_i^t - \boldsymbol{\mathcal{W}}_{i-1}^t - \mu\big(\widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{W}}_i^t) - \widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{W}}_{i-1}^t)\big)\right) \\ & - K\mathcal{V}(\boldsymbol{\mathcal{Y}}_i^t - \boldsymbol{\mathcal{Y}}_{i-1}^t), \end{aligned} \tag{5.102}$$

We let $\boldsymbol{\mathcal{W}}_1^{t+1}=\boldsymbol{\mathcal{W}}_{\bar{N}+1}^t$ and $\boldsymbol{\mathcal{W}}_0^{t+1}=\boldsymbol{\mathcal{W}}_{\bar{N}}^t$ after epoch t. Recalling from the second recursion in (5.56) that $\boldsymbol{\mathcal{Y}}_i^t-\boldsymbol{\mathcal{Y}}_{i-1}^t=\mathcal{V}\boldsymbol{\mathcal{W}}_i^t$, and substituting into (5.102) we get

$$\begin{aligned} & \boldsymbol{\mathcal{W}}_{i+1}^t - \boldsymbol{\mathcal{W}}_i^t \\ & = \overline{\mathcal{A}}\left(\boldsymbol{\mathcal{W}}_i^t - \boldsymbol{\mathcal{W}}_{i-1}^t - \mu\big(\widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{W}}_i^t) - \widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{W}}_{i-1}^t)\big)\right) - K\mathcal{V}^2\boldsymbol{\mathcal{W}}_i^t \\ & \overset{(5.98)}{=} \overline{\mathcal{A}}\left(\boldsymbol{\mathcal{W}}_i^t - \boldsymbol{\mathcal{W}}_{i-1}^t - \mu\big(\widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{W}}_i^t) - \widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{W}}_{i-1}^t)\big)\right) \\ & \quad - \frac{1}{2}(I_{KM}-\mathcal{A})\,\boldsymbol{\mathcal{W}}_i^t. \end{aligned} \tag{5.103}$$

Using $\overline{\mathcal{A}}=(I_{KM}+\mathcal{A})/2$, the above recursion can be rewritten as

$$\boldsymbol{\mathcal{W}}_{i+1}^t = \overline{\mathcal{A}}\left(2\boldsymbol{\mathcal{W}}_i^t - \boldsymbol{\mathcal{W}}_{i-1}^t - \mu\big(\widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{W}}_i^t) - \widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{W}}_{i-1}^t)\big)\right) \tag{5.104}$$

which is the second recursion in (5.52).

5.C Proof of recursion (5.61)

The proof of (5.61) is similar to (36)–(50) in [16] except that we have an additional gradient noise term $\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)$. We subtract $w^\star$ and $\mathcal{Y}_o^\star$ from both sides of (5.58) respectively and use the fact that $\overline{\mathcal{A}}w^\star=\frac{1}{2}(I_{MK}+\mathcal{A})w^\star=w^\star$ to get

$$\begin{cases} \widetilde{\boldsymbol{\mathcal{W}}}_{i+1}^t = \overline{\mathcal{A}}\Big(\widetilde{\boldsymbol{\mathcal{W}}}_i^t + \mu\nabla\mathcal{J}(\boldsymbol{\mathcal{W}}_i^t)\Big) + K\mathcal{V}\boldsymbol{\mathcal{Y}}_i^t + \mu\overline{\mathcal{A}}\,\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t) \\ \widetilde{\boldsymbol{\mathcal{Y}}}_{i+1}^t = \widetilde{\boldsymbol{\mathcal{Y}}}_i^t - \mathcal{V}\boldsymbol{\mathcal{W}}_{i+1}^t \end{cases} \tag{5.105}$$

Subtracting the optimality condition (5.59) from (5.105) gives

$$\begin{cases} \widetilde{\boldsymbol{w}}_{i+1}^t = \overline{\mathcal{A}}\Big(\widetilde{\boldsymbol{w}}_i^t + \mu[\nabla\mathcal{J}(\boldsymbol{w}_i^t) - \nabla\mathcal{J}(w^\star)]\Big) \\ \qquad\qquad + K\mathcal{V}(\boldsymbol{y}_i^t - \mathcal{Y}_o^\star) + \mu\overline{\mathcal{A}}\,\boldsymbol{s}(\boldsymbol{w}_i^t) \\ \widetilde{\boldsymbol{y}}_{i+1}^t = \widetilde{\boldsymbol{y}}_i^t - \mathcal{V}(\boldsymbol{w}_{i+1}^t - w^\star) \end{cases} \tag{5.106}$$

Recall that $\nabla\mathcal{J}(w)$ is twice-differentiable (see Assumption 5.1). We can then appeal to the mean-value theorem (see equations (40)–(43) in [16]) to express the gradient difference as

$$\nabla\mathcal{J}(\boldsymbol{w}_i^t) - \nabla\mathcal{J}(w^\star) = -\boldsymbol{\mathcal{H}}_i^t\widetilde{\boldsymbol{w}}_i^t, \tag{5.107}$$

where $\boldsymbol{\mathcal{H}}_i^t$ is defined in (2.102). With (5.107), recursion (5.106) becomes

$$\begin{cases} \widetilde{\boldsymbol{w}}_{i+1}^t = \overline{\mathcal{A}}\Big(I_{MK} - \mu\boldsymbol{\mathcal{H}}_i^t\Big)\widetilde{\boldsymbol{w}}_i^t - K\mathcal{V}\widetilde{\boldsymbol{y}}_i^t + \mu\overline{\mathcal{A}}\,\boldsymbol{s}(\boldsymbol{w}_i^t) \\ \widetilde{\boldsymbol{y}}_{i+1}^t = \widetilde{\boldsymbol{y}}_i^t + \mathcal{V}\widetilde{\boldsymbol{w}}_{i+1}^t \end{cases} \tag{5.108}$$

From relations (5.54) and (5.55), we conclude that $V^2 = (I_K - A)/2K$, which also implies that $\mathcal{V}^2 = (I_{MK} - \mathcal{A})/2K$. With this fact, we substitute the second recursion in (5.108) into the first recursion to get

$$\begin{cases} \overline{\mathcal{A}}\widetilde{\boldsymbol{w}}_{i+1}^t = \overline{\mathcal{A}}\Big(I_{MK} - \mu\boldsymbol{\mathcal{H}}_i^t\Big)\widetilde{\boldsymbol{w}}_i^t - K\mathcal{V}\widetilde{\boldsymbol{y}}_{i+1}^t + \mu\overline{\mathcal{A}}\,\boldsymbol{s}(\boldsymbol{w}_i^t) \\ \widetilde{\boldsymbol{y}}_{i+1}^t = \widetilde{\boldsymbol{y}}_i^t + \mathcal{V}\widetilde{\boldsymbol{w}}_{i+1}^t \end{cases} \tag{5.109}$$

which is also equivalent to

$$\begin{aligned} &\begin{bmatrix} \overline{\mathcal{A}} & K\mathcal{V} \\ -\mathcal{V} & I_{MK} \end{bmatrix} \begin{bmatrix} \widetilde{\boldsymbol{w}}_{i+1}^t \\ \widetilde{\boldsymbol{y}}_{i+1}^t \end{bmatrix} \\ &= \begin{bmatrix} \overline{\mathcal{A}}\Big(I_{MK} - \mu\boldsymbol{\mathcal{H}}_i^t\Big) & 0 \\ 0 & I_{MK} \end{bmatrix} \begin{bmatrix} \widetilde{\boldsymbol{w}}_i^t \\ \widetilde{\boldsymbol{y}}_i^t \end{bmatrix} + \begin{bmatrix} \mu\overline{\mathcal{A}} \\ 0 \end{bmatrix} \boldsymbol{s}(\boldsymbol{w}_i^t). \end{aligned} \tag{5.110}$$

Also recall (5.54) that $A = I_K - 2KU\Sigma U^{\mathsf{T}}$. Therefore,

$$\overline{A} = \frac{I_K + A}{2} = I_K - KU\Sigma U^{\mathsf{T}} = U(I_K - K\Sigma)U^{\mathsf{T}}. \tag{5.111}$$

This together with the fact that $V = U\Sigma^{1/2}U^{\mathsf{T}}$ leads to

$$V\overline{A} = U\Sigma^{1/2}U^{\mathsf{T}}U(I_K - K\Sigma)U^{\mathsf{T}} \tag{5.112}$$

$$= U\Sigma^{1/2}(I_K - K\Sigma)U^{\mathsf{T}} = U(I_K - K\Sigma)\Sigma^{1/2}U^{\mathsf{T}} = \overline{A}V, \tag{5.113}$$

which also implies that $\mathcal{V}\overline{\mathcal{A}} = \overline{\mathcal{A}}\mathcal{V}$. As a result, we can verify that

$$\begin{bmatrix} \overline{\mathcal{A}} & K\mathcal{V} \\ -\mathcal{V} & I_{MK} \end{bmatrix}^{-1} = \begin{bmatrix} I_{MK} & -K\mathcal{V} \\ \mathcal{V} & \overline{\mathcal{A}} \end{bmatrix}. \tag{5.114}$$

Substituting the above relation into (5.110), we get

$$\begin{aligned} \begin{bmatrix} \widetilde{\boldsymbol{w}}_{i+1}^t \\ \widetilde{\boldsymbol{\mathcal{Y}}}_{i+1}^t \end{bmatrix} = & \begin{bmatrix} \overline{\mathcal{A}}\Big(I_{MK} - \mu\boldsymbol{\mathcal{H}}_i^t\Big) & -K\mathcal{V} \\ \mathcal{V}\overline{\mathcal{A}}\Big(I_{MK} - \mu\boldsymbol{\mathcal{H}}_i^t\Big) & \overline{\mathcal{A}} \end{bmatrix} \begin{bmatrix} \widetilde{\boldsymbol{w}}_i^t \\ \widetilde{\boldsymbol{\mathcal{Y}}}_i^t \end{bmatrix} \\ & + \mu \begin{bmatrix} \overline{\mathcal{A}} \\ \mathcal{V}\overline{\mathcal{A}} \end{bmatrix} \boldsymbol{s}(\boldsymbol{w}_i^t) \end{aligned} \tag{5.115}$$

which matches equations (5.61)–(3.28).

5.D Proof of Lemma 5.1

Now We examine the recursion (5.71). By following the derivation in equations (71)–(77) from [16], we have

$$\mathcal{X}^{-1}\boldsymbol{\mathcal{T}}_i^t\mathcal{X} = \begin{bmatrix} \frac{1}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{I} & 0 & \frac{1}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{X}_{R,u} \\ 0 & 0 & 0 \\ \mathcal{X}_L\boldsymbol{\mathcal{T}}_i^t\mathcal{R}_1 & \mathcal{X}_L\boldsymbol{\mathcal{T}}_i^t\mathcal{R}_2 & \mathcal{X}_L\boldsymbol{\mathcal{T}}_i^t\mathcal{X}_R \end{bmatrix}, \tag{5.116}$$

where $\mathcal{I} \triangleq \mathbb{1}_K \otimes I_M$. It can also be verified that

$$\mathcal{X}^{-1}\mathcal{B}_l \overset{(5.68)}{=} \begin{bmatrix} \mathcal{L}_1^{\mathsf{T}} \\ \mathcal{L}_2^{\mathsf{T}} \\ \mathcal{X}_L \end{bmatrix} \begin{bmatrix} \overline{\mathcal{A}} \\ \mathcal{V}\overline{\mathcal{A}} \end{bmatrix} \overset{(5.70)}{=} \begin{bmatrix} \mathcal{I}^{\mathsf{T}}\overline{\mathcal{A}}/K \\ \mathcal{I}^{\mathsf{T}}\mathcal{V}\overline{\mathcal{A}}/K \\ \mathcal{X}_L\mathcal{B}_l \end{bmatrix} = \begin{bmatrix} \mathcal{I}^{\mathsf{T}}/K \\ 0 \\ \mathcal{X}_L\mathcal{B}_l \end{bmatrix}, \tag{5.117}$$

where the last equality holds because

$$\mathcal{I}^{\mathsf{T}}\overline{\mathcal{A}} = (\mathbb{1}_K^{\mathsf{T}}\overline{A}) \otimes I_M = \mathbb{1}_K^{\mathsf{T}} \otimes I_M = \mathcal{I}^{\mathsf{T}}, \tag{5.118}$$

$$\mathcal{I}^{\mathsf{T}}\mathcal{V}\overline{\mathcal{A}} = (\mathbb{1}_K^{\mathsf{T}}V\overline{A}) \otimes I_M \overset{(5.100)}{=} 0. \tag{5.119}$$

Substituting (5.116) and (5.117) into recursion (5.71), and also recalling the definition in (5.72), we get

$$\begin{bmatrix} \bar{\boldsymbol{x}}_{i+1}^t \\ \widehat{\boldsymbol{x}}_{i+1}^t \\ \check{\boldsymbol{x}}_{i+1}^t \end{bmatrix} = \begin{bmatrix} I_M - \frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{I} & 0 & -\frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{X}_{R,u} \\ 0 & I_M & 0 \\ -\mu\mathcal{X}_L\boldsymbol{\mathcal{T}}_i^t\mathcal{R}_1 & -\mu\mathcal{X}_L\boldsymbol{\mathcal{T}}_i^t\mathcal{R}_2 & \mathcal{D}_1 - \mu\mathcal{X}_L\boldsymbol{\mathcal{T}}_i^t\mathcal{X}_R \end{bmatrix} \cdot \begin{bmatrix} \bar{\boldsymbol{x}}_i^t \\ \widehat{\boldsymbol{x}}_i^t \\ \check{\boldsymbol{x}}_i^t \end{bmatrix} + \mu \begin{bmatrix} \frac{1}{K}\mathcal{I}^{\mathsf{T}} \\ 0 \\ \mathcal{X}_L\mathcal{B}_l \end{bmatrix} \boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t). \tag{5.120}$$

Notice that the second line of the above recursion is

$$\widehat{\boldsymbol{x}}_{i+1}^t = \widehat{\boldsymbol{x}}_i^t. \tag{5.121}$$

As a result, $\widehat{\boldsymbol{x}}_{i+1}^t$ will stay at 0 if the initial value $\widehat{\boldsymbol{x}}_0^0 = 0$. From (5.72) we can derive that

$$\widehat{\boldsymbol{x}}_0^0 \overset{(5.72)}{=} \mathcal{L}_2^{\mathsf{T}} \begin{bmatrix} \widetilde{\boldsymbol{\mathcal{W}}}_0^0 \\ \widetilde{\boldsymbol{\mathcal{Y}}}_0^0 \end{bmatrix} \overset{(3.43)}{=} \frac{1}{K}\mathcal{I}^{\mathsf{T}}(\mathcal{Y}_o - \boldsymbol{\mathcal{Y}}_0^0) \overset{(a)}{=} \frac{1}{K}\mathcal{I}^{\mathsf{T}}\mathcal{Y}_o \overset{(b)}{=} 0, \tag{5.122}$$

where equality (a) holds because $\boldsymbol{\mathcal{Y}}_0^0 = 0$. Equality (b) holds because $\mathcal{Y}_o$ lies in the range space of $\mathcal{V}$ (see Section 5.A.2) and $\mathcal{I}^{\mathsf{T}}\mathcal{V} = 0$ (see (5.100)). Therefore, with (5.121) and (5.122), we conclude that

$$\widehat{\boldsymbol{x}}_i^t = 0, \quad 0 \le i \le \bar{N} - 1,\ t \ge 0. \tag{5.123}$$

With (5.123), the transformed error recursion (3.47) reduces to

$$\begin{bmatrix} \bar{\boldsymbol{x}}_{i+1}^t \\ \check{\boldsymbol{x}}_{i+1}^t \end{bmatrix} = \begin{bmatrix} I_M - \frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{I} & -\frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{X}_{R,u} \\ -\mu\mathcal{X}_L\boldsymbol{\mathcal{T}}_i^t\mathcal{R}_1 & \mathcal{D}_1 - \mu\mathcal{X}_L\boldsymbol{\mathcal{T}}_i^t\mathcal{X}_R \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{x}}_i^t \\ \check{\boldsymbol{x}}_i^t \end{bmatrix} + \mu \begin{bmatrix} \frac{1}{K}\mathcal{I}^{\mathsf{T}} \\ \mathcal{X}_L\mathcal{B}_l \end{bmatrix} \boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t), \tag{5.124}$$

while (5.72) reduces to

$$\begin{bmatrix} \widetilde{\boldsymbol{\mathcal{W}}}_i^t \\ \widetilde{\boldsymbol{\mathcal{Y}}}_i^t \end{bmatrix} = \mathcal{X} \begin{bmatrix} \bar{\boldsymbol{x}}_i^t \\ 0_M \\ \check{\boldsymbol{x}}_i^t \end{bmatrix}. \tag{5.125}$$

5.E Proof of Lemma 5.2

Since $Q(w;x_n)$ is twice-differentiable, it follows from (5.5) that $\nabla_w^2 Q(w;x_n) \leq \delta I_M$ for $1 \leq n \leq N$, which in turn implies that

$$\nabla^2 J_k(w) = \frac{1}{N_k}\sum_{n=1}^{N_k} \nabla Q(w;x_{k,n}) \leq \delta I_M, \forall\ k \in \{1,\cdots,K\} \tag{5.126}$$

