

REPRESENTATION LEARNING IN DISTRIBUTED NETWORKS

by

ARPITA GANG

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Prof. Waheed U. Bajwa

And approved by

New Brunswick, New Jersey

October, 2022

ABSTRACT OF THE DISSERTATION

Representation Learning in Distributed Networks

By ARPITA GANG

Dissertation Director:

Prof. Waheed U. Bajwa

The effectiveness of machine learning (ML) in today's applications largely depends on the *goodness* of the representation of data used within the ML algorithms. While the massiveness in dimension of modern day data often requires lower-dimensional data representations in many applications for efficient use of available computational resources, the use of uncorrelated features is also known to enhance the performance of ML algorithms. Thus, an efficient representation learning solution should focus on dimension reduction as well as uncorrelated feature extraction. Even though Principal Component Analysis (PCA) and linear autoencoders are fundamental data preprocessing tools that are largely used for dimension reduction, when engineered properly they can also be used to extract uncorrelated features. At the same time, factors like ever-increasing volume of data or inherently distributed data generation impede the use of existing centralized solutions for representation learning that require availability of data at a single location. This thesis focuses on representation learning in the case of distributed data. Specifically, it tackles the distributed feature learning problem when data samples are scattered in an arbitrarily connected network. PCA, being the most widely used representation learning tool, is the main focus of this thesis. The overall objective is solving the distributed PCA problem in 1) batch setting, and 2) streaming settings. The goals include development of algorithms that are both provably convergent at an optimal rate and also efficient in terms of communication between nodes in the network.

Two novel algorithms for distributed PCA are proposed in the batch setting that converge

at a linear rate to the true eigenvectors of the global covariance matrix. The first proposed algorithm is called Distributed Sanger’s Algorithm (DSA), which is a one-time scale method that converges to a neighborhood of the true eigenvectors of the covariance matrix of the data distributed in an arbitrarily connected network. Although this algorithm is fast in convergence, it reaches to only a neighborhood of the optimal solution. We propose a second algorithm called FAST-PCA (Fast and exAct diSTributed PCA) that uses a gradient-tracking based technique to converge linearly but also exactly to the true solutions. Extensive theoretical analysis is provided for both the algorithms that prove their convergence and numerical results are also presented that further show the efficacy of the methods.

An important aspect of the modern world data is its ever increasing volume and thus ML algorithm in modern applications should be capable of adapting to new data. This motivates the study of representation learning problem in the streaming data setting as well. The problem of distributed PCA in the streaming data case is tackled in the second part of this thesis. To that end, an algorithm called Distributed Generalized Oja’s Algorithm (DIEGO) is proposed that estimates multiple dominant eigenvectors of the population covariance matrix in the streaming settings. Theoretical analysis of the algorithm is provided that proves its convergence. In streaming data settings, the distributed PCA problem is also looked into and analysed in a special case of the distributed network, namely the federated learning setup, which has a master-slave architecture rather than an arbitrary mesh network. Numerical results to study the effect of different parameters are presented to demonstrate the efficiency of the algorithm in both distributed and federated settings. For the purpose of a neural network based representation learning model, the algorithms are also implemented for training autoencoders with linear activation units.

Acknowledgements

“If you want to cultivate a habit, do it without any reservation, till it is firmly established. Until it is so confirmed, until it becomes a part of your character, let there be no exception, no relaxation of effort.” - Lord Mahavir

The journey of being a PhD student required a lot dedication and consistent effort. Fortunately, I have so many people in and around my life who led by example and showed me the value of passion, honesty and hard work. I would like to thank all those who have helped me, in one way or the other, to get me where I am today.

First and foremost, I would like to thank my advisor, Prof. Waheed Bajwa for his constant guidance, support, patience and encouragement in the past five years. I am especially grateful to him for teaching me how to be persistent and persevere in research, and also more importantly to believe in myself. Since the beginning, I am constantly amazed with his extraordinary perceptive abilities and attention to detail. Our meetings and discussions have been always productive and insightful. Additionally, I would also like to thank him for supporting me financially during my years at Rutgers.

I would also like to thank other members of my dissertation committee, Prof. Emina Soljanin, Prof. Anand Sarwate, Prof. Shirin Jalali and Prof. Anna Scaglione for taking out the time to be a part of my committee, and their numerous helpful suggestions. I would also like to thank professors at Rutgers University, Prof. Roy Yates, Prof. Salim El Rouayheb, Prof. Predrag Spasojevic, Prof. Emina Soljanin, for the wonderful courses they taught. The administrative staff of the ECE department at Rutgers also deserves a special mention. Christy, John, Chris, Arletta - all have been so wonderful and helpful throughout these years. Thank you to all of you.

I would like to thank my lab members and friends in New Brunswick who made the past few years fun and easier. Specifically, I would like to thank Talal, Zahra and Haroon for helping me during my initial years at Rutgers and making me feel so welcomed in the lab. I would also

like to thank Asad and Batoul who have been such wonderful friends in the last couple of years and helped me cope with the pandemic.