Moreover, since all $Q(w;x_n)$ are convex and at least one $Q(w;x_{n_o})$ is strongly convex (see equation (5.6), there must exist at least one node k_o such that

$$\nabla^2 J_{k_o}(w) = \frac{1}{N_{k_o}}\sum_{n=1}^{N_{k_o}} \nabla_w^2 Q(w;x_{k_o,n}) \geq \nu I_M, \tag{5.127}$$

which implies that the global risk function, $J(w)$, is ν-strongly convex as well. Substituting (5.126) and (5.127) into $\boldsymbol{H}_{k,i}^t$ defined in (2.102), for $t \geq 0$ and $0 \leq i \leq \bar{N}-1$ it holds that

$$\boldsymbol{H}_{k,i}^t \overset{(5.64)}{=} \int_0^1 \nabla^2 J_k\big(w^\star - r\widetilde{\boldsymbol{w}}_{k,i}^t\big)dr \overset{(5.126)}{\leq} \delta I_M, \forall k \in \{1,\cdots,K\} \tag{5.128}$$

$$\boldsymbol{H}_{k_o,i}^t \overset{(5.64)}{=} \int_0^1 \nabla^2 J_{k_o}\big(w^\star - r\widetilde{\boldsymbol{w}}_{k_o,i}^t\big)dr \overset{(5.127)}{\geq} \nu I_M, \tag{5.129}$$

$$\boldsymbol{\mathcal{H}}_i^t \overset{(5.64)}{=} \mathrm{diag}\{\boldsymbol{H}_{1,i}^t,\cdots,\boldsymbol{H}_{K,i}^t\} \overset{(5.128)}{\leq} \delta I_M. \tag{5.130}$$

Now we turn to derive the mean-square-error recursion. From the first line of error recursion (5.74), we have

$$\begin{aligned}\bar{\boldsymbol{x}}_{i+1}^t = &\Big(I_M - \frac{\mu}{K}\mathcal{I}^\mathsf{T}\boldsymbol{\mathcal{H}}_i^t\mathcal{I}\Big)\,\bar{\boldsymbol{x}}_i^t \\ &- \frac{\mu}{K}\left(\mathcal{I}^\mathsf{T}\boldsymbol{\mathcal{H}}_i^t\mathcal{X}_{R,u}\right)\check{\boldsymbol{x}}_i^t + \frac{\mu}{K}\mathcal{I}^\mathsf{T}\boldsymbol{s}(\boldsymbol{\mathcal{w}}_i^t).\end{aligned} \tag{5.131}$$

Recalling that $\mathcal{I} = \mathbb{1}_K \otimes I_M$, it holds that

$$\frac{1}{K}\mathcal{I}^\mathsf{T}\boldsymbol{\mathcal{H}}_i^t\mathcal{I} = \frac{1}{K}\sum_{k=1}^K \boldsymbol{H}_{k,i}^t. \tag{5.132}$$

Substituting relations (5.128) and (5.129) into (5.132), it holds that

$$\frac{\nu}{K}I_M \leq \frac{1}{K}\mathcal{I}^\mathsf{T}\boldsymbol{\mathcal{H}}_i^t\mathcal{I} \leq \delta I_M, \tag{5.133}$$

which also implies that

$$\begin{aligned}\left\|I_M - \frac{\mu}{K}\mathcal{I}^\mathsf{T}\boldsymbol{\mathcal{H}}_i^t\mathcal{I}\right\|^2 &\leq \max\left\{\left(1-\frac{\mu\nu}{K}\right)^2, (1-\mu\delta)^2\right\} \\ &\leq \left(1-\frac{\mu\nu}{K}\right)^2,\end{aligned} \tag{5.134}$$

where the last inequality holds when the step-size μ is small enough so that

$$\mu < 1/\delta. \tag{5.135}$$

Now we square both sides of equation (5.131) and reach

$$\begin{aligned}
&\|\bar{\boldsymbol{x}}_{i+1}^t\|^2 \\
&= \left\| \Big(I_M - \frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{I}\Big)\bar{\boldsymbol{x}}_i^t - \frac{\mu}{K}\big(\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{X}_{R,u}\big)\,\check{\boldsymbol{x}}_i^t + \frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\right\|^2 \\
&\overset{(a)}{=} \left\| (1-t)\frac{1}{1-t}\left(I_M - \frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{I}\right)\bar{\boldsymbol{x}}_i^t \right. \\
&\qquad \left. + t\frac{1}{t}\Big[-\frac{\mu}{K}\left(\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{X}_{R,u}\right)\check{\boldsymbol{x}}_i^t + \frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\Big]\right\|^2 \\
&\overset{(b)}{\le} \frac{1}{1-t}\left\| I_M - \frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{I}\right\|^2 \|\bar{\boldsymbol{x}}_i^t\|^2 \\
&\qquad + \frac{1}{t}\left\| \frac{\mu}{K}\left(\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{X}_{R,u}\right)\check{\boldsymbol{x}}_i^t + \frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\right\|^2 \\
&\overset{(c)}{\le} \frac{1}{1-t}\left\| I_M - \frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{I}\right\|^2 \|\bar{\boldsymbol{x}}_i^t\|^2 \\
&\qquad + \frac{2\mu^2}{tK^2}\left\|\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{X}_{R,u}\right\|^2\|\check{\boldsymbol{x}}_i^t\|^2 + \frac{2\mu^2}{tK^2}\|\mathcal{I}^{\mathsf{T}}\|^2\left\|\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\right\|^2 \\
&\overset{(d)}{\le} \frac{1}{1-t}\left(1 - \frac{\mu\nu}{K}\right)^2\|\bar{\boldsymbol{x}}_i^t\|^2 \\
&\qquad + \frac{2\mu^2\delta^2\|\mathcal{X}_{R,u}\|^2}{Kt}\|\check{\boldsymbol{x}}_i^t\|^2 + \frac{2\mu^2}{Kt}\left\|\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\right\|^2 \\
&\overset{(e)}{=} \left(1 - \frac{\mu\nu}{K}\right)\|\bar{\boldsymbol{x}}_i^t\|^2 + \frac{2\mu\delta^2\|\mathcal{X}_{R,u}\|^2}{\nu}\|\check{\boldsymbol{x}}_i^t\|^2 + \frac{2\mu}{\nu}\left\|\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\right\|^2
\end{aligned} \tag{5.136}$$

where equality (a) holds for any constant $t \in (0,1)$, inequality (b) holds because of the Jensen's inequality, inequality (c) holds because $\|a+b\|^2 \le 2\|a\|^2 + 2\|b\|^2$ for any two vectors a and b, and inequality (d) holds because of relation (5.134) and

$$\|\mathcal{I}^{\mathsf{T}}\|^2 = K, \tag{5.137}$$

$$\left\|\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{X}_{R,u}\right\|^2 \le \|\mathcal{I}^{\mathsf{T}}\|^2\|\boldsymbol{\mathcal{H}}_i^t\|^2\|\mathcal{X}_{R,u}\|^2 \le K\delta^2\|\mathcal{X}_{R,u}\|^2. \tag{5.138}$$

Equality (e) holds when $t = \mu\nu/K$.

Next we turn to the second line of recursion (5.74):

$$\check{\boldsymbol{x}}_{i+1}^t = \mathcal{D}_1\check{\boldsymbol{x}}_i^t - \mu\Big(\mathcal{X}_L\boldsymbol{\mathcal{T}}_i^t\mathcal{R}_1\bar{\boldsymbol{x}}_i^t + \mathcal{X}_L\boldsymbol{\mathcal{T}}_i^t\mathcal{X}_R\check{\boldsymbol{x}}_i^t - \mathcal{X}_L\mathcal{B}_l\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\Big) \tag{5.139}$$

By squaring and applying Jensen's inequality, we have

$$\|\check{\boldsymbol{x}}_{i+1}^t\|^2 \leq \frac{1}{t}\|\mathcal{D}_1\|^2\|\check{\boldsymbol{x}}_i^t\|^2+\frac{3\mu^2}{1-t}\Big(\|\mathcal{X}_L\boldsymbol{\mathcal{T}}_i^t\mathcal{R}_1\|^2\|\bar{\boldsymbol{x}}_i^t\|^2 + \|\mathcal{X}_L\boldsymbol{\mathcal{T}}_i^t\mathcal{X}_R\|^2\|\check{\boldsymbol{x}}_i^t\|^2+\|\mathcal{X}_L\mathcal{B}_l\|^2\|\boldsymbol{s}(\boldsymbol{\mathcal{w}}_i^t)\|^2\Big) \tag{5.140}$$

for any constant $t \in (0,1)$. From the definition of $\boldsymbol{\mathcal{T}}_i^t$ in (3.28) and recalling from (5.111) that $\overline{\mathcal{A}}\mathcal{V} = \mathcal{V}\overline{\mathcal{A}}$, we have

$$\boldsymbol{\mathcal{T}}_i^t = \begin{bmatrix} \overline{\mathcal{A}} & 0 \\ 0 & \overline{\mathcal{A}} \end{bmatrix}\begin{bmatrix} I_{KM} & 0 \\ \mathcal{V} & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{\mathcal{H}}_i^t & 0 \\ 0 & \boldsymbol{\mathcal{H}}_i^t \end{bmatrix}. \tag{5.141}$$

It can also be verified that

$$\begin{aligned}\left\|\begin{bmatrix} I_{KM} & 0 \\ \mathcal{V} & 0 \end{bmatrix}\right\|^2 &= \lambda_{\max}\left(\begin{bmatrix} I_{KM} & 0 \\ \mathcal{V} & 0 \end{bmatrix}^{\mathsf{T}}\begin{bmatrix} I_{KM} & 0 \\ \mathcal{V} & 0 \end{bmatrix}\right)\\ &= \lambda_{\max}\left(\begin{bmatrix} I_{KM}+\mathcal{V}^2 & 0 \\ 0 & 0 \end{bmatrix}\right)\\ &= \lambda_{\max}\left(I_{KM}+\frac{I_{KM}-\overline{\mathcal{A}}}{2K}\right) \leq 2\end{aligned} \tag{5.142}$$

where the last inequality holds because $0 < \lambda(\overline{\mathcal{A}}) \leq 1$. With (5.141), (5.142) and the facts that $\lambda_{\max}(\overline{\mathcal{A}}) = 1$, $\lambda_{\max}(\boldsymbol{\mathcal{H}}_i^t) \leq \delta$, we conclude that

$$\|\boldsymbol{\mathcal{T}}_i^t\|^2 \leq \left\|\begin{bmatrix} \overline{\mathcal{A}} & 0 \\ 0 & \overline{\mathcal{A}} \end{bmatrix}\right\|^2\left\|\begin{bmatrix} I_{KM} & 0 \\ \mathcal{V} & 0 \end{bmatrix}\right\|^2\left\|\begin{bmatrix} \boldsymbol{\mathcal{H}}_i^t & 0 \\ 0 & \boldsymbol{\mathcal{H}}_i^t \end{bmatrix}\right\|^2 \leq 2\delta^2. \tag{5.143}$$

Similarly, using $\overline{\mathcal{A}}\mathcal{V} = \mathcal{V}\overline{\mathcal{A}}$ we can rewrite $\mathcal{B}_l$ defined in (3.28) as

$$\mathcal{B}_l = \begin{bmatrix} \overline{\mathcal{A}} & 0 \\ 0 & \overline{\mathcal{A}} \end{bmatrix}\begin{bmatrix} I_{KM} \\ \mathcal{V} \end{bmatrix}, \tag{5.144}$$

and it can be verified that

$$\begin{aligned}\left\|\begin{bmatrix} I_{KM} \\ \mathcal{V} \end{bmatrix}\right\|^2 &= \lambda_{\max}\left(\begin{bmatrix} I_{KM} \\ \mathcal{V} \end{bmatrix}^{\mathsf{T}}\begin{bmatrix} I_{KM} \\ \mathcal{V} \end{bmatrix}\right)\\ &= \lambda_{\max}\left(I_{KM}+\mathcal{V}^2\right)\\ &= \lambda_{\max}\left(I_{KM}+\frac{I_{KM}-\overline{\mathcal{A}}}{2K}\right) \leq 2.\end{aligned} \tag{5.145}$$

As a result,

$$\|\mathcal{B}_l\|^2 \leq \left\| \begin{bmatrix} \overline{\mathcal{A}} & 0 \\ 0 & \overline{\mathcal{A}} \end{bmatrix} \right\|^2 \left\| \begin{bmatrix} I_{KM} \\ \mathcal{V} \end{bmatrix} \right\|^2 \leq 2. \tag{5.146}$$

Furthermore,

$$\begin{aligned} \|\mathcal{R}_1\|^2 &= \left\| \begin{bmatrix} \mathbb{1}_K \\ 0 \end{bmatrix} \otimes I_M \right\|^2 \\ &= \lambda_{\max} \left(\begin{bmatrix} \mathbb{1}_K \\ 0 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mathbb{1}_K \\ 0 \end{bmatrix} \otimes I_M \right) = K. \end{aligned} \tag{5.147}$$

With (5.143)–(5.147), we have

$$\|\mathcal{X}_L \boldsymbol{\mathcal{T}}_i^t \mathcal{R}_1\|^2 \leq \|\mathcal{X}_L\|^2 \|\boldsymbol{\mathcal{T}}_i^t\|^2 \|\mathcal{R}_1\|^2 \leq 2K\delta^2 \|\mathcal{X}_L\|^2, \tag{5.148}$$

$$\|\mathcal{X}_L \boldsymbol{\mathcal{T}}_i^t \mathcal{X}_R\|^2 \leq 2\delta^2 \|\mathcal{X}_L\|^2 \|\mathcal{X}_R\|^2, \tag{5.149}$$

$$\|\mathcal{X}_L \mathcal{B}_l\|^2 \leq 2\|\mathcal{X}_L\|^2. \tag{5.150}$$

Substituting (5.148) into (5.140) and recalling that $\|\mathcal{D}_1\| = \lambda < 1$, we have

$$\begin{aligned} &\|\check{\boldsymbol{\mathcal{X}}}_{i+1}^t\|^2 \\ &\leq \frac{1}{t}\lambda^2 \|\check{\boldsymbol{\mathcal{X}}}_i^t\|^2 + \frac{3\mu^2}{1-t}\Big(2K\delta^2 \|\mathcal{X}_L\|^2 \|\bar{\boldsymbol{\mathcal{X}}}_i^t\|^2 \\ &\qquad + 2\delta^2 \|\mathcal{X}_L\|^2 \|\mathcal{X}_R\|^2 \|\check{\boldsymbol{\mathcal{X}}}_i^t\|^2 + 2\|\mathcal{X}_L\|^2 \|\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{W}}_i^t)\|^2\Big) \\ &= \left(\lambda + \frac{6\mu^2\delta^2 \|\mathcal{X}_L\|^2 \|\mathcal{X}_R\|^2}{1-\lambda}\right) \|\check{\boldsymbol{\mathcal{X}}}_i^t\|^2 \\ &\qquad + \frac{6K\mu^2\delta^2 \|\mathcal{X}_L\|^2}{1-\lambda} \|\bar{\boldsymbol{\mathcal{X}}}_i^t\|^2 + \frac{6\|\mathcal{X}_L\|^2 \mu^2}{1-\lambda} \|\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{W}}_i^t)\|^2, \end{aligned} \tag{5.151}$$

where the last equality holds by setting $t = \lambda$. If we let

$$\begin{aligned} &a_1 = 1/K,\ a_2 = \|\mathcal{X}_{R,u}\|^2,\ a_3 = \frac{6\|\mathcal{X}_L\|^2 \|\mathcal{X}_R\|^2}{1-\lambda}, \\ &a_4 = \frac{6K\|\mathcal{X}_L\|^2}{1-\lambda},\ a_5 = \frac{6\|\mathcal{X}_L\|^2}{1-\lambda} \end{aligned} \tag{5.152}$$

and take expectations of inequalities (5.140) and (5.151), we arrive at recursion (3.164), where $a_l,\ 1 \leq l \leq 5$ are positive constants that are independent of $\bar{N}$, δ and ν.

5.F Proof of Lemma 5.3

We first introduce the gradient noise at node k:

$$\boldsymbol{s}_k(\boldsymbol{w}_{k,i}^t) \triangleq \widehat{\nabla J}_k(\boldsymbol{w}_{k,i}^t) - \nabla J_k(\boldsymbol{w}_{k,i}^t). \tag{5.153}$$

With (5.153) and (5.57), we have

$$\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t) = \mathrm{col}\{\boldsymbol{s}_1(\boldsymbol{w}_{1,i}^t), \boldsymbol{s}_2(\boldsymbol{w}_{2,i}^t), \cdots, \boldsymbol{s}_N(\boldsymbol{w}_{N,i}^t)\}. \tag{5.154}$$

Now we bound the term $\|\boldsymbol{s}_k(\boldsymbol{w}_{k,i}^t)\|^2$. Note that

$$\begin{aligned}
&\boldsymbol{s}_k(\boldsymbol{w}_{k,i}^t) \\
&= \widehat{\nabla J}_k(\boldsymbol{w}_{k,i}^t) - \nabla J_k(\boldsymbol{w}_{k,i}^t) \\
&\overset{(5.15)}{=} \nabla Q(\boldsymbol{w}_{k,i}^t; x_{k,\boldsymbol{n}_{k,i}^t}) - \nabla Q(\boldsymbol{w}_{k,0}^t; x_{k,\boldsymbol{n}_{k,i}^t}) + \boldsymbol{g}_k^t - \nabla J_k(\boldsymbol{w}_{k,i}^t) \\
&\overset{(5.16)}{=} \nabla Q(\boldsymbol{w}_{k,i}^t; x_{k,\boldsymbol{n}_{k,i}^t}) - \nabla Q(\boldsymbol{w}_{k,0}^t; x_{k,\boldsymbol{n}_{k,i}^t}) \\
&\quad + \frac{1}{\bar{N}} \sum_{j=0}^{\bar{N}-1} \nabla Q\left(\boldsymbol{w}_{k,j}^{t-1}; x_{k,\boldsymbol{n}_{k,j}^{t-1}}\right) - \frac{1}{\bar{N}} \sum_{n=1}^{\bar{N}} \nabla Q\left(\boldsymbol{w}_{k,i}^t; x_{k,n}\right)
\end{aligned} \tag{5.155}$$

Since $\boldsymbol{n}_{k,j}^{t-1} = \boldsymbol{\sigma}^{t-1}(j+1)$ is sampled by random reshuffling without replacement, it holds that

$$\begin{aligned}
\sum_{j=0}^{\bar{N}-1} \nabla Q\left(\boldsymbol{w}_{k,\bar{N}}^{t-1}; x_{k,\boldsymbol{n}_{k,j}^{t-1}}\right) &= \sum_{n=1}^{\bar{N}} \nabla Q\left(\boldsymbol{w}_{k,\bar{N}}^{t-1}; x_{k,n}\right) \\
&\overset{(a)}{=} \sum_{n=1}^{\bar{N}} \nabla Q\left(\boldsymbol{w}_{k,0}^t; x_{k,n}\right)
\end{aligned} \tag{5.156}$$

where equality (a) holds because $\boldsymbol{w}_{k,0}^t = \boldsymbol{w}_{k,\bar{N}}^{t-1}$. With relation (5.156), we can rewrite (5.155) as

$$\begin{aligned}
&\boldsymbol{s}_k(\boldsymbol{w}_{k,i}^t) \\
&= \nabla Q(\boldsymbol{w}_{k,i}^t; x_{k,\boldsymbol{n}_{k,i}^t}) - \nabla Q(\boldsymbol{w}_{k,0}^t; x_{k,\boldsymbol{n}_{k,i}^t}) \\
&\quad + \frac{1}{\bar{N}} \sum_{j=0}^{\bar{N}-1} \nabla Q\left(\boldsymbol{w}_{k,j}^{t-1}; x_{k,\boldsymbol{n}_{k,j}^{t-1}}\right) - \frac{1}{\bar{N}} \sum_{j=0}^{\bar{N}-1} \nabla Q\left(\boldsymbol{w}_{k,\bar{N}}^{t-1}; x_{k,\boldsymbol{n}_{k,j}^{t-1}}\right) \\
&\quad + \frac{1}{\bar{N}} \sum_{n=1}^{\bar{N}} \nabla Q\left(\boldsymbol{w}_{k,0}^t; x_{k,n}\right) - \frac{1}{\bar{N}} \sum_{n=1}^{\bar{N}} \nabla Q\left(\boldsymbol{w}_{k,i}^t; x_{k,n}\right)
\end{aligned} \tag{5.157}$$

By squaring and applying Jensen's inequality, we have

$$
\begin{aligned}
&\|\boldsymbol{s}_k(\boldsymbol{w}_{k,i}^t)\|^2 \\
&\le 3\left\|\nabla Q(\boldsymbol{w}_{k,i}^t; x_{k,\boldsymbol{n}_{k,i}^t}) - \nabla Q(\boldsymbol{w}_{k,0}^t; x_{k,\boldsymbol{n}_{k,i}^t})\right\|^2 \\
&\quad + \frac{3}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\left\|\nabla Q\left(\boldsymbol{w}_{k,j}^{t-1}; x_{k,\boldsymbol{n}_{k,j}^{t-1}}\right) - \nabla Q\left(\boldsymbol{w}_{k,\bar{N}}^{t-1}; x_{k,\boldsymbol{n}_{k,j}^{t-1}}\right)\right\|^2 \\
&\quad + \frac{3}{\bar{N}}\sum_{n=1}^{\bar{N}}\left\|\nabla Q\left(\boldsymbol{w}_{k,0}^t; x_{k,n}\right) - \nabla Q\left(\boldsymbol{w}_{k,i}^t; x_{k,n}\right)\right\|^2 \\
&\le 6\delta^2\|\boldsymbol{w}_{k,i}^t - \boldsymbol{w}_{k,0}^t\|^2 + \frac{3\delta^2}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\left\|\boldsymbol{w}_{k,j}^{t-1} - \boldsymbol{w}_{k,\bar{N}}^{t-1}\right\|^2
\end{aligned}
\tag{5.158}
$$

where the last inequality holds because of the Lipschitz inequality (5.5) in Assumption 1. Consequently,

$$
\begin{aligned}
&\|\boldsymbol{s}(\boldsymbol{\mathcal{w}}_i^t)\|^2 \\
&\overset{(5.154)}{=} \sum_{k=1}^{K}\|\boldsymbol{s}_k(\boldsymbol{w}_{k,i}^t)\|^2 \\
&\le 6\delta^2\sum_{k=1}^{K}\|\boldsymbol{w}_{k,i}^t - \boldsymbol{w}_{k,0}^t\|^2 + \frac{3\delta^2}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\sum_{k=1}^{K}\left\|\boldsymbol{w}_{k,j}^{t-1} - \boldsymbol{w}_{k,\bar{N}}^{t-1}\right\|^2 \\
&= 6\delta^2\|\boldsymbol{\mathcal{w}}_i^t - \boldsymbol{\mathcal{w}}_0^t\|^2 + \frac{3\delta^2}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\left\|\boldsymbol{\mathcal{w}}_j^{t-1} - \boldsymbol{\mathcal{w}}_{\bar{N}}^{t-1}\right\|^2 \\
&= 6\delta^2\|\widetilde{\boldsymbol{\mathcal{w}}}_i^t - \widetilde{\boldsymbol{\mathcal{w}}}_0^t\|^2 + \frac{3\delta^2}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\left\|\widetilde{\boldsymbol{\mathcal{w}}}_j^{t-1} - \widetilde{\boldsymbol{\mathcal{w}}}_{\bar{N}}^{t-1}\right\|^2 \\
&\le 6\delta^2\big(\|\widetilde{\boldsymbol{\mathcal{w}}}_i^t - \widetilde{\boldsymbol{\mathcal{w}}}_0^t\|^2 + \|\widetilde{\boldsymbol{\mathcal{y}}}_i^t - \widetilde{\boldsymbol{\mathcal{y}}}_0^t\|^2\big) \\
&\quad + \frac{3\delta^2}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\left(\left\|\widetilde{\boldsymbol{\mathcal{w}}}_j^{t-1} - \widetilde{\boldsymbol{\mathcal{w}}}_{\bar{N}}^{t-1}\right\|^2 + \left\|\widetilde{\boldsymbol{\mathcal{y}}}_j^{t-1} - \widetilde{\boldsymbol{\mathcal{y}}}_{\bar{N}}^{t-1}\right\|^2\right).
\end{aligned}
\tag{5.159}
$$

Now note that

$$\begin{aligned}
&\|\widetilde{\boldsymbol{\mathcal{W}}}_i^t - \widetilde{\boldsymbol{\mathcal{W}}}_0^t\|^2 + \|\widetilde{\boldsymbol{\mathcal{Y}}}_i^t - \widetilde{\boldsymbol{\mathcal{Y}}}_0^t\|^2 \\
&= \left\| \begin{bmatrix} \widetilde{\boldsymbol{\mathcal{W}}}_i^t \\ \widetilde{\boldsymbol{\mathcal{Y}}}_i^t \end{bmatrix} - \begin{bmatrix} \widetilde{\boldsymbol{\mathcal{W}}}_0^t \\ \widetilde{\boldsymbol{\mathcal{Y}}}_0^t \end{bmatrix} \right\|^2 \overset{(5.75)}{\leq} \|\mathcal{X}\|^2 \left\| \begin{bmatrix} \bar{\boldsymbol{\mathcal{X}}}_i^t \\ 0_M \\ \check{\boldsymbol{\mathcal{X}}}_i^t \end{bmatrix} - \begin{bmatrix} \bar{\boldsymbol{\mathcal{X}}}_0^t \\ 0_M \\ \check{\boldsymbol{\mathcal{X}}}_0^t \end{bmatrix} \right\|^2 \\
&= \|\mathcal{X}\|^2 \left(\|\bar{\boldsymbol{\mathcal{X}}}_i^t - \bar{\boldsymbol{\mathcal{X}}}_0^t\|^2 + \|\check{\boldsymbol{\mathcal{X}}}_i^t - \check{\boldsymbol{\mathcal{X}}}_0^t\|^2 \right) \\
&\leq \|\mathcal{X}\|^2 \|\bar{\boldsymbol{\mathcal{X}}}_i^t - \bar{\boldsymbol{\mathcal{X}}}_0^t\|^2 + 2\|\mathcal{X}\|^2 \|\check{\boldsymbol{\mathcal{X}}}_i^t\|^2 + 2\|\mathcal{X}\|^2 \|\check{\boldsymbol{\mathcal{X}}}_0^t\|^2 \qquad (5.160)
\end{aligned}$$

Similarly, it holds that

$$\begin{aligned}
&\left\|\widetilde{\boldsymbol{\mathcal{W}}}_j^{t-1} - \widetilde{\boldsymbol{\mathcal{W}}}_{\bar{N}}^{t-1}\right\|^2 + \left\|\widetilde{\boldsymbol{\mathcal{Y}}}_j^{t-1} - \widetilde{\boldsymbol{\mathcal{Y}}}_{\bar{N}}^{t-1}\right\|^2 \\
&\leq \|\mathcal{X}\|^2 \|\bar{\boldsymbol{\mathcal{X}}}_j^{t-1} - \bar{\boldsymbol{\mathcal{X}}}_{\bar{N}}^{t-1}\|^2 + 2\|\mathcal{X}\|^2 \|\check{\boldsymbol{\mathcal{X}}}_j^{t-1}\|^2 + 2\|\mathcal{X}\|^2 \|\check{\boldsymbol{\mathcal{X}}}_0^t\|^2. \qquad (5.161)
\end{aligned}$$

Substituting (5.160) and (5.161) into (5.159) and letting $b = \|\mathcal{X}\|^2$, we have

$$\begin{aligned}
\|\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\|^2 &\leq 6b\delta^2 \|\bar{\boldsymbol{\mathcal{X}}}_i^t - \bar{\boldsymbol{\mathcal{X}}}_0^t\|^2 + 12b\delta^2 \|\check{\boldsymbol{\mathcal{X}}}_i^t\|^2 + 18b\delta^2 \|\check{\boldsymbol{\mathcal{X}}}_0^t\|^2 \\
&\quad + \frac{3b\delta^2}{\bar{N}} \sum_{j=0}^{\bar{N}-1} \|\bar{\boldsymbol{\mathcal{X}}}_j^{t-1} - \bar{\boldsymbol{\mathcal{X}}}_{\bar{N}}^{t-1}\|^2 + \frac{6b\delta^2}{\bar{N}} \sum_{j=0}^{\bar{N}-1} \|\check{\boldsymbol{\mathcal{X}}}_j^{t-1}\|^2 \qquad (5.162)
\end{aligned}$$

By taking expectations, we achieve inequality (5.77).

5.G Proof of Lemma 5.4

It is established in Lemma 5.2 that when step-size μ satisfies

$$\mu < \frac{1}{\delta}, \qquad (5.163)$$

the dynamic system (3.164) holds. Using Jensen's inequality, the second line of (3.164) becomes

$$\begin{aligned}
&\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_{i+1}^t\|^2 \\
&\leq \left(\lambda+a_3\mu^2\delta^2\right)\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_i^t\|^2+2a_4\mu^2\delta^2\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_i^t-\bar{\boldsymbol{\mathcal{X}}}_0^t\|^2 \\
&\quad+2a_4\mu^2\delta^2\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^t\|^2+a_5\mu^2\mathbb{E}\|\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{W}}_i^t)\|^2 \\
&\overset{(5.77)}{\leq} \Big(\lambda+(a_3+12a_5b)\mu^2\delta^2\Big)\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_i^t\|^2 \\
&\quad+(2a_4+6a_5b)\mu^2\delta^2\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_i^t-\bar{\boldsymbol{\mathcal{X}}}_0^t\|^2+2a_4\mu^2\delta^2\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^t\|^2 \\
&\quad+18a_5b\mu^2\delta^2\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^t\|^2+\frac{3a_5b\mu^2\delta^2}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_j^{t-1}-\bar{\boldsymbol{\mathcal{X}}}_{\bar{N}}^{t-1}\|^2 \\
&\quad+\frac{6a_5b\mu^2\delta^2}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_j^{t-1}\|^2. \qquad (5.164)
\end{aligned}$$

Now we let $\lambda_1=(1+\lambda)/2<1$. It can be verified that when the step-size μ is small enough so that

$$\mu\leq\sqrt{\frac{1-\lambda}{2(a_3+12a_5b)\delta^2}}, \qquad (5.165)$$

it holds that

$$\lambda+(a_3+12a_5b)\mu^2\delta^2\leq\lambda_1<1. \qquad (5.166)$$

Substituting (5.166) into (5.164), we have

$$\begin{aligned}
&\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_{i+1}^t\|^2 \\
&\leq \lambda_1\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_i^t\|^2+(2a_4+6a_5b)\mu^2\delta^2\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_i^t-\bar{\boldsymbol{\mathcal{X}}}_0^t\|^2 \\
&\quad+2a_4\mu^2\delta^2\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^t\|^2+18a_5b\mu^2\delta^2\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^t\|^2 \\
&\quad+\frac{3a_5b\mu^2\delta^2}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_j^{t-1}-\bar{\boldsymbol{\mathcal{X}}}_{\bar{N}}^{t-1}\|^2+\frac{6a_5b\mu^2\delta^2}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_j^{t-1}\|^2. \qquad (5.167)
\end{aligned}$$

Iterating (5.167), for $0 \leq i \leq \bar{N}-1$, we get

$$
\begin{aligned}
&\mathbb{E}\|\check{\boldsymbol{x}}_{i+1}^t\|^2 \\
&\leq \lambda_1^{i+1}\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2 + (2a_4+6a_5b)\mu^2\delta^2\sum_{j=0}^{i}\lambda_1^{i-j}\mathbb{E}\|\bar{\boldsymbol{x}}_j^t-\bar{\boldsymbol{x}}_0^t\|^2 \\
&\quad + \Big(2a_4\mu^2\delta^2\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 + 18a_5b\mu^2\delta^2\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2\Big)\sum_{j=0}^{i}\lambda_1^{i-j} \\
&\quad + \Bigg(\frac{3a_5b\mu^2\delta^2}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1}-\bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2 \\
&\qquad + \frac{6a_5b\mu^2\delta^2}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\Bigg)\sum_{j=0}^{i}\lambda_1^{i-j} \\
&\overset{(a)}{\leq} \lambda_1^{i+1}\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2 + (2a_4+6a_5b)\mu^2\delta^2\sum_{j=0}^{i}\mathbb{E}\|\bar{\boldsymbol{x}}_j^t-\bar{\boldsymbol{x}}_0^t\|^2 \\
&\quad + 2a_4\mu^2\delta^2(i+1)\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 + 18a_5b\mu^2\delta^2(i+1)\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2 \\
&\quad + \frac{3a_5b\mu^2\delta^2(i+1)}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1}-\bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2 \\
&\quad + \frac{6a_5b\mu^2\delta^2(i+1)}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2 \\
&= \Big(\lambda_1^{i+1} + 18a_5b\mu^2\delta^2(i+1)\Big)\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2 \\
&\quad + (2a_4+6a_5b)\mu^2\delta^2\sum_{j=0}^{i}\mathbb{E}\|\bar{\boldsymbol{x}}_j^t-\bar{\boldsymbol{x}}_0^t\|^2 \\
&\quad + 2a_4\mu^2\delta^2(i+1)\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 \\
&\quad + \frac{3a_5b\mu^2\delta^2(i+1)}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1}-\bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2 \\
&\quad + \frac{6a_5b\mu^2\delta^2(i+1)}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2, \qquad (5.168)
\end{aligned}
$$

where (a) holds because $\lambda_1 < 1$ and hence $\sum_{j=0}^{i}\lambda_1^{i-j} \leq i+1$. Next we let $\lambda_2 = (1+\lambda_1)/2 < 1$. If the step-size μ is chosen small enough such that

$$
\lambda_1^{i+1} + 2a_4\mu^2\delta^2(i+1) \leq \lambda_2, \quad \forall\ i = 0, \cdots, \bar{N}-1 \qquad (5.169)
$$

then it follows that

$$
\begin{aligned}
&\mathbb{E}\|\check{\boldsymbol{x}}_{i+1}^t\|^2 \\
&\leq \lambda_2 \mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2 + (2a_4 + 6a_5 b)\mu^2\delta^2 \sum_{j=0}^{i} \mathbb{E}\|\bar{\boldsymbol{x}}_j^t - \bar{\boldsymbol{x}}_0^t\|^2 \\
&\quad + 2a_4\mu^2\delta^2(i+1)\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 \\
&\quad + \frac{3a_5 b\mu^2\delta^2(i+1)}{\bar{N}} \sum_{j=0}^{\bar{N}-1} \mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1} - \bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2 \\
&\quad + \frac{6a_5 b\mu^2\delta^2(i+1)}{\bar{N}} \sum_{j=0}^{\bar{N}-1} \mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2 \\
&\leq \lambda_2 \mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2 + (2a_4 + 6a_5 b)\mu^2\delta^2 \sum_{j=0}^{\bar{N}-1} \mathbb{E}\|\bar{\boldsymbol{x}}_j^t - \bar{\boldsymbol{x}}_0^t\|^2 \\
&\quad + 2a_4\mu^2\delta^2\bar{N}\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 \\
&\quad + 3a_5 b\mu^2\delta^2\bar{N}\left(\frac{1}{\bar{N}} \sum_{j=0}^{\bar{N}-1} \mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1} - \bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2\right) \\
&\quad + 6a_5 b\mu^2\delta^2\bar{N}\left(\frac{1}{\bar{N}} \sum_{j=0}^{\bar{N}-1} \mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\right), \quad \forall\ i = 0, \cdots, \bar{N}-1
\end{aligned} \tag{5.170}
$$