I learned what friendship is from few people back in India. To all of you (you know who you are) thank you for making me a better person, for teaching me patience, for not letting me fall down and for always picking me up when I do. Himanshi and Ankita - you two deserve special mention. Through thick and thin, good and bad you two always stood by me.

Nothing in my life is possible without my family- my parents, my grandparents, my brother and my soon-to-be husband. My grandparents have been the epitome of strength and hardwork. Through every difficult phase, they taught me to never give up, to keep head high and just keep going. Give your best, leave the rest - they are living examples of that. Maa, Dada - thank you for being you. My brother Nishant is my daily dose of laughter and brings smile to my face in the strangest ways. Thank you for making my life easier and listening to rants even when they are senseless. My parents define selfless love and sacrifice for me. For all that you do for, I can never thank you enough. My simple, sweet mother constantly worries about me, prays for me and takes care of me even from across the globe. I don't know how she does it, but that's mother's love; one can't understand it. My father, who everyone says I am an exact copy of, understands me even on days I can't understand myself. I don't even have to finish my sentence and he knows just how to help me. No one else on this planet can do that. Every good or bad news, he is the one I want to call first. I can't list everything here (because the list will never end) but Mummy, Papa - thank you for everything. Lastly, to the love of my life, Umang - you are the most patient and sweetest man I have ever met. Thank you for being a constant support and cheerleader in my life.

Dedication

To my parents. Everything I am, everything I will ever be, I owe it to you two.

Table of Contents

Abstract	ii
Acknowledgements	iv
Dedication	vi
List of Figures	1
List of Tables	3
1. Introduction	4
1.1. Motivation	4
1.2. Representation learning	5
1.2.1. Principal Component Analysis	7
1.2.2. Autoencoders	8
1.3. Overview and Contributions	8
2. Review of Existing Algorithms	11
2.1. Centralized PCA	11
2.2. Distributed and Decentralized PCA	13
3. Distributed Sanger’s Algorithm: A Linearly Converging Algorithm for Dis- tributed PCA	16
3.1. Introduction	16
3.1.1. Our Contributions	17
3.2. Problem Formulation	18
3.3. The Proposed Algorithm	21
3.4. Convergence Analysis	24
3.4.1. Convergence Analysis of a Modified GHA	24

3.4.2.	Convergence Analysis of Distributed Sanger’s Algorithm (DSA)	26
3.5.	Statements and Proofs of Auxiliary Lemmas	31
3.5.1.	Statement and Proof of Supporting Lemma for Modified GHA	31
3.5.2.	Statement and Proof of supporting Lemma for DSA	39
3.6.	Experimental Results	45
3.6.1.	Synthetic Data	46
3.6.2.	Real-World Data	49
3.7.	Conclusion	50
4.	FAST-PCA: A Fast and Exact Algorithm for Distributed PCA	51
4.1.	Introduction	51
4.1.1.	Our Contributions	52
4.2.	Proposed Algorithm	53
4.3.	Convergence Analysis of Auxiliary Results	55
4.3.1.	Convergence Analysis of a Modified GHA	55
4.3.2.	Convergence Analysis of a Modified Krasulina	56
4.4.	Main Results	58
4.5.	Statements and Proofs of Supporting Lemmas	71
4.5.1.	Statement and Proof of Supporting Lemma for Modified Krasulina	71
4.5.2.	Statement and Proof of Supporting Lemma for FAST-PCA	72
4.6.	Experimental Results	77
4.6.1.	Synthetic Data	78
4.6.2.	Real-World Data	79
Even Distribution of Data	80	
Uneven Distribution of Data	80	
4.6.3.	Autoencoder Implementation	81
4.7.	Conclusion	82
5.	DIEGO: Distributed PCA in Streaming Data Settings	83
5.1.	Introduction	83
5.1.1.	Our Contributions	84
5.2.	Problem Description	84
5.3.	Proposed Algorithm	85

5.4. Convergence Analysis	87
5.5. Distributed PCA in a Federated Learning Setup	99
5.5.1. Problem Setup	99
5.5.2. Convergence Analysis	99
5.6. Numerical Results	106
5.6.1. Synthetic Data	107
Effect of Graph Connectivity	107
Effect of Eigengap	108
Comparison with other methods	108
5.6.2. Real World Data	108