Notice that

$$
\lambda_1^{i+1} + 2a_4\mu^2\delta^2(i+1) \leq \lambda_1 + 2a_4\mu^2\delta^2\bar{N},\ \forall i = 0, \cdots, \bar{N}-1. \tag{5.171}
$$

Therefore, to guarantee (5.169), it is enough to set

$$
\lambda_1 + 2a_4\mu^2\delta^2\bar{N} \leq \lambda_2 \Longleftrightarrow \mu \leq \sqrt{\frac{\lambda_2 - \lambda_1}{2a_4\delta^2\bar{N}}}. \tag{5.172}
$$

From (5.170) we can derive

$$\begin{aligned}
&\sum_{i=1}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_i^t\|^2\\
&\quad\le \lambda_2(\bar{N}-1)\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2\\
&\qquad+(2a_4+6a_5b)\mu^2\delta^2(\bar{N}-1)\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^t-\bar{\boldsymbol{x}}_0^t\|^2\\
&\qquad+2a_4\mu^2\delta^2\bar{N}(\bar{N}-1)\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2\\
&\qquad+3a_5b\mu^2\delta^2\bar{N}(\bar{N}-1)\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1}-\bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2\right)\\
&\qquad+6a_5b\mu^2\delta^2\bar{N}(\bar{N}-1)\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\right).
\end{aligned}\tag{5.173}$$

As a result,

$$\begin{aligned}
&\frac{1}{\bar{N}}\sum_{i=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_i^t\|^2\\
&\quad=\frac{1}{\bar{N}}\left(\sum_{i=1}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_i^t\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2\right)\\
&\quad\le\frac{\lambda_2(\bar{N}-1)+1}{\bar{N}}\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2\\
&\qquad+(2a_4+6a_5b)\mu^2\delta^2\bar{N}\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^t-\bar{\boldsymbol{x}}_0^t\|^2\right)\\
&\qquad+2a_4\mu^2\delta^2\bar{N}\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2\\
&\qquad+3a_5b\mu^2\delta^2\bar{N}\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1}-\bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2\right)\\
&\qquad+6a_5b\mu^2\delta^2\bar{N}\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\right).
\end{aligned}\tag{5.174}$$

To simplify the notation, we let

$$\begin{aligned}
&\lambda_3=\frac{\lambda_2(\bar{N}-1)+1}{\bar{N}},\\
&c_1=2a_4,\ c_2=2a_4+6a_5b,\ c_3=3a_5b,\ c_4=6a_5b.
\end{aligned}\tag{5.175}$$

Using $\lambda_2<1$, we have

$$\lambda_3=\frac{\lambda_2(\bar{N}-1)+1}{\bar{N}}<\frac{\bar{N}-1+1}{\bar{N}}=1.\tag{5.176}$$

In summary, when μ satisfies (5.163), (5.165) and (5.172), i.e.

$$\mu \leq \min\left\{\frac{1}{\delta}, \sqrt{\frac{1-\lambda}{2(a_3+12a_5b)\delta^2}}, \sqrt{\frac{\lambda_2-\lambda_1}{2a_4\delta^2\bar{N}}}\right\}, \tag{5.177}$$

we conclude recursion (5.82). To get a simple form for the step-size, with $\lambda_2-\lambda_1=(1-\lambda)/4$ we can further restrict μ as

$$\begin{aligned}\mu &\leq \min\left\{1, \sqrt{\frac{1}{2(a_3+12a_5b)}}, \sqrt{\frac{1}{8a_4}}\right\}\sqrt{\frac{1-\lambda}{\delta^2\bar{N}}}\\ &\triangleq C_1\sqrt{\frac{1-\lambda}{\delta^2\bar{N}}}.\end{aligned} \tag{5.178}$$

It is obvious that all step-sizes within the range defined in (5.178) will also satisfy (5.177). Moreover, recursion (5.83) holds by setting $i=\bar{N}-1$ in (5.170).

5.H Proof of Lemma 5.5

Substituting (5.77) into the first line of (3.164), we have

$$\begin{aligned}&\mathbb{E}\|\bar{\boldsymbol{x}}_{i+1}^t\|^2\\ &\leq (1-a_1\mu\nu)\mathbb{E}\|\bar{\boldsymbol{x}}_i^t\|^2+\frac{2a_2\mu\delta^2}{\nu}\mathbb{E}\|\check{\boldsymbol{x}}_i^t\|^2+\frac{2\mu}{\nu}\mathbb{E}\|\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\|^2\\ &\overset{(5.77)}{\leq} (1-a_1\mu\nu)\mathbb{E}\|\bar{\boldsymbol{x}}_i^t\|^2+\frac{2a_2\mu\delta^2}{\nu}\mathbb{E}\|\check{\boldsymbol{x}}_i^t\|^2\\ &\quad+\frac{12b\delta^2\mu}{\nu}\mathbb{E}\|\bar{\boldsymbol{x}}_i^t-\bar{\boldsymbol{x}}_0^t\|^2+\frac{24b\delta^2\mu}{\nu}\mathbb{E}\|\check{\boldsymbol{x}}_i^t\|^2\\ &\quad+\frac{36b\delta^2\mu}{\nu}\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2+\frac{6b\delta^2\mu}{\nu\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1}-\bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2\\ &\quad+\frac{12b\delta^2\mu}{\bar{N}\nu}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\\ &=(1-a_1\mu\nu)\mathbb{E}\|\bar{\boldsymbol{x}}_i^t\|^2+\frac{(2a_2+24b)\mu\delta^2}{\nu}\mathbb{E}\|\check{\boldsymbol{x}}_i^t\|^2\\ &\quad+\frac{12b\delta^2\mu}{\nu}\mathbb{E}\|\bar{\boldsymbol{x}}_i^t-\bar{\boldsymbol{x}}_0^t\|^2+\frac{36b\delta^2\mu}{\nu}\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2\\ &\quad+\frac{6b\delta^2\mu}{\nu\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1}-\bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2+\frac{12b\delta^2\mu}{\bar{N}\nu}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\end{aligned} \tag{5.179}$$

Iterate (5.179), then for $0 \leq i \leq \bar{N}-1$ it holds that

$$
\begin{aligned}
&\mathbb{E}\|\bar{\boldsymbol{x}}_{i+1}^t\|^2 \\
&\leq (1-a_1\mu\nu)^{i+1}\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 \\
&\quad + \frac{(2a_2+24b)\mu\delta^2}{\nu}\sum_{j=0}^{i}(1-a_1\mu\nu)^{i-j}\mathbb{E}\|\check{\boldsymbol{x}}_j^t\|^2 \\
&\quad + \frac{12b\delta^2\mu}{\nu}\sum_{j=0}^{i}(1-a_1\mu\nu)^{i-j}\mathbb{E}\|\bar{\boldsymbol{x}}_j^t-\bar{\boldsymbol{x}}_0^t\|^2 \\
&\quad + \Bigg(\frac{36b\delta^2\mu}{\nu}\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2 + \frac{6b\delta^2\mu}{\nu\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1}-\bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2 \\
&\qquad + \frac{12b\delta^2\mu}{\bar{N}\nu}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\Bigg)\sum_{j=0}^{i}(1-a_1\mu\nu)^j \\
&\leq (1-a_1\mu\nu)^{i+1}\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 + \frac{(2a_2+24b)\mu\delta^2}{\nu}\sum_{j=0}^{i}\mathbb{E}\|\check{\boldsymbol{x}}_j^t\|^2 \\
&\quad + \frac{12b\delta^2\mu}{\nu}\sum_{j=0}^{i}\mathbb{E}\|\bar{\boldsymbol{x}}_j^t-\bar{\boldsymbol{x}}_0^t\|^2 + \Bigg(\frac{36b\delta^2\mu}{\nu}\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2 \\
&\qquad + \frac{6b\delta^2\mu}{\nu\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1}-\bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2 \\
&\qquad + \frac{12b\delta^2\mu}{\bar{N}\nu}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\Bigg)(i+1),
\end{aligned}
\tag{5.180}
$$

where the last inequality hold when we choose μ small enough such that

$$
0 < 1-a_1\mu\nu < 1 \Longleftrightarrow \mu < \frac{1}{a_1\nu}. \tag{5.181}
$$

Let $i = \bar{N} - 1$ in (5.180). It holds that

$$
\begin{aligned}
&\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t+1}\|^2 \\
&\leq (1 - a_1\mu\nu)^{\bar{N}}\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 + \frac{(2a_2 + 24b)\mu\delta^2}{\nu}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^t\|^2 \\
&\quad + \frac{12b\delta^2\mu}{\nu}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^t - \bar{\boldsymbol{x}}_0^t\|^2 + \left(\frac{36b\delta^2\bar{N}\mu}{\nu}\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2\right. \\
&\quad \left. + \frac{6b\delta^2\mu}{\nu}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1} - \bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2 + \frac{12b\delta^2\mu}{\nu}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\right) \\
&= (1 - a_1\mu\nu)^{\bar{N}}\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 + \frac{(2a_2 + 24b)\mu\delta^2\bar{N}}{\nu}\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^t\|^2\right) \\
&\quad + \frac{12b\delta^2\mu\bar{N}}{\nu}\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^t - \bar{\boldsymbol{x}}_0^t\|^2\right) + \frac{36b\delta^2\bar{N}\mu}{\nu}\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2 \\
&\quad + \frac{6b\delta^2\mu\bar{N}}{\nu}\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1} - \bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2\right) \\
&\quad + \frac{12b\delta^2\mu\bar{N}}{\nu}\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\right).
\end{aligned} \tag{5.182}
$$

According to Lemma 5.4, the inequality (5.82) holds when step-size μ satisfies

$$
\mu \leq C_1\sqrt{\frac{1-\lambda}{\delta^2\bar{N}}}. \tag{5.183}
$$

Substituting (5.82) into (5.182), we get

$$\begin{aligned}
&\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^{t+1}\|^2 \\
&\leq \left((1-a_1\mu\nu)^{\bar{N}} + \frac{c_1(2a_2+24b)\mu^3\delta^4\bar{N}^2}{\nu}\right)\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^t\|^2 \\
&\quad + \left(\frac{36b\delta^2\bar{N}\mu}{\nu} + \frac{\lambda_3(2a_2+24b)\mu\delta^2\bar{N}}{\nu}\right)\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^t\|^2 \\
&\quad + \left(\frac{12b\delta^2\mu\bar{N}}{\nu} + \frac{c_2(2a_2+24b)\mu^3\delta^4\bar{N}^2}{\nu}\right) \\
&\qquad \cdot \left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_j^t - \bar{\boldsymbol{\mathcal{X}}}_0^t\|^2\right) \\
&\quad + \left(\frac{6b\delta^2\mu\bar{N}}{\nu} + \frac{c_3(2a_2+24b)\mu^3\delta^4\bar{N}^2}{\nu}\right) \\
&\qquad \cdot \left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_j^{t-1} - \bar{\boldsymbol{\mathcal{X}}}_{\bar{N}}^{t-1}\|^2\right) \\
&\quad + \left(\frac{12b\delta^2\mu\bar{N}}{\nu} + \frac{c_4(2a_2+24b)\mu^3\delta^4\bar{N}^2}{\nu}\right) \\
&\qquad \cdot \left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_j^{t-1}\|^2\right). \qquad (5.184)
\end{aligned}$$

For the term $(1-a_1\mu\nu)^{\bar{N}}$, it is established in Appendix 5.I that if

$$\mu \leq \frac{1}{a_1\bar{N}\nu}, \qquad (5.185)$$

then the inequality $(1-a_1\mu\nu)^{\bar{N}} \leq 1 - a_1\bar{N}\mu\nu/2$ holds. Furthermore, if the step-size μ is chosen small enough such that

$$\begin{aligned}
1 - \frac{a_1\bar{N}\mu\nu}{2} + \frac{c_1(2a_2+24b)\mu^3\delta^4\bar{N}^2}{\nu} &\leq 1 - \frac{a_1\bar{N}\mu\nu}{3} \\
\frac{12b\delta^2\mu\bar{N}}{\nu} + \frac{c_2(2a_2+24b)\mu^3\delta^4\bar{N}^2}{\nu} &\leq \frac{24b\delta^2\bar{N}\mu}{\nu} \\
\frac{6b\delta^2\mu\bar{N}}{\nu} + \frac{c_3(2a_2+24b)\mu^3\delta^4\bar{N}^2}{\nu} &\leq \frac{12b\delta^2\mu\bar{N}}{\nu} \\
\frac{12b\delta^2\mu\bar{N}}{\nu} + \frac{c_4(2a_2+24b)\mu^3\delta^4\bar{N}^2}{\nu} &\leq \frac{24b\delta^2\mu\bar{N}}{\nu} \qquad (5.186)
\end{aligned}$$

recursion (5.184) will imply

$$\begin{aligned}
&\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t+1}\|^2 \\
&\leq \left(1-\frac{\bar{N}}{3}a_1\mu\nu\right)\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t}\|^2 \\
&\quad+\left(\frac{(36b+2\lambda_3 a_2+24\lambda_3 b)\mu\delta^2\bar{N}}{\nu}\right)\mathbb{E}\|\check{\boldsymbol{x}}_0^{t}\|^2 \\
&\quad+\frac{24b\delta^2\mu\bar{N}}{\nu}\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t}-\bar{\boldsymbol{x}}_0^{t}\|^2\right) \\
&\quad+\frac{12b\delta^2\mu\bar{N}}{\nu}\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1}-\bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2\right) \\
&\quad+\frac{24b\delta^2\mu\bar{N}}{\nu}\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\right). \qquad (5.187)
\end{aligned}$$

To simplify the notation, we let

$$d_1=36b+2\lambda_3 a_2+24\lambda_3 b, d_2=24b, d_3=12b, d_4=24b, \qquad (5.188)$$

then recursion (5.85) is proved. To guarantee (5.181), (5.183), (5.185) and (5.186), it is enough to set

$$\begin{aligned}
\mu\leq\min\Bigg\{&\frac{1}{a_1\nu},\ C_1\sqrt{\frac{1-\lambda}{\delta^2\bar{N}}},\ \frac{1}{a_1\bar{N}\nu}, \\
&\sqrt{\frac{a_1}{6c_1(2a_2+24b)\bar{N}}}\left(\frac{\nu}{\delta^2}\right),\sqrt{\frac{12b}{c_2(2a_2+24b)\delta^2\bar{N}}}, \\
&\sqrt{\frac{6b}{c_3(2a_2+24b)\delta^2\bar{N}}},\sqrt{\frac{12b}{c_4(2a_2+24b)\delta^2\bar{N}}}\Bigg\} \qquad (5.189)
\end{aligned}$$

Note that $\nu^2/\delta^2<1$ and $1-\lambda<1$. To get a simple form for the step-size, we can further restrict μ as

$$\begin{aligned}
\mu&\leq\min\left\{C_1,\frac{1}{a_1},\sqrt{\frac{a_1}{2c_1(2a_2+24b)}},\sqrt{\frac{12b}{c_2(2a_2+24b)}},\right. \\
&\quad\left.\sqrt{\frac{6b}{c_3(2a_2+24b)}},\sqrt{\frac{12b}{c_4(2a_2+24b)}}\right\}\left(\frac{\nu\sqrt{1-\lambda}}{\delta^2\bar{N}}\right) \\
&\triangleq\ C_2\left(\frac{\nu\sqrt{1-\lambda}}{\delta^2\bar{N}}\right), \qquad (5.190)
\end{aligned}$$

where C_2 is independent of ν, δ and $\bar{N}$.

5.I Upper Bound on $(1-a_1\mu\nu)^{\bar{N}}$

We first examine the term $(1-x)^{\bar{N}}$ where $x \in (0,1)$. Using Taylor's theorem, $(1-x)^{\bar{N}}$ can be expanded as

$$(1-x)^{\bar{N}} = 1-\bar{N}x + \frac{\bar{N}(\bar{N}-1)(1-\tau)^{\bar{N}-2}}{2}x^2, \tag{5.191}$$

where $\tau \in (0,x)$ is some constant, and hence, $\tau < 1$. To ensure $(1-x)^{\bar{N}} \leq 1-\frac{1}{2}\bar{N}x$, we require

$$\begin{aligned} & 1-\bar{N}x + \frac{\bar{N}(\bar{N}-1)(1-\tau)^{\bar{N}-2}}{2}x^2 \leq 1-\frac{\bar{N}x}{2} \\ \iff & x \leq \frac{1}{(\bar{N}-1)(1-\tau)^{\bar{N}-2}}. \end{aligned} \tag{5.192}$$

Note that

$$\frac{1}{\bar{N}} < \frac{1}{\bar{N}-1} < \frac{1}{(\bar{N}-1)(1-\tau)^{\bar{N}-2}}. \tag{5.193}$$

If we choose $x \leq 1/\bar{N}$, then it will also satisfy (5.192). By letting $x = a_1\mu\nu$, it holds that

$$(1-a_1\mu\nu)^{\bar{N}} \leq 1-\frac{a_1\bar{N}\mu\nu}{2}. \tag{5.194}$$

when $\mu \leq 1/(a_1\bar{N}\nu)$.

5.J Proof of Lemma 5.6

From the first line in recursion (5.74), we have

$$\bar{\boldsymbol{x}}_{i+1}^t - \bar{\boldsymbol{x}}_i^t = -\frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{I}\bar{\boldsymbol{x}}_i^t - \frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{X}_{R,u}\check{\boldsymbol{x}}_i^t + \frac{\mu}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t) \tag{5.195}$$

By squaring and applying Jensen's inequality, we have

$$\begin{aligned} & \|\bar{\boldsymbol{x}}_{i+1}^t - \bar{\boldsymbol{x}}_i^t\|^2 \\ & \leq \; 3\mu^2 \left\| \frac{1}{K}\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{I} \right\|^2 \|\bar{\boldsymbol{x}}_i^t\|^2 \\ & \quad + \frac{3\mu^2}{K^2}\|\mathcal{I}^{\mathsf{T}}\boldsymbol{\mathcal{H}}_i^t\mathcal{X}_{R,u}\|^2\|\check{\boldsymbol{x}}_i^t\|^2 \\ & \quad + \frac{3\mu^2}{K^2}\|\mathcal{I}^{\mathsf{T}}\|^2\|\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\|^2 \\ & \overset{(a)}{\leq} \; 3\mu^2\delta^2\|\bar{\boldsymbol{x}}_i^t\|^2 + \frac{3\mu^2}{K}\delta^2\|\mathcal{X}_{R,u}\|^2\|\check{\boldsymbol{x}}_i^t\|^2 + \frac{3\mu^2}{K}\|\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\|^2 \end{aligned} \tag{5.196}$$

where inequality (a) holds because of equations (5.133) and (5.137). By taking expectations, we have

$$
\begin{aligned}
&\mathbb{E}\|\bar{\boldsymbol{x}}_{i+1}^t-\bar{\boldsymbol{x}}_i^t\|^2\\
&\leq 3\mu^2\delta^2\mathbb{E}\|\bar{\boldsymbol{x}}_i^t\|^2+\frac{3\mu^2}{K}\delta^2\|\mathcal{X}_{R,u}\|^2\mathbb{E}\|\check{\boldsymbol{x}}_i^t\|^2+\frac{3\mu^2}{K}\mathbb{E}\|\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\|^2\\
&\leq 6\mu^2\delta^2\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2+6\mu^2\delta^2\mathbb{E}\|\bar{\boldsymbol{x}}_i^t-\bar{\boldsymbol{x}}_0^t\|^2\\
&\quad+\frac{3\mu^2}{K}\delta^2\|\mathcal{X}_{R,u}\|^2\mathbb{E}\|\check{\boldsymbol{x}}_i^t\|^2+\frac{3\mu^2}{K}\mathbb{E}\|\boldsymbol{s}(\boldsymbol{\mathcal{W}}_i^t)\|^2\\
&\overset{(5.77)}{\leq} 6\mu^2\delta^2\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2+\frac{54b\mu^2\delta^2}{K}\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|\\
&\quad+\left(\frac{3\|\mathcal{X}_{R,u}\|^2+36b}{K}\right)\mu^2\delta^2\mathbb{E}\|\check{\boldsymbol{x}}_i^t\|^2\\
&\quad+\left(6+\frac{18b}{K}\right)\mu^2\delta^2\mathbb{E}\|\bar{\boldsymbol{x}}_i^t-\bar{\boldsymbol{x}}_0^t\|^2\\
&\quad+\frac{9b\delta^2\mu^2}{K}\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1}-\bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2\right)\\
&\quad+\frac{18b\delta^2\mu^2}{K}\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\right),0\leq i\leq\bar{N}-1
\end{aligned}
\tag{5.197}
$$

For simplicity, if we let

$$
\begin{aligned}
&e_1=\frac{54b}{K},\quad e_2=\frac{3\|\mathcal{X}_{R,u}\|^2+36b}{K},\\
&e_3=6+\frac{18b}{K},\quad e_4=\frac{9b}{K},\quad e_5=\frac{18b}{K},
\end{aligned}
\tag{5.198}
$$

inequality (5.197) becomes

$$
\begin{aligned}
&\mathbb{E}\|\bar{\boldsymbol{x}}_{i+1}^t-\bar{\boldsymbol{x}}_i^t\|^2\\
&\leq 6\mu^2\delta^2\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2+e_1\mu^2\delta^2\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2+e_2\mu^2\delta^2\mathbb{E}\|\check{\boldsymbol{x}}_i^t\|^2\\
&\quad+e_3\mu^2\delta^2\mathbb{E}\|\bar{\boldsymbol{x}}_i^t-\bar{\boldsymbol{x}}_0^t\|^2\\
&\quad+e_4\mu^2\delta^2\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1}-\bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2\right)\\
&\quad+e_5\mu^2\delta^2\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\right).
\end{aligned}
\tag{5.199}
$$

For $1 \leq i \leq \bar{N}-1$, we have

$$
\begin{aligned}
&\mathbb{E}\|\bar{\boldsymbol{x}}_i^t - \bar{\boldsymbol{x}}_0^t\|^2 \\
&\leq i \sum_{j=1}^{i} \mathbb{E}\|\bar{\boldsymbol{x}}_j^t - \bar{\boldsymbol{x}}_{j-1}^t\|^2 \\
&\overset{(5.199)}{\leq} 6\mu^2\delta^2 i^2 \mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 + e_1\mu^2\delta^2 i^2 \mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2 \\
&\quad + e_2\mu^2\delta^2 i \sum_{j=1}^{i} \mathbb{E}\|\check{\boldsymbol{x}}_{j-1}^t\|^2 + e_3\mu^2\delta^2 i \sum_{j=1}^{i} \mathbb{E}\|\bar{\boldsymbol{x}}_{j-1}^t - \bar{\boldsymbol{x}}_0^t\|^2 \\
&\quad + e_4\mu^2\delta^2 i^2 \left(\frac{1}{\bar{N}} \sum_{j=0}^{\bar{N}-1} \mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1} - \bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2 \right) \\
&\quad + e_5\mu^2\delta^2 i^2 \left(\frac{1}{\bar{N}} \sum_{j=0}^{\bar{N}-1} \mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2 \right) \\
&\leq 6\mu^2\delta^2 \bar{N}^2 \mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2 + e_1\mu^2\delta^2 \bar{N}^2 \mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2 \\
&\quad + e_2\mu^2\delta^2 \bar{N}^2 \left(\frac{1}{\bar{N}} \sum_{j=0}^{\bar{N}-1} \mathbb{E}\|\check{\boldsymbol{x}}_j^t\|^2 \right) \\
&\quad + e_3\mu^2\delta^2 \bar{N}^2 \left(\frac{1}{\bar{N}} \sum_{j=0}^{\bar{N}-1} \mathbb{E}\|\bar{\boldsymbol{x}}_j^t - \bar{\boldsymbol{x}}_0^t\|^2 \right) \\
&\quad + e_4\mu^2\delta^2 \bar{N}^2 \left(\frac{1}{\bar{N}} \sum_{j=0}^{\bar{N}-1} \mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1} - \bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2 \right) \\
&\quad + e_5\mu^2\delta^2 \bar{N}^2 \left(\frac{1}{\bar{N}} \sum_{j=0}^{\bar{N}-1} \mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2 \right). \qquad (5.200)
\end{aligned}
$$

From the above recursion, we can also derive

$$\begin{aligned}
&\frac{1}{\bar{N}}\sum_{i=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_i^t-\bar{\boldsymbol{x}}_0^t\|^2\\
&\leq 6\mu^2\delta^2\bar{N}^2\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2+e_1\mu^2\delta^2\bar{N}^2\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2\\
&\quad+e_2\mu^2\delta^2\bar{N}^2\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^t\|^2\right)\\
&\quad+e_3\mu^2\delta^2\bar{N}^2\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^t-\bar{\boldsymbol{x}}_0^t\|^2\right)\\
&\quad+e_4\mu^2\delta^2\bar{N}^2\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1}-\bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2\right)\\
&\quad+e_5\mu^2\delta^2\bar{N}^2\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\right)
\end{aligned}\tag{5.201}$$

According to Lemma 5.4, the inequality (5.82) holds when step-size μ satisfies

$$\mu\leq C_1\sqrt{\frac{1-\lambda}{\delta^2\overline{N}}}.\tag{5.202}$$

Substituting (5.82) into (5.201), we have

$$\begin{aligned}
&\frac{1}{\bar{N}}\sum_{i=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_i^t-\bar{\boldsymbol{x}}_0^t\|^2\\
&\leq\left(6\mu^2\delta^2\bar{N}^2+c_1e_2\mu^4\delta^4\bar{N}^3\right)\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2\\
&\quad+(e_1+\lambda_3e_2)\mu^2\delta^2\bar{N}^2\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2\\
&\quad+\left(e_3\mu^2\delta^2\bar{N}^2+c_2e_2\mu^4\delta^4\bar{N}^3\right)\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^t-\bar{\boldsymbol{x}}_0^t\|^2\right)\\
&\quad+\left(e_4\mu^2\delta^2\bar{N}^2+c_3e_2\mu^4\delta^4\bar{N}^3\right)\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1}-\bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2\right)\\
&\quad+\left(e_5\mu^2\delta^2\bar{N}^2+c_4e_2\mu^4\delta^4\bar{N}^3\right)\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\right).
\end{aligned}\tag{5.203}$$

If the step-size μ is chosen small enough such that

$$\begin{aligned}
&6\mu^2\delta^2\bar{N}^2 + c_1 e_2 \mu^4\delta^4\bar{N}^3 \leq 12\mu^2\delta^2\bar{N}^2,\\
&e_3\mu^2\delta^2\bar{N}^2 + c_2 e_2 \mu^4\delta^4\bar{N}^3 \leq 2e_3\mu^2\delta^2\bar{N}^2,\\
&e_4\mu^2\delta^2\bar{N}^2 + c_3 e_2 \mu^4\delta^4\bar{N}^3 \leq 2e_4\mu^2\delta^2\bar{N}^2,\\
&e_5\mu^2\delta^2\bar{N}^2 + c_4 e_2 \mu^4\delta^4\bar{N}^3 \leq 2e_5\mu^2\delta^2\bar{N}^2.
\end{aligned} \tag{5.204}$$

then recursion (5.201) can be simplified to equation (5.87), where we define $e_6 \stackrel{\Delta}{=} e_1 + \lambda_2 e_2$. To guarantee (5.202) and (5.204), it is enough to set

$$\begin{aligned}
\mu &\leq \min\left\{C_1, \sqrt{\frac{6}{c_1 e_2}}, \sqrt{\frac{e_3}{c_2 e_2}}, \sqrt{\frac{e_4}{c_3 e_2}}, \sqrt{\frac{e_5}{c_4 e_2}}\right\}\sqrt{\frac{1-\lambda}{\delta^2\bar{N}}}\\
&\stackrel{\Delta}{=} C_3\sqrt{\frac{1-\lambda}{\delta^2\bar{N}}}.
\end{aligned} \tag{5.205}$$

Next we establish the recursion for $\sum_{i=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_i^t-\bar{\boldsymbol{x}}_{\bar{N}}^t\|^2/\overline{N}$. Note that for $0\leq i\leq\bar{N}-1$, it holds that

$$
\begin{aligned}
&\mathbb{E}\|\bar{\boldsymbol{x}}_i^t-\bar{\boldsymbol{x}}_{\bar{N}}^t\|^2\\
&\leq(\bar{N}-i)\sum_{j=i}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_{j+1}^t-\bar{\boldsymbol{x}}_j^t\|^2\\
&\overset{(5.199)}{\leq}\ 6\mu^2\delta^2(\bar{N}-i)^2\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2+e_1\mu^2\delta^2(N-i)^2\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2\\
&\quad+e_2\mu^2\delta^2(\bar{N}-i)\sum_{j=i}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^t\|^2\\
&\quad+e_3\mu^2\delta^2(N-i)\sum_{j=i}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_{j-1}^t-\bar{\boldsymbol{x}}_0^t\|^2\\
&\quad+e_4\mu^2\delta^2(\bar{N}-i)^2\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1}-\bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2\right)\\
&\quad+e_5\mu^2\delta^2(\bar{N}-i)^2\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\right)\\
&\leq 6\mu^2\delta^2\bar{N}^2\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2+e_1\mu^2\delta^2\bar{N}^2\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2\\
&\quad+e_2\mu^2\delta^2\bar{N}^2\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^t\|^2\right)\\
&\quad+e_3\mu^2\delta^2\bar{N}^2\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^t-\bar{\boldsymbol{x}}_0^t\|^2\right)\\
&\quad+e_4\mu^2\delta^2\bar{N}^2\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\bar{\boldsymbol{x}}_j^{t-1}-\bar{\boldsymbol{x}}_{\bar{N}}^{t-1}\|^2\right)\\
&\quad+e_5\mu^2\delta^2\bar{N}^2\left(\frac{1}{\bar{N}}\sum_{j=0}^{\bar{N}-1}\mathbb{E}\|\check{\boldsymbol{x}}_j^{t-1}\|^2\right).
\end{aligned}
\tag{5.206}
$$

Since the right-hand side of inequality (5.206) is the same as inequality (5.200), we can follow (5.201)–(5.205) to conclude recursion (5.88).