List of Figures

1.1. Types of network architectures	6
1.2. Pictorial Depictions	8
1.3. Types of Representation Learning	9
3.1. The role of collaboration in the distributed PCA problem and the effect of changing the step size on the performance of DSA. The distributed setup corresponds to an Erdos–Renyi graph ($p = 0.5$) with $M = 10$ nodes, while the dimension of data is $d = 10$ and the number of estimated eigenvectors is $K = 3$	46
3.2. Comparison between the performances of DSA, DPGD and SeqDistPM for $K = 1$ and $\Delta_K = 0.8$ in terms of communications efficiency, i.e., decrease in average estimation error as a function of the number of data units communicated throughout the network.	47
3.3. Comparison between DSA, DPGD, and SeqDistPM for $K = 5$ in terms of communications efficiency.	48
3.4. Comparison between DSA, OI, GHA, and DPGD for MNIST dataset as a function of the number of algorithmic iterations.	49
3.5. Comparison between DSA, OI, GHA, and DPGD for CIFAR-10 dataset as a function of the number of algorithmic iterations.	49
4.1. Performance comparison of FAST-PCA with various algorithms for two different eigengaps.	79
4.2. Performance comparison of FAST-PCA with various algorithms for two different eigengaps and two graph topologies in the case of (almost) equal eigenvalues. . .	80
4.3. Performance comparison of FAST-PCA with various algorithms for MNIST and CIFAR10	80
4.4. Performance comparison of FAST-PCA-O/K for even and uneven distribution of MNIST data	81

5.1. Effect of different parameters on the performance of DIEGO for $K = 1$	107
5.2. Performance comparison of DIEGO with centralized Oja and the Distributed Oja in federated learning setting in a network of $M = 20$ nodes.	108
5.3. Performance comparison of DIEGO with centralized Oja and the Distributed Oja in federated learning setting in a network of $M = 100$ nodes.	109
5.4. Performance comparison of DIEGO with centralized Oja and the Distributed Oja in federated learning setting while estimating $K = 5$ eigenvectors.	109
5.5. Performance comparison of DIEGO with centralized Oja and the Distributed Oja in federated learning setting while estimating $K = 5$ eigenvectors for MNIST dataset.	110
5.6. Performance comparison of DIEGO with centralized Oja and the Distributed Oja in federated learning setting while estimating $K = 5$ eigenvectors for Higgs dataset.	110

List of Tables

2.1. Comparison of Communication and Iteration Cost	15
4.1. Comparison of Communication and Iteration Cost	71
4.2. Comparison of Classification accuracy	82

Chapter 1

Introduction

1.1 Motivation

The modern world is a data-driven era as massive, high-dimensional datasets are becoming an increasingly essential part of nearly every aspect of our lives, ranging from healthcare to finance and from social media to the Internet-of-Things (IoT). In a related trend, machine learning algorithms are finding their applications in every possible domain because of their data-driven approaches and the ability to generalize to new unseen data. These machine learning algorithms require considerable amount of data preprocessing for their efficient and effective use. For these reasons, a lot of effort have been put over in the years in efficient data preprocessing such that the resultant representation of the data can be used in downstream machine learning algorithms like classification [1], prediction [2] etc. One of the major steps in this preprocessing is dimension reduction and feature learning for compression and extraction of useful features from raw data. Some of the most used techniques for dimension reduction and feature learning are principal component analysis (PCA) [3], linear discriminant analysis (LDA) [4], restricted Boltzmann machines (RBM) [5], autoencoders [6] etc. Moreover, the performance of machine learning methods is heavily dependent on the choice of data representation (or features) on which they are applied. Learning features from the data that embody the most important, explanatory and distinguishing information is an essential requirement of representation learning of the data. It has been argued in literature e.g., in [7], that one of the factors that makes a representation (of a data sample) “good” is having disentangled, more commonly known as uncorrelated, features in the learned representation. Different explanatory features of the data tend to change independently of each other in the input distribution in the real-world inputs.

This implies that if the learned representations have uncorrelated features, changes or noise in one will not affect the others. Hence, dimension reduction in a computationally efficient manner such that the resulting features are uncorrelated have been an active area of research in signal processing and machine learning domains. However, the continuous explosion in the volume (number of samples) and dimension (number of features) of data presents new challenges to the prevalent methods of representation learning.

Another aspect of this modern day data is it being inherently distributed geographically across locations in cases such as Internet of Things (IoT) [8], smart cities [9], autonomous vehicles, etc., where existing ways of representation learning are not directly applicable. Modern world data tends to be distributed for a multitude of reasons; it can be inherently distributed (e.g., in IoT) where privacy or communication bottleneck prevents collation of data at a single location, or it can be distributed due to storage and/or computational limitations. Distributed setups can be largely classified into two types: i) those having a central entity/server that coordinates with the other nodes in a master-slave architecture like in parallel computing or federated learning, and ii) those lacking any central entity, in which the nodes are connected in an arbitrary network. Figure 1.1 pictorially depicts the two architectures. Although the terms distributed and decentralized are used interchangeably for both the setups in literature, we call the former scenario *decentralized* and the latter *distributed* in this thesis. In the decentralized architectures, the nodes communicate only with a master node which aggregates information from all the nodes and passes it back to them. Hence the network topology doesn't play a role in such solutions. On the other hand, distributed architectures are more general, lacking any master node. In such networks, the nodes communicate with each other and reach a solution through mutual collaboration. Both these scenarios are prevalent in the world, but the distributed case is more general as it encompasses all arbitrary network structures. The absence of a central server has further advantages like absence of single point of failure, no communication bottleneck at the central server etc. Motivated by these reasons, this thesis focus on the problem of dimension reduction and uncorrelated/disentangled representation learning in a distributed network.