5.K Proof of Theorem 5.7

With Lemmas 5.4, 5.5 and 5.6, when the step-size μ satisfies

$$\boxed{\mu \leq \min\left\{C_1\sqrt{\frac{1-\lambda}{\delta^2\bar{N}}},\ \ C_2\left(\frac{\nu\sqrt{1-\lambda}}{\delta^2\bar{N}}\right),\ \ C_3\sqrt{\frac{1-\lambda}{\delta^2\bar{N}}}\right\},} \tag{5.207}$$

it holds that

$$\begin{aligned}\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^{t+1}\|^2 \leq& \left(1-\frac{\bar{N}}{3}a_1\mu\nu\right)\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^{t}\|^2+\frac{d_1\mu\delta^2\bar{N}}{\nu}\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^{t}\|^2\\&+\frac{d_2\delta^2\mu\bar{N}}{\nu}\mathbf{A}^t+\frac{d_3\delta^2\mu\bar{N}}{\nu}\mathbf{B}^{t-1}+\frac{d_4\delta^2\mu\bar{N}}{\nu}\mathbf{C}^{t-1}\end{aligned} \tag{5.208}$$

$$\begin{aligned}\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^{t+1}\|^2 \leq& c_1\mu^2\delta^2\bar{N}\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^{t}\|^2+\lambda_2\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^{t}\|^2\\&+c_2\mu^2\delta^2\bar{N}\mathbf{A}^t+c_3\mu^2\delta^2\bar{N}\mathbf{B}^{t-1}+c_4\mu^2\delta^2\bar{N}\mathbf{C}^{t-1}\end{aligned} \tag{5.209}$$

$$\begin{aligned}\mathbf{A}^{t+1} \leq& 12\mu^2\delta^2\bar{N}^2\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^{t+1}\|^2+e_6\mu^2\delta^2\bar{N}^2\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^{t+1}\|^2\\&+2e_3\mu^2\delta^2\bar{N}^2\mathbf{A}^{t+1}+2e_4\mu^2\delta^2\bar{N}^2\mathbf{B}^{t}+2e_5\mu^2\delta^2\bar{N}^2\mathbf{C}^{t}\end{aligned} \tag{5.210}$$

$$\begin{aligned}\mathbf{B}^{t} \leq& 12\mu^2\delta^2\bar{N}^2\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^{t}\|^2+e_6\mu^2\delta^2\bar{N}^2\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^{t}\|^2\\&+2e_3\mu^2\delta^2\bar{N}^2\mathbf{A}^{t}+2e_4\mu^2\delta^2\bar{N}^2\mathbf{B}^{t-1}+2e_5\mu^2\delta^2\bar{N}^2\mathbf{C}^{t-1}\end{aligned} \tag{5.211}$$

$$\begin{aligned}\mathbf{C}^{t} \leq& c_1\mu^2\delta^2\bar{N}\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^{t}\|^2+\lambda_3\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^{t}\|^2\\&+c_2\mu^2\delta^2\bar{N}\mathbf{A}^t+c_3\mu^2\delta^2\bar{N}\mathbf{B}^{t-1}+c_4\mu^2\delta^2\bar{N}\mathbf{C}^{t-1}\end{aligned} \tag{5.212}$$

Let γ be an arbitrary positive constant whose value will be decided later. From the above inequalities we have

$$\begin{aligned}
&\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^{t+1}\|^2+\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^{t+1}\|^2+\gamma\left(\mathbf{A}^{t+1}+\mathbf{B}^t+\mathbf{C}^t\right)\\
\leq&\left(1-\frac{\bar{N}}{3}a_1\mu\nu+c_1\mu^2\delta^2\bar{N}\right)\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^t\|^2+\left(\lambda_2+\frac{d_1\mu\delta^2\bar{N}}{\nu}\right)\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^t\|^2\\
&+\left(\frac{d_2\delta^2\mu\bar{N}}{\nu}+c_2\mu^2\delta^2\bar{N}\right)\mathbf{A}^t+\left(\frac{d_3\delta^2\mu\bar{N}}{\nu}+c_3\mu^2\delta^2\bar{N}\right)\mathbf{B}^{t-1}\\
&+\left(\frac{d_4\delta^2\mu\bar{N}}{\nu}+c_4\mu^2\delta^2\bar{N}\right)\mathbf{C}^{t-1}\\
&+\gamma f_1\mu^2\delta^2\bar{N}^2\left(\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^{t+1}\|^2+\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^{t+1}\|^2\right)\\
&+\gamma f_2\mu^2\delta^2\bar{N}^2(\mathbf{A}^{t+1}+\mathbf{B}^t+\mathbf{C}^t)+\gamma f_3\mu^2\delta^2\bar{N}^2\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^t\|^2\\
&+\gamma(\lambda_3+e_6\mu^2\delta^2\bar{N}^2)\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^t\|^2\\
&+\gamma f_4\mu^2\delta^2\bar{N}^2(\mathbf{A}^t+\mathbf{B}^{t-1}+\mathbf{C}^{t-1}),
\end{aligned}\tag{5.213}$$

where the constants $\{f_i\}_{i=1}^4$ are defined as

$$f_1=\max\{12,e_6\},\, f_2=2\max\{e_3,e_4,e_5\},\tag{5.214}$$

$$f_3=12+c_1,\qquad f_4=\max\{2e_3+c_2,2e_4+c_3,2e_5+c_4\}.\tag{5.215}$$

If the step-size μ is chosen small enough such that

$$1-\frac{\bar{N}}{3}a_1\mu\nu+c_1\mu^2\delta^2\bar{N}\leq 1-\frac{\bar{N}}{4}a_1\mu\nu,\tag{5.216}$$

$$\lambda_2+\frac{d_1\mu\delta^2\bar{N}}{\nu}\leq\frac{1+\lambda_2}{2}\;\triangleq\;\lambda_4<1,\tag{5.217}$$

$$\frac{d_2\delta^2\mu\bar{N}}{\nu}+c_2\mu^2\delta^2\bar{N}\leq\frac{2d_2\delta^2\mu\bar{N}}{\nu},\tag{5.218}$$

$$\frac{d_3\delta^2\mu\bar{N}}{\nu}+c_3\mu^2\delta^2\bar{N}\leq\frac{2d_3\delta^2\mu\bar{N}}{\nu},\tag{5.219}$$

$$\frac{d_4\delta^2\mu\bar{N}}{\nu}+c_4\mu^2\delta^2\bar{N}\leq\frac{2d_4\delta^2\mu\bar{N}}{\nu},\tag{5.220}$$

recursion (5.213) can be simplified to

$$\begin{aligned}
&(1-\gamma f_1\mu^2\delta^2\bar{N}^2)\left(\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^{t+1}\|^2+\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^{t+1}\|^2\right)\\
&\qquad+\gamma(1-f_2\mu^2\delta^2\bar{N}^2)\left(\mathbf{A}^{t+1}+\mathbf{B}^{t}+\mathbf{C}^{t}\right)\\
\le\;&\left(1-\frac{\bar{N}}{4}a_1\mu\nu\right)\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^{t}\|^2+\lambda_4\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^{t}\|^2\\
&+\frac{2d_2\delta^2\mu\bar{N}}{\nu}\mathbf{A}^t+\frac{2d_3\delta^2\mu\bar{N}}{\nu}\mathbf{B}^{t-1}+\frac{2d_4\delta^2\mu\bar{N}}{\nu}\mathbf{C}^{t-1}\\
&+\gamma f_3\mu^2\delta^2\bar{N}^2\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^{t}\|^2+\gamma(\lambda_3+e_6\mu^2\delta^2\bar{N}^2)\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^{t}\|^2\\
&+\gamma f_4\mu^2\delta^2\bar{N}^2(\mathbf{A}^t+\mathbf{B}^{t-1}+\mathbf{C}^{t-1})\\
\le\;&\left(1-\frac{\bar{N}}{4}a_1\mu\nu+\gamma f_3\mu^2\delta^2\bar{N}^2\right)\mathbb{E}\|\bar{\boldsymbol{\mathcal{X}}}_0^{t}\|^2\\
&+\left[\lambda_4+\gamma(\lambda_3+e_6\mu^2\delta^2\bar{N}^2)\right]\mathbb{E}\|\check{\boldsymbol{\mathcal{X}}}_0^{t}\|^2\\
&+\left(\frac{f_5\delta^2\mu\bar{N}}{\nu}+\gamma f_4\mu^2\delta^2\bar{N}^2\right)(\mathbf{A}^t+\mathbf{B}^{t-1}+\mathbf{C}^{t-1}),
\end{aligned}\tag{5.221}$$

where $f_5 \triangleq 2\max\{d_2,d_3,d_4\}$. To guarantee (5.217)–(5.220), it is enough to set

$$\mu\le\min\left\{\frac{a_1\nu}{12c_1\delta^2},\frac{(1-\lambda_2)\nu}{2d_1\delta^2\bar{N}},\frac{d_2}{c_2\nu},\frac{d_3}{c_3\nu},\frac{d_4}{c_4\nu}\right\}.\tag{5.222}$$

Since $\nu/\delta<1$, it holds that

$$\frac{d_l}{c_l\nu}\ge\frac{d_l}{c_l\nu}\frac{\nu^2}{\delta^2\bar{N}}=\frac{d_l\nu}{c_l\delta^2\bar{N}},2\le l\le 4.\tag{5.223}$$

Also recall that $1-\lambda_2=(1-\lambda)/4$. Therefore, if μ satisfies

$$\boxed{\mu\le\min\left\{\frac{a_1}{12c_1},\frac{1}{8d_1},\frac{d_2}{c_2},\frac{d_3}{c_3},\frac{d_4}{c_4}\right\}\frac{\nu(1-\lambda)}{\delta^2\bar{N}}\triangleq C_4\frac{\nu(1-\lambda)}{\delta^2\bar{N}}}\tag{5.224}$$

it also satisfies (5.222). Next we continue simplifying recursion (5.221). Suppose μ and γ are chosen such that

$$1-\frac{\bar{N}}{4}a_1\mu\nu+\gamma f_3\mu^2\delta^2\bar{N}^2\le 1-\frac{\bar{N}}{8}a_1\mu\nu,\tag{5.225}$$

$$\lambda_4+\gamma(\lambda_3+e_6\mu^2\delta^2\bar{N}^2)\le\frac{1+\lambda_4}{2}\triangleq\lambda_5<1,\tag{5.226}$$

$$\frac{f_5\delta^2\mu\bar{N}}{\nu}+\gamma f_4\mu^2\delta^2\bar{N}^2\le\frac{2f_5\delta^2\mu\bar{N}}{\nu},\tag{5.227}$$

recursion (5.221) can be further simplified to

$$\begin{aligned}
&(1-\gamma f_1\mu^2\delta^2\bar{N}^2)\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t+1}\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^{t+1}\|^2\right)\\
&\qquad\qquad+\gamma(1-f_2\mu^2\delta^2\bar{N}^2)\left(\mathbf{A}^{t+1}+\mathbf{B}^{t}+\mathbf{C}^{t}\right)\\
\leq\ &\left(1-\frac{\bar{N}}{8}a_1\mu\nu\right)\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t}\|^2\\
&+\lambda_5\mathbb{E}\|\check{\boldsymbol{x}}_0^{t}\|^2+\frac{2f_5\delta^2\mu\bar{N}}{\nu}(\mathbf{A}^{t}+\mathbf{B}^{t-1}+\mathbf{C}^{t-1}).
\end{aligned}\tag{5.228}$$

Now we check the conditions on μ and γ to satisfy (5.225)–(5.227). Since $\lambda_3<1$, if we choose μ and γ such that

$$\lambda_3+e_6\mu^2\delta^2\bar{N}^2\leq 1,\tag{5.229}$$

$$\lambda_4+\gamma\leq\frac{1+\lambda_4}{2},\tag{5.230}$$

then inequality (5.226) holds. To guarantee (5.225), (5.227) and (5.230), it is enough to set

$$\gamma\leq\frac{1-\lambda_4}{2},\mu\leq\sqrt{\frac{1-\lambda_3}{e_6\delta^2\bar{N}^2}},\gamma\mu\leq\min\left\{\frac{a_1\nu}{8f_3\delta^2\bar{N}},\frac{f_5}{f_4\nu\bar{N}}\right\}.\tag{5.231}$$

Moreover, if we further choose step-size μ such that

$$\boxed{\lambda_5\leq 1-\frac{\bar{N}}{8}a_1\mu\nu\Longleftrightarrow\mu\leq\frac{8(1-\lambda_5)}{a_1\nu\bar{N}},}\tag{5.232}$$

recursion (5.228) becomes

$$\begin{aligned}
&(1-\gamma f_1\mu^2\delta^2\bar{N}^2)\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t+1}\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^{t+1}\|^2\right)\\
&\qquad+\gamma(1-f_2\mu^2\delta^2\bar{N}^2)\left(\mathbf{A}^{t+1}+\mathbf{B}^{t}+\mathbf{C}^{t}\right)\\
\leq\ &\left(1-\frac{\bar{N}}{8}a_1\mu\nu\right)\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t}\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^{t}\|^2\right)\\
&\qquad+\frac{2f_5\delta^2\mu\bar{N}}{\nu}(\mathbf{A}^{t}+\mathbf{B}^{t-1}+\mathbf{C}^{t-1})
\end{aligned}\tag{5.233}$$

When μ and γ are chosen such that

$$1-\gamma f_1\mu^2\delta^2\bar{N}^2>0\Longleftrightarrow\gamma\mu^2<\frac{1}{f_1\delta^2\bar{N}^2},\tag{5.234}$$

recursion (5.233) is equivalent to

$$\begin{aligned}
&\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t+1}\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^{t+1}\|^2\right)\\
&\quad+\gamma\left(\frac{1-f_2\mu^2\delta^2\bar{N}^2}{1-\gamma f_1\mu^2\delta^2\bar{N}^2}\right)\left(\mathbf{A}^{t+1}+\mathbf{B}^t+\mathbf{C}^t\right)\\
&\le\frac{1-\frac{\bar{N}}{8}a_1\mu\nu}{1-\gamma f_1\mu^2\delta^2\bar{N}^2}\left\{\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2\right)\right.\\
&\quad\left.+\frac{2f_5\delta^2\mu\bar{N}}{\nu(1-a_1\bar{N}\mu\nu/8)}(\mathbf{A}^t+\mathbf{B}^{t-1}+\mathbf{C}^{t-1})\right\}
\end{aligned}\tag{5.235}$$

If we also choose μ such that

$$1-f_2\mu^2\delta^2\bar{N}^2\ge\frac{1}{2},\quad\text{and}\quad 1-\frac{1}{8}a_1\bar{N}\mu\nu\ge\frac{1}{2},\tag{5.236}$$

recursion (5.235) can be simplified as

$$\begin{aligned}
&\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t+1}\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^{t+1}\|^2\right)+\frac{\gamma}{2}\left(\mathbf{A}^{t+1}+\mathbf{B}^t+\mathbf{C}^t\right)\\
&\le\frac{1-\frac{1}{8}a_1\bar{N}\mu\nu}{1-\gamma f_1\mu^2\delta^2\bar{N}^2}\left\{\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2\right)\right.\\
&\quad\left.+\frac{4f_5\delta^2\mu\bar{N}}{\nu}(\mathbf{A}^t+\mathbf{B}^{t-1}+\mathbf{C}^{t-1})\right\}.
\end{aligned}\tag{5.237}$$

To guarantee (5.236), it is enough to set

$$\boxed{\mu\le\min\left\{\sqrt{\frac{1}{2f_2\delta^2\bar{N}^2}},\frac{4}{a_1\nu\bar{N}}\right\}.}\tag{5.238}$$

If we let

$$\gamma=8f_5\delta^2\mu\bar{N}/\nu>0,\tag{5.239}$$

then recursion (5.237) is equivalent to

$$\begin{aligned}
&\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t+1}\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^{t+1}\|^2\right)+\frac{\gamma}{2}\left(\mathbf{A}^{t+1}+\mathbf{B}^t+\mathbf{C}^t\right)\\
&\le\frac{1-\frac{\bar{N}}{8}a_1\mu\nu}{1-8f_1f_5\mu^3\delta^4\bar{N}^3/\nu}\left\{\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^t\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^t\|^2\right)\right.\\
&\quad\left.+\frac{\gamma}{2}(\mathbf{A}^t+\mathbf{B}^{t-1}+\mathbf{C}^{t-1})\right\}.
\end{aligned}\tag{5.240}$$

If μ is small enough such that

$$1-\frac{8f_1f_5\mu^3\delta^4\bar{N}^3}{\nu}>1-\frac{1}{8}a_1\bar{N}\mu\nu\iff\boxed{\mu<\sqrt{\frac{a_1}{64f_1f_5}}\frac{\nu}{\delta^2\bar{N}}}\tag{5.241}$$

it then holds that

$$\begin{aligned}&\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t+1}\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^{t+1}\|^2\right)+\frac{\gamma}{2}\left(\mathbf{A}^{t+1}+\mathbf{B}^{t}+\mathbf{C}^{t}\right)\\&\leq\rho\left\{\left(\mathbb{E}\|\bar{\boldsymbol{x}}_0^{t}\|^2+\mathbb{E}\|\check{\boldsymbol{x}}_0^{t}\|^2\right)+\frac{\gamma}{2}(\mathbf{A}^{t}+\mathbf{B}^{t-1}+\mathbf{C}^{t-1})\right\},\end{aligned}\tag{5.242}$$

where

$$\rho=\frac{1-\frac{\bar{N}}{8}a_1\mu\nu}{1-8f_1f_5\mu^3\delta^4\bar{N}^3/\nu}<1.\tag{5.243}$$

Finally, we decide the feasible range of step-size μ. Substituting γ into (5.231) and (5.234), it requires

$$\begin{aligned}\mu\leq\min\Bigg\{&\frac{1-\lambda_4}{16f_5}\frac{\nu}{\delta^2\bar{N}},\sqrt{\frac{1-\lambda_3}{e_6}}\sqrt{\frac{1}{\delta^2\bar{N}}},\sqrt{\frac{a_1}{64f_3f_5}}\left(\frac{\nu}{\delta^2\bar{N}}\right),\\&\sqrt{\frac{1}{8f_4}}\frac{1}{\delta\bar{N}},\left(\frac{\nu}{8f_1f_5\delta^4\bar{N}^3}\right)^{1/3}\Bigg\}.\end{aligned}\tag{5.244}$$

Note that $1-\lambda_4=(1-\lambda)/8$ and $1-\lambda_3\geq(1-\lambda)/8$, and hence if we restrict μ as

$$\begin{aligned}\mu\leq\min\Bigg\{&\frac{1}{128f_5},\sqrt{\frac{1}{8e_6}},\sqrt{\frac{a_1}{64f_3f_5}},\sqrt{\frac{1}{8f_4}},\\&\left(\frac{1}{8f_1f_5}\right)^{1/3}\Bigg\}\frac{\nu(1-\lambda)}{\delta^2\bar{N}}\ \triangleq\ \frac{C_5\nu(1-\lambda)}{\delta^2\bar{N}}\end{aligned}\tag{5.245}$$

it can be verified that such μ satisfies (5.244). Combining all step-size requirements in (5.207), (5.224), (5.232), (5.238), (5.241) and (5.245) recalling $1-\lambda_5=(1-\lambda)/16$, we can always find a constant C.

$$C\ \triangleq\ \min\left\{C_1,C_2,C_3,C_4,C_5,\frac{1}{2a_1},\sqrt{\frac{1}{2f_2}},\frac{4}{a_1},\sqrt{\frac{a_1}{64f_1f_5}}\right\}\tag{5.246}$$

such that if step-size μ satisfies

$$\mu<\frac{C\nu(1-\lambda)}{\delta^2\bar{N}},\tag{5.247}$$

then all requirements in (5.207), (5.224), (5.232), (5.238), (5.241) and (5.245) will be satisfied. Note that C is independent of ν, δ and $\bar{N}$.