1.2 Representation learning

Simply put, representation learning means learning a good representation of the raw features of the data that can be used efficiently by downstream machine learning algorithms. There can be many attributes of representation learning and few of those are discussed below:

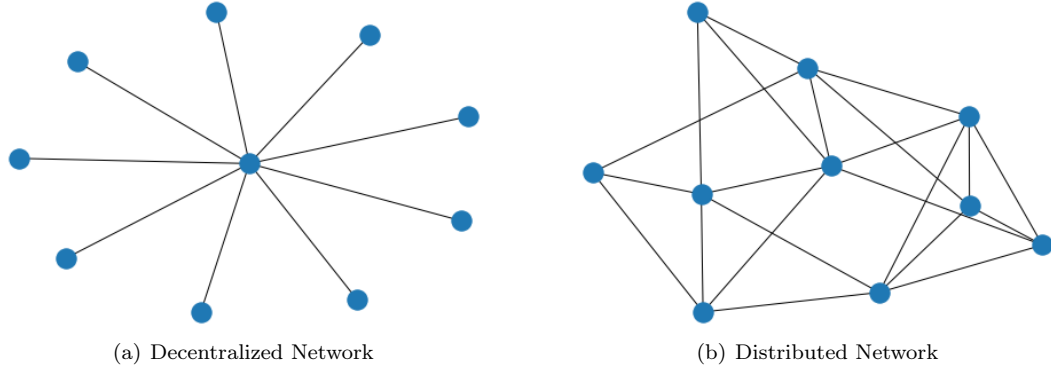


Figure 1.1: Types of network architectures

1. Tackling curse of dimensionality: The raw input feature space of data is usually very high-dimensional, but data compression and efficient processing in the modern world is based on the proposition that relevant information in the data actually lies near or within low dimensional spaces. A good representation of data is the one that captures this relevant low dimensional information and hence cures this curse of high dimension that comes with the explosion of data.
2. Abstraction: More and more abstract features can be learnt in terms of less abstract features in layered fashion using certain deep architectures [10]. The abstraction captures certain complex features of data that are invariant to local changes in the input and have higher predictive power.
3. Transferable representations: Another aspect of a good representation is its re-usability. This means if the features learned for a particular application can be used with little or no modification for another application e.g., word embeddings [11].
4. Uncorrelatedness/Disentanglement: Learning uncorrelated representations have also gained attraction in feature learning. Correlated features bring redundant information and in turn lead to unnecessary increase in dimension of learned representations. This ultimately can have consequences in downstream machine learning models e.g., random forests can be good at detecting interactions between different features, but highly correlated features can mask these interactions.

A good feature representation will have a combination of the above attributes, not necessarily all. In this thesis, we will talk about the representations that have reduced dimension (that tackle the curse of dimensionality) and uncorrelated (disentangled) features. Two of the most widely used representation learning tools are Principal Component Analysis (PCA) and autoencoders.

1.2.1 Principal Component Analysis

Principal Component Analysis (PCA) transforms a large set of correlated features to a smaller set of uncorrelated features that contain maximum information of the raw data. The goal of dimension reduction can be accomplished by learning a low-dimensional subspace spanned by the dominant eigenvectors of the covariance matrix of the distribution to which the data samples belong. Mathematically speaking, for a data point $\mathbf{y} \in \mathbb{R}^d$ sampled from a distribution with zero mean and covariance $\Sigma \in \mathbb{R}^{d \times d}$, dimension reduction can be achieved by projecting \mathbf{y} onto a matrix $\mathbf{X} \in \mathbb{R}^{d \times K}$, $K \ll d$, such that \mathbf{X} spans a subspace spanned by the leading K eigenvectors of Σ under the constraint $\mathbf{X}^T \mathbf{X} = \mathbf{I}$, that is \mathbf{X} lies on a Stiefel manifold. When \mathbf{y} is compressed as $\tilde{\mathbf{y}} = \mathbf{X}^T \mathbf{y}$ with such an \mathbf{X} , its reconstruction $\mathbf{X} \mathbf{X}^T \mathbf{y}$ has minimum error in Frobenius norm sense. However, this approach can only be called *principal subspace analysis* as it does not ensure that the resultant K features in $\tilde{\mathbf{y}}$ are uncorrelated. The uncorrelatedness constraint requires $\mathbb{E} [\tilde{\mathbf{y}} \tilde{\mathbf{y}}^T] = \mathbb{E} [\mathbf{X}^T \mathbf{y} \mathbf{y}^T \mathbf{X}]$ to be a diagonal matrix, which is fulfilled only when \mathbf{X} contains the eigenvectors of Σ , not just any orthogonal basis of the subspace spanned by the said eigenvectors.

As explained above, the true and complete purpose of PCA is served when the search for the optimal solution ends with the specific set of eigenvectors of the data covariance matrix, and not just with the subspace it spans. That is, PCA aims to find the specific directions in which maximum energy of the data lies (see Figure 1.2(a)). Even though the problem of dimensionality reduction of data has many optimal solutions (corresponding to all the sets of basis vectors spanning the K -dimensional space), our goal is to find only the ones that give the eigenvectors as the basis. In terms of optimization, geometry of the PCA problem in which one tries to minimize the mean-squared reconstruction error under an orthogonality constraint, it is a non-convex strict-saddle function. In a strict-saddle function, all the stationary points except the local minima are strict saddles wherein the Hessians have at least one negative eigenvalue that helps in escaping these saddle points. Also, in the case of PCA the local minima are the same as the global minima. These geometric aspects make PCA, despite being non-convex, a “nice and solvable” problem whose optimal solution can be reached efficiently. However, note that the set of global minima contains, along with the set of eigenvectors as basis, all other possible bases that are rotated with respect to the eigenvectors. And our goal is not to find just any of the global minima but to look into a very particular subset of it, where the basis is not rotated. Thus, the true purpose of PCA is fulfilled by a specific element of the Stiefel manifold that corresponds to the eigenvectors of Σ .

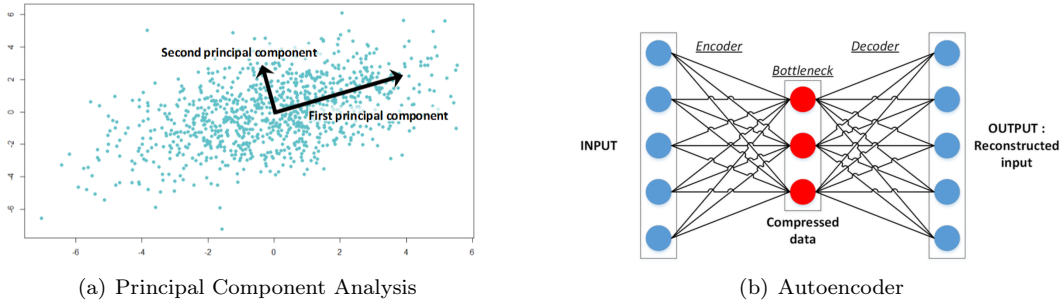


Figure 1.2: Pictorial Depictions

1.2.2 Autoencoders

Autoencoder is another popular neural-network based tool for representation learning. It is an unsupervised artificial neural network that learns how to efficiently compress and encode data so as to reconstruct the data back from the reduced encoded representation to a version that is as close to the original input as possible. Like PCA, autoencoders were originally viewed as a dimensionality reduction technique and thus used a “bottleneck” layer whose dimension is smaller than the input dimension as shown in Figure 1.2(b). This bottleneck layer represents the “code” or the lower dimensional representation of the input data. A study in [6] showed that the optimum weights of an linear autoencoder, when the loss function is the reconstruction error, are given by the subspace spanned by the eigenvectors of the input covariance matrix. It was further shown in [12, 13] that with a different training algorithm based on the Hebbian rule [14], the network weights will converge to the eigenvectors of the input correlation matrix not just the subspace spanned by them. That is, the encoding from an autoencoder will be uncorrelated and hence will be a “better” representation. The good generalization power of neural network-based systems along with their ease of parallelization in case of massive data make them very attractive and efficient solutions for representation learning.

1.3 Overview and Contributions

Figure 1.3 depict some of the representation learning methods that exist in literature. The theme of this thesis is based on principal component analysis (PCA) in distributed settings. Our focus is to develop solutions for distributed PCA when the data samples are scattered across an arbitrarily connected network with no central node. While PCA is often reduced to dimension reduction, we focus on the dual goal of PCA that requires dimensionality reduction as well as feature decorrelation. Main contributions of this thesis include proposed two novel