Algorithm 5.4 (diffusion-AVRG at node k for unbalanced data)

Initialize $\boldsymbol{w}_{k,0}$ arbitrarily; let $q_k = N_k/N$, $\boldsymbol{\psi}_{k,0} = \boldsymbol{w}_{k,0}$, $\boldsymbol{g}_k^0 = 0$, and $\nabla Q(\boldsymbol{\theta}_{k,0}^0; x_{k,n}) \leftarrow 0,\ 1 \le n \le N_k$

Repeat $i = 0, 1, 2, \cdots$

calculate t and s such that $i{=}tN_k{+}s$, where $t \in \mathbb{Z}_+$ and $s{=}\ \mathrm{mod}(i, N_k)$;

If $s = 0$:

generate a random permutation$\boldsymbol{\sigma}_k^t$; let $\boldsymbol{g}_k^{t+1}{=}0,\ \boldsymbol{\theta}_{k,0}^t = \boldsymbol{w}_{k,i}$;

End

generate the local stochastic gradient:

$$\boldsymbol{n}_s^t = \boldsymbol{\sigma}_k^t(s+1), \tag{5.23}$$

$$\widehat{\nabla J}_k(\boldsymbol{w}_{k,i}) = \nabla Q(\boldsymbol{w}_{k,i}; x_{k,\boldsymbol{n}_s^t}) - \nabla Q(\boldsymbol{\theta}_{k,0}^t; x_{k,\boldsymbol{n}_s^t}) + \boldsymbol{g}_k^t, \tag{5.24}$$

$$\boldsymbol{g}_k^{t+1} \leftarrow \boldsymbol{g}_k^{t+1} + \frac{1}{N_k}\nabla Q(\boldsymbol{w}_{k,i}; x_{k,\boldsymbol{n}_s^t}), \tag{5.25}$$

update $\boldsymbol{w}_{k,i+1}$ with exact diffusion:

$$\boldsymbol{\psi}_{k,i+1} = \boldsymbol{w}_{k,i} - \mu q_k \widehat{\nabla J}_k(\boldsymbol{w}_{k,i}), \tag{5.26}$$

$$\boldsymbol{\phi}_{k,i+1} = \boldsymbol{\psi}_{k,i+1} + \boldsymbol{w}_{k,i} - \boldsymbol{\psi}_{k,i}, \tag{5.27}$$

$$\boldsymbol{w}_{k,i+1} = \sum_{\ell \in \mathcal{N}_k} \overline{a}_{\ell k}\boldsymbol{\phi}_{\ell,i+1}. \tag{5.28}$$

End

Algorithm 5.5 (diffusion-SVRG at node k for unbalanced data)

Initialize $\boldsymbol{w}_{k,0}$ arbitrarily; let $q_k = N_k/N$, $\boldsymbol{\psi}_{k,0} = \boldsymbol{w}_{k,0}$

Repeat $i = 0, 1, 2, \cdots$

calculate t and s such that $i{=}tN_k{+}s$, where $t \in \mathbb{Z}_+$ and $s{=}\mod(i, N_k)$;

If $s = 0$:

generate a random permutation function $\boldsymbol{\sigma}_k^t$, set $\boldsymbol{\theta}_{k,0}^t = \boldsymbol{w}_{k,i}$

and compute the full gradient:

$$\boldsymbol{g}_k^t = \frac{1}{N_k}\sum_{n=1}^{N_k} \nabla Q(\boldsymbol{\theta}_{k,0}^t; x_{k,n}), \tag{5.29}$$

End

generate the local stochastic gradient:

$$\boldsymbol{n}_s^t = \boldsymbol{\sigma}_k^t(s+1), \tag{5.30}$$

$$\widehat{\nabla J}_k(\boldsymbol{w}_{k,i}) = \nabla Q(\boldsymbol{w}_{k,i}; x_{k,\boldsymbol{n}_s^t}) - \nabla Q(\boldsymbol{\theta}_{k,0}^t; x_{k,\boldsymbol{n}_s^t}) + \boldsymbol{g}_k^t, \tag{5.31}$$

update $\boldsymbol{w}_{k,i+1}$ with exact diffusion:

$$\boldsymbol{\psi}_{k,i+1} = \boldsymbol{w}_{k,i} - \mu q_k \widehat{\nabla J}_k(\boldsymbol{w}_{k,i}), \tag{5.32}$$

$$\boldsymbol{\phi}_{k,i+1} = \boldsymbol{\psi}_{k,i+1} + \boldsymbol{w}_{k,i} - \boldsymbol{\psi}_{k,i}, \tag{5.33}$$

$$\boldsymbol{w}_{k,i+1} = \sum_{\ell\in\mathcal{N}_k} \overline{a}_{\ell k}\boldsymbol{\phi}_{\ell,i+1}. \tag{5.34}$$

End

Algorithm 5.6 (diffusion-AVRG with mini-batch at node k)

Initialize $\boldsymbol{w}_{k,0}^0$ arbitrarily; let $\boldsymbol{\psi}_{k,0}^0 = \boldsymbol{w}_{k,0}^0$, $\boldsymbol{g}_k^0 = 0$; equally partition the data into L batches, and each batch has size B. Set $\nabla Q_k^{(\ell)}(\boldsymbol{w}_0^0) \leftarrow 0,\ \ 1 \le \ell \le \bar{L}$

Repeat epoch $t = 0, 1, 2, \cdots$

generate a random permutation function $\boldsymbol{\sigma}_k^t$ and set $\boldsymbol{g}_k^{t+1} = 0$.

Repeat iteration $i = 0, 1, \cdots, L-1$:

$$\boldsymbol{\ell}_{k,i}^t = \boldsymbol{\sigma}_k^t(i+1), \tag{5.38}$$

$$\widehat{\nabla J}_k(\boldsymbol{w}_{k,i}^t) = \nabla Q_k^{(\boldsymbol{\ell}_{k,i}^t)}(\boldsymbol{w}_{k,i}^t) - \nabla Q_k^{(\boldsymbol{\ell}_{k,i}^t)}(\boldsymbol{w}_{k,0}^t) + \boldsymbol{g}_k^t, \tag{5.39}$$

$$\boldsymbol{g}_k^{t+1} \leftarrow \boldsymbol{g}_k^{t+1} + \frac{1}{L}\nabla Q_k^{(\boldsymbol{\ell}_{k,i}^t)}(\boldsymbol{w}_{k,i}^t), \tag{5.40}$$

update $\boldsymbol{w}_{k,i+1}^t$ with exact diffusion:

$$\boldsymbol{\psi}_{k,i+1}^t = \boldsymbol{w}_{k,i}^t - \mu\widehat{\nabla J}_k(\boldsymbol{w}_{k,i}^t), \tag{5.41}$$

$$\boldsymbol{\phi}_{k,i+1}^t = \boldsymbol{\psi}_{k,i+1}^t + \boldsymbol{w}_{k,i}^t - \boldsymbol{\psi}_{k,i}^t, \tag{5.42}$$

$$\boldsymbol{w}_{k,i+1}^t = \sum_{\ell \in \mathcal{N}_k} \overline{a}_{\ell k}\boldsymbol{\phi}_{\ell,i+1}^t. \tag{5.43}$$

End

set $\boldsymbol{w}_{k,0}^{t+1} = \boldsymbol{w}_{k,L}^t$ and $\boldsymbol{\psi}_{k,0}^{t+1} = \boldsymbol{\psi}_{k,L}^t$

End

CHAPTER 6

Conclusion and Future Work

In this dissertation, we proposed an exact diffusion strategy and studied its performance for distributed optimization, adaptation and learning over networks. The main results can be summarized as follows:

- Diffusion strategy solves an approximate problem of the target problem (1.1), which explains why diffusion converges to a small neighborhood around, rather than converges exactly to, the global solution $w^\star$ to problem (1.1).

- We proposed an exact diffusion method to eliminate the bias. Exact diffusion has the same computational complexity as diffusion, and it converges exponentially fast to $w^\star$ under standard assumptions. Furthermore, exact diffusion works for broader family of combination matrices than EXTRA [75], namely, locally-balanced combination matrices. When symmetric and doubly stochastic matrices are employed, exact diffusion is proved to have a wider stability range and hence an improved convergence rate than EXTRA.

- We extended exact diffusion to the distributed adaptation and online learning scenario. Under this stochastic setting, we provide conditions under which exact diffusion has superior steady-state mean-square deviation (MSD) performance than traditional algorithms without bias-correction. In particular, it is proven that this superiority is more evident over sparsely-connected network topologies such as lines, cycles, or grids.

- We extended exact diffusion to the distributed empirical learning scenario. Under this setting, we integrate the amortized variance-reduced learning algorithm to exact diffusion and enable it to converge exponentially fast to the global solution. We

also proposed algorithms that work for the unbalanced data scenario and non-smooth scenario.

While exact diffusion has been extended to various useful scenarios, there are several open issues that deserve further investigation:

- Exact diffusion is studied for undirected network in this dissertation. However, there are applications in which the network is directed. For example, it is very common in practice that agent k can send information to agent ℓ while agent ℓ cannot send information to agent k. In this case, the link between agent k and ℓ is directed. How to modify the exact diffusion strategy so that it fits into this important scenario is still an open question. One possible solution is to use the push-sum technique [161] to correct the bias incurred by the directed network topology. Another possible solution is to employ the push-pull strategy proposed by [97, 98].

- Exact diffusion is studied for smooth objective functions in this dissertation. However, there are applications that have a composite problem structure that involve both the smooth and non-smooth terms in the objective function. Various algorithms have been proposed to solve the composite distributed optimization problems such as [88, 91, 106, 108]. While convergence of these algorithms is studied in literature, it is still unknown whether exits a distributed algorithm that can solve the composite optimization problem with linear convergence rate. Very recently, it is proved in [109] that a new distributed primal-dual algorithm can converge linearly to the global solution when all agents share the same non-smooth regularization term. This is an encouraging result since it is very common for all agents to have the same regularization under the machine learning setting. Can one prove linear convergence of proximal exact diffusion under the same assumption?

- Exact diffusion is studied for convex objective functions in this dissertation. However, there are applications that have a non-convex problem structure such as deep learning. It is important to answer questions such as whether exact diffusion can escape from

saddle points and converge to a local minimum of the non-convex problem, and whether the collaboration among the agents is beneficial to finding the global solution of the non-convex problem. Some insightful work appear recently that study the performance of diffusion for non-convex optimization, see [162, 163]. These results may help clarify the behavior of exact diffusion for distributed non-convex optimization.

REFERENCES

[1] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[2] S. Pu and A. Nedić, "A distributed stochastic gradient tracking method," in *IEEE Conference on Decision and Control (CDC)*, Miami, FL, Dec. 2018, pp. 963–968.

[3] R. Xin, A. K. Sahu, U. A. Khan, and S. Kar, "Distributed stochastic optimization with gradient tracking over strongly-connected networks," *arXiv:1903.07266*, 2019.

[4] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, April 2014.

[5] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[6] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

[7] S. Kar and J. M. F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 674–690, 2011.

[8] S. Kar, J. M. F. Moura, and K. Ramanan, "Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3575–3605, 2012.

[9] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.

[10] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proc. IEEE Conference on Decision and Control (CDC)*. IEEE, 2005, pp. 6698–6703.

[11] S. Sardellitti, M. Giona, and S. Barbarossa, "Fast distributed average consensus algorithms based on advection-diffusion processes," *IEEE Transactions on Signal Processing*, vol. 58, no. 2, pp. 826–842, 2010.

[12] P. Braca, S. Marano, and V. Matta, "Running consensus in wireless sensor networks," in *Proc. IEEE International Conference on Information Fusion*, Cologne, Germany, 2008, pp. 1–6.

[13] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: convergence analysis and network scaling," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592–606, 2012.

[14] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 205–220, 2013.

[15] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact dffusion for distributed optimization and learning – Part I: Algorithm development," *IEEE Transactions on Signal Processing*, vol. 67, no. 3, pp. 708 – 723, 2019.

[16] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact dffusion for distributed optimization and learning – Part II: Convergence analysis," *IEEE Transactions on Signal Processing*, vol. 67, no. 3, pp. 724 – 739, Feb. 2019.

[17] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su, "Scaling distributed machine learning with the parameter server," in *Proc. Operating Systems Design and Implementation (OSDI)*, 2014, pp. 583–598, Broomfield, Denver, Colorado.

[18] Q. Ho, J. Cipar, H. Cui, S. Lee, J. K. Kim, P. B. Gibbons, G. A. Gibson, G. Ganger, and E. P. Xing, "More effective distributed ML via a stale synchronous parallel parameter server," in *Proc. Advances in neural information processing systems (NIPS)*, Lake Tahoe, NV, 2013, pp. 1223–1231.

[19] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Salt Lake City, UT, 2001, vol. 4, pp. 2033–2036.

[20] L. A. Rossi, B. Krishnamachari, and C. C.J. Kuo, "Distributed parameter estimation for monitoring diffusion phenomena using physical models," in *Proc. IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, Santa Clara, CA, 2004, pp. 460–469.

[21] D. Li, K. D. Wong, Y. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17–29, 2002.

[22] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[23] K. Yuan, Q. Ling, W. Yin, and A. Ribeiro, "A linearized bregman algorithm for decentralized basis pursuit," in *European Signal Processing Conference (EUSIPCO 2013)*. IEEE, 2013, pp. 1–5.

[24] K. Yuan, Q. Ling, and Z. Tian, "A decentralised linear programming approach to energy–efficient event detection," *International Journal of Sensor Networks*, vol. 17, no. 1, pp. 52–62, 2015.

[25] K. Yuan, Q. Ling, and Z. Tian, "Communication-efficient decentralized event monitoring in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2198–2207, 2014.

[26] F. Zeng, C. Li, and Z. Tian, "Distributed compressive spectrum sensing in cooperative multihop cognitive networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 37–48, 2010.

[27] J. J. Meng, W. Yin, H. Li, E. Hossain, and Z. Han, "Collaborative spectrum sensing from sparse observations in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 2, pp. 327–337, 2011.

[28] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, 2007.

[29] K. Zhou and S. I. Roumeliotis, "Multirobot active target tracking with combinations of relative observations," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 678–695, 2011.

[30] S. M. Amin and B. F. Wollenberg, "Toward a smart grid: power delivery for the 21st century," *IEEE Power and Energy Magazine*, vol. 3, no. 5, pp. 34–41, 2005.

[31] C. Ibars, M. Navarro, and L. Giupponi, "Distributed demand management in smart grid with a congestion game," in *Proc. IEEE International Conference on Smart Grid Communications (SmartGridComm),*, Gaithersburg, MD, 2010, IEEE, pp. 495–500.

[32] H. Kim, Y.-J. Kim, K. Yang, and M. Thottan, "Cloud-based demand response for smart grid: Architecture and distributed algorithms," in *Proc. IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Brussels, Belgium, 2011, IEEE, pp. 398–403.

[33] G. B. Giannakis, V. Kekatos, N. Gatsis, S.-J. Kim, H. Zhu, and B W., "Monitoring and optimization for power grids: A signal processing perspective," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 107–128, 2013.

[34] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, 2012.

[35] J. Plata-Chaves, N. Bogdanović, and K. Berberidis, "Distributed diffusion-based lms for node-specific adaptive parameter estimation," *IEEE Transactions on Signal Processing*, vol. 63, no. 13, pp. 3448–3460, 2015.

[36] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks—Part I: Transient analysis," *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3487–3517, 2015.

[37] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 1001–1016, 2015.

[38] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, "A sparsity promoting adaptive algorithm for distributed learning," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5412–5425, Oct. 2012.

[39] H. Raja and W. U. Bajwa, "Cloud k-svd: A collaborative dictionary learning algorithm for big, distributed data," *IEEE Transactions on Signal Processing*, vol. 64, no. 1, pp. 173–188, 2015.

[40] X. Zhao and A. H. Sayed, "Distributed clustering and learning over networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 13, pp. 3285–3300, 2015.

[41] J. Qin, W. Fu, H. Gao, and W. X. Zheng, "Distributed k-means algorithm and fuzzy c-means algorithm for sensor networks based on multiagent consensus theory," *IEEE transactions on cybernetics*, vol. 47, no. 3, pp. 772–783, 2016.

[42] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4129–4144, 2014.

[43] B. Ying, K. Yuan, and A. H. Sayed, "Supervised learning under distributed features," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 977–992, 2018.

[44] J. Liu, M. Chu, and J. E. Reich, "Multitarget tracking in distributed sensor networks," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 36–46, 2007.

[45] X. Zhang, "Adaptive control and reconfiguration of mobile wireless sensor networks for dynamic multi-target tracking," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2429–2444, 2011.

[46] H. Salami, B. Ying, and A. H. Sayed, "Social learning over weakly connected graphs," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 2, pp. 222–238, 2017.