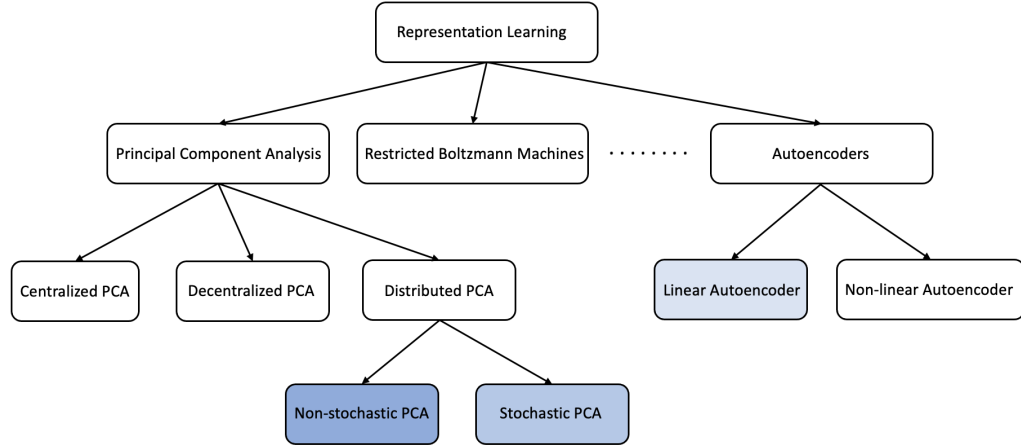


Figure 1.3: Types of Representation Learning

algorithms for distributed PCA in deterministic settings when the data is fixed at each node, one algorithm for distributed PCA in stochastic setting for the case of online data, application of our algorithm for distributed autoencoder training and its effect on some classification tasks. The details of these contributions are:

1. In Chapter 3, we propose our first algorithm for distributed PCA. We consider a connected network lacking any central server where nodes can communicate with their neighbors. Oja's method is one of the classical methods based on the Hebbian learning rule for estimation of the dominant eigenvector in the case of streaming data. We use its generalization by Sanger for our case of distributed data in non-streaming settings. Since each node holds a part of the sample while we are trying to learn eigenvectors for the entire data (global data), consensus is an important part for our solution. All distributed solutions that exist in literature are two time-step algorithms that use multiple rounds of communications between neighbors to achieve this consensus. We propose a one time-scale algorithm called the distributed Sanger's algorithm (DSA) [15, 16] that converges to the true eigenvectors of the global covariance matrix in a fast and efficient way. We provide extensive analysis of our algorithm and proof that our solution converges, upto a neighborhood, to the true eigenvectors of the global covariance matrix at a linear rate for a constant step size. Further we provide numerical results on synthetic and real data to prove the efficacy of the solution.
2. In Chapter 4, we continue our focus on distributed PCA in non-streaming settings. Our

first solution, in the previous chapter, only reaches a neighborhood of the true solution when a constant step size is used. It can however be shown that it converges exactly by using decreasing step size, but that would mean compromising the rate. Hence we lose the advantage of being fast. In short, DSA is fast but not exact. To mitigate that, we propose another algorithm called Fast and exAct diSTributed PCA (FAST-PCA) [17] based on the gradient-tracking approach. We propose two variants of the algorithm FAST-PCA-O and FAST-PCA-K motivated from the Oja’s and Krasulina’s rule respectively. Like Oja’s, Krasulina’s method is another classical algorithm for dominant eigenvector estimation in streaming data case. Although its extension for multiple eigenvector estimation does not exist. We propose a generalization of Krasulina’s rule for multiple eigenvector estimation in the distributed case in FAST-PCA-K. Our proposed FAST-PCA method is an iterative update algorithm and its main attributes are that it is fast since it lacks any explicit consensus loop and hence reduces the communication overhead, and it converges exactly to the true eigenvectors of the global covariance matrix at a linear rate, thereby filling the gap of inexact convergence left by DSA. Through extensive analysis of both versions of our algorithm, we show that our proposed methods converge to the true eigenvectors of the global covariance matrix exactly and at a linear rate. We also provide numerical results to support our claims.

3. In Chapter 5, we shift our focus to the case of streaming data for the distributed PCA problem. The increasing volume of data makes the case of streaming and distributed PCA a very relevant problem. In this setting, we propose an algorithm called DIStributEd Generalized Oja’s Method (DIEGO) which is a distributed version of Oja’s algorithm coupled with an orthogonalization technique to estimate top K eigenvectors of the population covariance matrix. We show through our analysis for the case of $K = 1$ that the proposed algorithm indeed converges to the dominant eigenvector. In addition, we show that in a different distributed setup, like a federated learning setup, the algorithm has a faster convergence than centralized Oja’s algorithm. Extensive numerical experiments are provided to study the effect of various parameters like eigengap, graph connectivity on the performance of the proposed algorithm. We also compare the performance of our algorithm with the centralized PCA solution that clearly shows an increase in the rate of convergence.

Chapter 2

Review of Existing Algorithms

Data compression and representation has been an active area of research for decades. The explosion of machine learning in so many applications have only added to its importance. Besides machine learning, the nature of growing data also encouraged the study and development of stochastic algorithms for PCA. Additionally, problems focused on finding solutions in decentralized and distributed settings needed different approaches to PCA and representation learning in general. In this chapter, we provide an overview of the existing works of PCA in different settings.

2.1 Centralized PCA

Non-stochastic PCA: The problem of dimension reduction goes back to as early as 1901 when Pearson [18] aimed to fit a line to a set of data points. Later, Hotelling [3] proposed a PCA method for decorrelating and compressing a set of data points by finding their principal components. Since then, many iterative methods like power method, orthogonal iterations [19], Lanczos method [20] have been proposed to estimate eigenvectors or low-dimensional subspaces of symmetric matrices, a class under which covariance matrices fall. These methods are useful when all the data is stored at a single location.

Stochastic PCA: In case of large or growing datasets, it might be difficult to store the data at a single location or make a pass over the whole data at once. To alleviate this issue, over the past several decades significant attention has been given and research has been done to come up with stochastic algorithms for PCA under the assumption that the data has reasonable statistical properties. A stochastic approximation algorithm was proposed by Krasulina in [21] for the

estimation of the dominant eigenvector in the streaming data case. From the point of view of training neural networks for data compression, an algorithm very similar to Krasulina’s method was later proposed by Oja [12], which was based on the Hebbian rule [14] which has its roots in neurobiology. Krasulina’s and Oja’s method are the earliest work for stochastic PCA. Oja’s rule was also extended for multiple eigenvector estimation by Sanger [13] who combined Hebbian rule with Gram-Schmidt orthogonalization. It was shown that the weights of an autoencoder trained using this Hebbian rule converge to the eigenvectors of the input correlation matrix. The convergence of all these algorithms used ideas from stochastic approximation methods [22] and were proven to converge asymptotically when decreasing step sizes that converge to zero were used. A different perspective on convergence of Oja’s and Sanger’s rule was later given in [23, 24]. These works used a deterministic discrete time approach to prove convergence in case of constant step sizes. Krasulina’s method was generalized for the estimation of a subspace of dimension greater than one in [25], although it only guarantees convergence to the principal subspace, instead of principal components, at a linear rate under the low-rank matrix assumption.

Recently, some work has been done for obtaining the finite sample complexities i.e., non-asymptotic guarantees, for the PCA problem in stochastic settings. The work in [26] use variance reduction techniques for faster convergence. However, it requires multiple passes over the data, which makes them not completely fit for fast streaming settings. Another work in [27] uses mini-batching along with an added momentum term for faster convergence. The analysis in both of these works require an initialization close to the true eigenvectors, which is also not ideal in practice. With random initialization, Balasubramani et. al [28] provided finite sample guarantees for Krasulina’s and Oja’s method. Several other methods have been proposed in the recent years like [29, 30] etc. for the PCA problem in case of noisy or spiked gaussian covariance matrices. The authors in [31] proved that the sample guarantees for Oja’s rule can achieve the optimal sample complexity for the first eigenvector, in the sense that it matches the matrix Bernstein inequality [32, 33]. The authors in [34] extended this analysis in [31] to mini-batch settings as well as for subspace estimation. Among other works, [35] also achieve this optimality and provide eigengap-free convergence guarantees for Oja’s rule in case of subspace estimation without taking the variance of data samples into account.

PCA as an Optimization Problem: One approach towards solving the PCA problem in streaming settings is to relax the non convex PCA problem to a convex optimization problem and then use stochastic gradient descent to solve the resulting stochastic convex optimization

problem [36]. Taking this approach immediately opens up the rich literature for solving stochastic convex problems, but it comes with tradeoffs. One tradeoff is the requirement $\mathcal{O}(d^2)$ storage space in case of d dimensional data as opposed to $\mathcal{O}(dk)$ when we solve the PCA problem in its original nonconvex form. Secondly, the relaxation allows to only estimate a k dimensional subspace instead of the true leading k eigenvectors, thus preventing the learning of uncorrelated features. Due to these limitations, PCA is often solved in its non convex form. Some nonconvex optimization methods have also been developed in recent years that can be used to tackle the PCA problem in the presence of streaming data. In [37] the PCA problem is solved as an optimization program over the Grassmannian manifold. However, the resulting analysis relies on the availability of a good initial guess. In contrast, the authors in [38] analyze the use of the SGD for solving certain nonconvex problems that include subspace tracking. Subspace tracking can solve the principal subspace analysis problem in streaming data case, but solving PCA completely would require different approach. Their analysis also requires the step size to be significantly small for eventual convergence which implies slower convergence in practice.

2.2 Distributed and Decentralized PCA

The main focus of this thesis is distributed networks. As such, we briefly touch upon decentralized solutions for PCA and talk more elaborately about existing distributed PCA solutions.

Decentralized PCA: The solutions for the PCA problem in this setting focus on computation reduction at a single node and are useful when data is large and can be simply split between machines in contrast to distributed settings, where the assumption is data is scattered inherently and can be coordinated by a single node. Some of the decentralized PCA solutions in case of non-stochastic settings are [39, 40]. In stochastic settings, some of the methods that exist for decentralized PCA were proposed in [41, 42].

Distributed PCA: In any distributed network, data can be distributed in either of the two ways i) by features, where each node in the network has access to a subset of the features that comprise a data point, and ii) by samples where nodes holds full data points. The solutions for these two data distribution types are significantly different. A detailed review of various distributed PCA algorithms for both kinds of data distribution is done in [43]. For the case of feature-wise distribution as in [44–46], each node in the network estimates one or a subset of features of the entire subspace. The work in [44] estimates top K eigenvectors of the graph adjacency matrix of a network, while another significant work in [45] proposed an algorithm for estimation of top K eigenvectors of the covariance matrix sequentially starting from the

eigenvector corresponding to the largest eigenvalue. This sequential approach slows down the convergence of the algorithm when a higher-dimensional eigenspace needs to be estimated. Another work by us in [47] did away with sequential estimation and proposed an orthogonal iteration based method for simultaneous estimation of the dominant k dimensional subspace, albeit not eigenvectors, using a distributed QR algorithm [48].