[47] A Nedić, A. Olshevsky, and C. A. Uribe, "Fast convergence rates for distributed non-bayesian learning," *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5538–5553, 2017.

[48] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Proc. International Conference on Machine Learning*, 2018.

[49] L. Cassano, K. Yuan, and A. H. Sayed, "Distributed value-function learning with linear convergence rates," in *European Control Conference (ECC)*. IEEE, 2019, pp. 505–511.

[50] Lucas Cassano, Kun Yuan, and Ali H Sayed, "Multi-agent fully decentralized off-policy learning with linear convergence rates," *Submitted for publication,* Also available as arXiv:1810.07792, 2018.

[51] H.-T. Wai, Z. Yang, P. Z. Wang, and M. Hong, "Multi-agent reinforcement learning via double averaging primal-dual optimization," in *Advances in Neural Information Processing Systems*, 2018, pp. 9649–9660.

[52] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola, "Parallelized stochastic gradient descent," in *Proc. Advances in neural information processing systems (NIPS)*, Vancouver, Canada, 2010, pp. 2595–2603.

[53] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," in *Proc. Advances in neural information processing systems (NIPS)*, Granada, Spain, 2011, pp. 873–881.

[54] O. Shamir, N. Srebro, and T. Zhang, "Communication-efficient distributed optimization using an approximate Newton-type method," in *Proc. International Conference on Machine Learning (ICML)*, Beijing, China, 2014, pp. 1000–1008, Beijing, China.

[55] Y. Zhang and X. Lin, "DISCO: Distributed optimization for self-concordant empirical loss," in *Proc. International conference on machine learning (ICML)*, Lille, France, 2015, pp. 362–370.

[56] C.-P. Lee, C. H. Lim, and S. J. Wright, "A distributed quasi-newton algorithm for empirical risk minimization with nonsmooth regularization," in *Proc. International Conference on Knowledge Discovery & Data Mining (SIGKDD)*. ACM, 2018, pp. 1646–1655, London, United Kingdom.

[57] V. Smith, S. Forte, C. Ma, M. Takac, M. I Jordan, and M. Jaggi, "CoCoA: A general framework for communication-efficient distributed optimization," *Journal of Machine Learning Research*, vol. 18, pp. 230, 2018.

[58] Z. Peng, Y. Xu, M. Yan, and W. Yin, "ARock: an algorithmic framework for asynchronous parallel coordinate updates," *ArXiv e-prints arXiv:1506.02396*, June 2015.

[59] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via alternating direction method of multipliers," *Found. Trends Mach. Lear.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[60] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block ADMM with $o(1/k)$ convergence," *Journal of Scientific Computing*, vol. 71, no. 2, pp. 712–736, 2017.

[61] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *Advances in Neural Information Processing Systems*, Long Beach, CA, 2017, pp. 5330–5340.

[62] X. Lian, W. Zhang, C. Zhang, and J. Liu, "Asynchronous decentralized parallel stochastic gradient descent," in *Proc. International Conference on MachineLearning (ICML)*, Stockholm, Sweden, 2018.

[63] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[64] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2016.

[65] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, Long Beach, CA, 2017, pp. 4424–4434.

[66] et. al. Bonawitz, K., "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019.

[67] L. He, A. Bian, and M. Jaggi, "COLA: Decentralized linear learning," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, Montréal, Canada, 2018, pp. 4536–4546.

[68] A. Koloskova, S. U. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *Proc. International conference on machine learning (ICML)*, Long Beach, CA, 2019.

[69] K. Yuan, B. Ying, J. Liu, and A. H. Sayed, "Variance-reduced stochastic learning by networked agents under random reshuffling," *IEEE Transactions on Signal Processing*, vol. 67, no. 2, pp. 351–366, 2018.

[70] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[71] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks—Part II: Performance analysis," *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3518–3548, 2015.

[72] K. Yuan, S. A. Alghunaim, B. Ying, and A. H. Sayed, "On the influence of bias-correction on distributed stochastic optimization," *Submitted for publication,* Also available as arXiv:1903.10956, 2019.

[73] K. Yuan, B. Ying, S. Vlaski, and A. H. Sayed, "Stochastic gradient descent with finite samples sizes," in *Proc. IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2016, pp. 1–6.

[74] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[75] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[76] B. Ying, K. Yuan, and A. H. Sayed, "Variance-reduced stochastic learning under random reshuffling," *Submitted for publication,* Also available as arXiv: 1708.01383, Aug. 2017.

[77] A. Mokhtari and A. Ribeiro, "DSA: Decentralized double stochastic averaging gradient algorithm," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2165–2199, 2016.

[78] A. Mokhtari, Q. Ling, and A. Ribeiro, "Network newton distributed optimization methods," *IEEE Transactions on Signal Processing*, vol. 65, no. 1, pp. 146–161, 2016.

[79] D. Bajovic, D. Jakovetic, N. Krejic, and N. K. Jerinkic, "Newton-like method with diagonal correction for distributed optimization," *SIAM Journal on Optimization*, vol. 27, no. 2, pp. 1171–1203, 2017.

[80] M. Eisen, A. Mokhtari, and A. Ribeiro, "Decentralized quasi-newton methods," *IEEE Transactions on Signal Processing*, vol. 65, no. 10, pp. 2613–2628, 2017.

[81] D. Jakovetić, J. Xavier, and J. MF Moura, "Fast distributed gradient methods," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1131–1146, 2014.

[82] A. Berahas, R. Bollapragada, N. S. Keskar, and E. Wei, "Balancing communication and computation in distributed optimization," *IEEE Transactions on Automatic Control*, 2018.

[83] H. Li, C. Fang, W. Yin, and Z. Lin, "A sharp convergence rate analysis for distributed accelerated gradient methods," *arXiv preprint arXiv:1810.01053*, 2018.

[84] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5262–5276, 2010.

[85] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Püschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2718–2723, 2013.

[86] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "Linearly convergent decentralized consensus optimization with the alternating direction method of multipliers," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 4613–4617.

[87] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "DLM: Decentralized linearized alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 4051–4064, 2015.

[88] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 482–497, 2015.

[89] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "DQM: Decentralized quadratically approximated alternating direction method of multipliers," *IEEE Transactions on Signal Processing,*, vol. 64, no. 19, pp. 5158–5173, 2016.

[90] Y. Liu, W. Xu, G. Wu, Z. Tian, and Q. Ling, "Communication-censored admm for decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2565–2579, 2019.

[91] W. Shi, Q. Ling, G. Wu, and W. Yin, "A proximal gradient algorithm for decentralized composite optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 22, pp. 6013–6023, 2015.

[92] P. D. Lorenzo and G. Scutari, "NEXT: In-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[93] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.

[94] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *to appear in IEEE Transactions on Control of Network Systems*, 2017.

[95] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes," in *IEEE Conference on Decision and Control (CDC)*, Osaka, Japan, 2015, pp. 2055–2060.

[96] A. Nedić, A. Olshevsky, W. Shi, and C. A. Uribe, "Geometrically convergent distributed optimization with uncoordinated step-sizes," *arXiv:1609.05877*, Sep. 2016.

[97] S. Pu, W. Shi, J. Xu, and A. Nedić, "A push-pull gradient method for distributed optimization in networks," in *Proc. IEEE Conference on Decision and Control (CDC)*, Miami Beach, FL, 2018, IEEE, pp. 3385–3390.

[98] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 315–320, 2018.

[99] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "A decentralized second-order method with exact linear convergence rate for consensus optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 507–522, 2016.

[100] J. Zeng and W. Yin, "ExtraPush for convex smooth decentralized optimization over directed networks," *arXiv:1511.02942*, Nov. 2015.

[101] C. Xi and U. A. Khan, "DEXTRA: A fast algorithm for optimization over directed graphs," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 4980–4993, 2017.

[102] T. Wu, K. Yuan, Q. Ling, W. Yin, and A. H. Sayed, "Decentralized consensus optimization with asynchrony and delays," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 2, pp. 293–307, 2017.

[103] S. A. Alghunaim, K. Yuan, and A. H. Sayed, "Decentralized exact coupled optimization," in *Proc. Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2017, pp. 338–345.

[104] S. A. Alghunaim, K. Yuan, and A. H. Sayed, "Dual coupled diffusion for distributed optimization with affine constraints," in *Proc. IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 829–834.

[105] S. A. Alghunaim, K. Yuan, and A. H. Sayed, "A proximal diffusion strategy for multi-agent optimization with sparse affine constraints," *to appear in IEEE Transactions on Automatic Control*, 2018.

[106] Z. Li, W. Shi, and M. Yan, "A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates," *arXiv:1704.07807*, Apr. 2017.

[107] Z. Li and M. Yan, "A primal-dual algorithm with optimal stepsizes and its application in decentralized consensus optimization," *arXiv preprint arXiv:1711.06785*, 2017.

[108] N. S. Aybat, Z. Wang, T. Lin, and S. Ma, "Distributed linearized alternating direction method of multipliers for composite convex consensus optimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 1, pp. 5–20, 2017.

[109] S. A. Alghunaim, K. Yuan, and A. H. Sayed, "A linearly convergent proximal gradient algorithm for decentralized optimization," *Submitted for publication,* Also available as arXiv:1905.07996, 2019.

[110] K. Seaman, F. Bach, S. Bubeck, Y.-T. Lee, and L. Massoulié, "Optimal algorithms for smooth and strongly convex distributed optimization in networks," in *Proc. International Conference on Machine Learning (ICML)*, Sydney, Australia, 2017, vol. 70, pp. 3027–3036.

[111] C. A Uribe, S. Lee, A. Gasnikov, and A. Nedić, "Optimal algorithms for distributed optimization," *arXiv preprint arXiv:1712.00232*, 2017.

[112] M. Maros and J. Jaldén, "Panda: A dual linearly converging method for distributed optimization over time-varying undirected graphs," in *IEEE Conference on Decision and Control (CDC)*, Miami, FL, Dec. 2018, pp. 6520–6525.

[113] H. Hendrikx, L. Massoulié, and F. Bach, "Accelerated decentralized optimization with local updates for smooth and strongly convex objectives," *arXiv preprint arXiv:1810.02660*, 2018.

[114] S. U. Pillai, T. Suel, and S. Cha, "The Perron-Frobenius theorem: some of its applications," *IEEE Signal Processing Magazine*, vol. 22, no. 2, pp. 62–75, 2005.

[115] X. Zhao, *Learning under Imperfections by Networked Agents*, Ph.D. Dissertation, Electrical Engineering Department, UCLA, Sep. 2014.

[116] P. Whittle, "Equilibrium distributions for an open migration process," *Journal of Applied Probability*, pp. 567–571, 1968.

[117] J. R. Norris, *Markov Chains*, Number 2. Cambridge university press, 1998.

[118] W. K. Hastings, "Monte carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[119] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Transactions on automatic control*, vol. 55, no. 9, pp. 2069–2084, 2010.

[120] A. Nedic and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2015.

[121] D. Needell, R. Ward, and N. Srebro, "Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, Montreal, Canada, 2014, pp. 1017–1025.

[122] P. Zhao and T. Zhang, "Stochastic optimization with importance sampling for regularized loss minimization," in *Proc. International Conference on Machine Learning (ICML)*, Lille, France, 2015, pp. 1–9.

[123] C. Xi and U. A. Khan, "On the linear convergence of distributed optimization over directed graphs," *arXiv:1510.02149*, Oct. 2015.

[124] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge university press, 2004.

[125] R. Fletcher, *Practical Methods of Optimization,*, John Wiley & Sons, New York, NY, USA, second edition, 1987.

[126] Z. J. Towfic and A. H. Sayed, "Adaptive penalty-based distributed stochastic convex optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 15, pp. 3924–3938, 2014.

[127] D. P. Bertsekas, *Nonlinear programming*, Taylor & Francis, 1997.

[128] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms*, John Wiley & Sons, 2013.

[129] Ali Jadbabaie, Asuman Ozdaglar, and Michael Zargham, "A distributed newton method for network optimization," in *IEEE Conference on Decision and Control (CDC)*. IEEE, 2009, pp. 2736–2741.

[130] M. R. Hestenes, "Multiplier and gradient methods," *Journal of optimization theory and applications*, vol. 4, no. 5, pp. 303–320, 1969.

[131] A. Miele, E. E. Cragg, R. R. Lyer, and A. V. Levy, "Use of the augmented penalty function in mathematical programming problems: Part 1," *Journal of Optimization Theory and Applications*, vol. 8, no. 2, pp. 115–130, 1971.

[132] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*, Academic press, 2014.

[133] J. Nocedal and S. Wright, *Numerical optimization*, Springer Science & Business Media, 2006.

[134] R. A. Freeman, P. Yang, and K. M. Lynch, "Stability and convergence properties of dynamic average consensus estimators," in *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE, 2006, pp. 338–343.

[135] M. Zhu and S. Martinez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[136] B. Ying, K. Yuan, and A. H. Sayed, "Dynamic average diffusion with randomized coordinate updates," *to appear in IEEE Transactions on Information and Signal Processing over Networks*, 2019.

[137] F. Saadatniaki, R. Xin, and U. A. Khan, "Optimization over time-varying directed graphs with row and column-stochastic matrices," *arXiv:1810.07393*, 2018.

[138] S. A. Alghunaim, E. K. Ryu, K. Yuan, and A. H. Sayed, "Decentralized proximal gradient algorithms with linear convergence rates," *submitted for publication*, 2019.

[139] Y. Nesterov, *Introductory lectures on convex optimization: A Basic course*, Springer Science & Business Media, NY, 2004.

[140] E. I. Jury, "A simplified stability criterion for linear discrete systems," *Proceedings of the IRE*, vol. 50, no. 6, pp. 1493–1500, 1962.

[141] A. Nedić, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[142] S. Kar and J. M. Moura, "Consensus+ innovations distributed inference over networks: cooperation and sensing in networked systems," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 99–109, 2013.

[143] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[144] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1245–1260, 2018.

[145] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu, "D2: Decentralized training over decentralized data," in *Proc. International Conference on Machine Learning (ICML)*, Stockholm, Sweden, 2018, pp. 1 – 8.

[146] Z. J. Towfic and A. H. Sayed, "Stability and performance limits of adaptive primal-dual networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2888–2903, 2015.

[147] X. Mao, K. Yuan, Y. Hu, Y. Gu, A. H. Sayed, and W. Yin, "Walkman: A communication-efficient random-walk algorithm for decentralized optimization," *Submitted for publication. Also available at arXiv:1804.06568*, Apr. 2018.

[148] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. International Conference on Computational Statistics* (COMPSTAT), pp. 177–186. Springer, Paris, 2010.

[149] M. Schmidt, N. Le Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," *Mathematical Programming*, vol. 162, no. 1, pp. 83–112, Mar. 2017.

[150] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. Advances in Neural Information Processing Systems* (NIPS), Lake Tahoe, NV, 2013, pp. 315–323.

[151] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Proc. Advances in Neural Information Processing Systems* (NIPS), Montreal, Canada, 2014, pp. 1646–1654.

[152] M. Li, D. G. Andersen, A. J. Smola, and K. Yu, "Communication efficient distributed machine learning with the parameter server," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, Montréal, Canada, 2014, pp. 19–27.

[153] M. Jaggi, V. Smith, M. Takác, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan, "Communication-efficient distributed dual coordinate ascent," in *Proc. Advances in Neural Information Processing Systems* (NIPS), Montréal, Canada, 2014, pp. 3068–3076.

[154] J. D. Lee, Q. Lin, T. Ma, and T. Yang, "Distributed stochastic variance reduced gradient methods and a lower bound for communication complexity," *arXiv:1507.07595*, Jul. 2015.

[155] C. Hardy, E. L. Merrer, and B. Sericola, "Distributed deep learning on edge-devices: feasibility via adaptive compression," *arXiv:1702.04683*, Feb. 2017.

[156] B. Ying and A. H. Sayed, "Performance limits of stochastic sub-gradient learning, Part I: Single agent case," *Signal Processing*, vol. 144, pp. 271–282, Mar. 2017.

[157] B. Ying and A. H. Sayed, "Performance limits of stochastic sub-gradient learning, Part II: Multi-agent case," *Signal Processing*, vol. 144, no. 253-264, Mar. 2017.

[158] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "A decentralized second-order method with exact linear convergence rate for consensus optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 507–522, 2016.

[159] L. Xiao and T. Zhang, "A proximal stochastic gradient method with progressive variance reduction," *SIAM Journal on Optimization*, vol. 24, no. 4, pp. 2057–2075, Dec. 2014.

[160] B. Ying, K. Yuan, S. Vlaski, and A. H. Sayed, "Stochastic learning under random reshuffling with constant step-sizes," *IEEE Transactions on Signal Processing*, vol. 67, no. 2, pp. 474–489, 2018.

[161] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *IEEE Symposium on Foundations of Computer Science*. IEEE, 2003, pp. 482–491.

[162] S. Vlaski and A. H Sayed, "Distributed learning in non-convex environments–part i: Agreement at a linear rate," *Submitted for publication,* Also available as arXiv:1907.01848, 2019.

[163] S. Vlaski and A. H. Sayed, "Distributed learning in non-convex environments–part i: Agreement at a linear rate," *Submitted for publication,* Also available as arXiv:1907.01848, 2019.