In this thesis, we focus on the case of sample-wise data distribution, where each node estimates the entire basis and consensus in the network is a necessary condition. The sample-wise data distribution was considered in [49–51], where a power method-based approach was proposed for estimation of the dominant eigenvector ($K = 1$). This method requires an explicit consensus loop [52] in every iteration of the power method and the final error is a function of the number of consensus iterations. The power method-based distributed PCA solutions can be used for multiple ($K > 1$) eigenvector estimation in a sequential manner, where lower-order eigenvectors are estimated using the residue of the covariance matrix left after its projection on the higher-order eigenvectors. Since estimation of any lower-order eigenvector requires that the higher-order eigenvectors are fully estimated, this sequential approach results in a rather slow algorithm. To overcome the issues of the sequential approach, an orthogonal iteration-based solution (S-DOT) for the case of $K > 1$ was proposed in [47]. Although this method estimates the K -dimensional subspace simultaneously, its convergence guarantees are in terms of subspace angles and thus it proves convergence to the principal subspace. Moreover, all these aforementioned methods require an explicit consensus loop making these algorithms inefficient in terms of communication overhead. Another recent paper on distributed PCA [53] used a gradient-tracking idea to develop a two-time scale algorithm called DeEPCA for subspace estimation. The algorithm proposed in this work is, however, also a two-time scale method based on orthogonal iterations, although they removed the dependence of the number of consensus iterations on the final error thereby reducing the number of communication required. However, their convergence guarantees were for subspace estimation, not eigenvectors, thereby making it a PSA algorithm effectively. Table 2.1 shows the convergence rates of these important algorithms for the case of sample-wise distributed data in non-streaming settings. The table provides a comparison of the communication and iteration complexities of various distributed PCA (principal component analysis) and PSA (principal subspace analysis) algorithms in terms of error ϵ and eigengap gap . Here $gap_r = \frac{\lambda_{K+1}}{\lambda_K}$ for PSA and $gap_r = \max_{k=1,\dots,K} \frac{\lambda_{k+1}}{\lambda_k}$ for PCA algorithms. Also, $gap = \lambda_K - \lambda_{K+1}$ for PSA algorithms and $gap = \min_{k=1,\dots,K} \lambda_k - \lambda_{k+1}$ for PCA algorithms.

Distributed Stochastic PCA: Recently, some work have been proposed for distributed

Table 2.1: Comparison of Communication and Iteration Cost

	Comm./Iteration	No. of Iterations	Total Comm.	PCA/PSA
DistSeqPM	$\mathcal{O}(K \frac{1}{\log gap_r^{-1}} \log \frac{1}{\epsilon})$	$\mathcal{O}(K \frac{1}{\log gap_r^{-1}} \log \frac{1}{\epsilon})$	$\mathcal{O}(K^2 \frac{1}{\log^2 gap_r^{-1}} \log^2 \frac{1}{\epsilon})$	PCA
S-DOT	$\mathcal{O}(\frac{1}{\log gap_r^{-1}} \log \frac{1}{\epsilon})$	$\mathcal{O}(\frac{1}{\log gap_r^{-1}} \log \frac{1}{\epsilon})$	$\mathcal{O}(\frac{1}{\log^2 gap_r^{-1}} \log^2 \frac{1}{\epsilon})$	PSA
DeEPCA	$\mathcal{O}(\log \frac{1}{gap})$	$\mathcal{O}(\frac{1}{gap} \log \frac{1}{\epsilon})$	$\mathcal{O}(\frac{1}{gap} \log \frac{1}{gap} \log \frac{1}{\epsilon})$	PSA

PCA in the stochastic (online) settings. The goal in some of these works has been improving the communication efficiency of distributed methods for PCA. The work in [42] computes the top eigenvector of local covariances at each node and then add all these local eigenvectors in the last iteration to compute the final estimate of eigenvector. Hence, this way they only need a single round of communication at the last iteration of algorithm. Since, we add up all the local estimates in the last iteration of algorithm the approach taken in [42] is not very well suited for streaming settings. The work in [54] on the other hand proposes a distributed PCA algorithm based on the Krasulina’s method using a mini-batching approach. Finite sample complexity analysis of the algorithm showed speed up in the convergence proportional to the size of the mini-batch. The conditions on the size of mini-batch beyond which the speed up advantage dies out is also derived here. The algorithm, however, is for the estimation of the dominant eigenvector only and assumes that exact averaging of estimates at the nodes is possible which is possible in specific cases like federated learning.

Distributed PCA as an Optimization Problem: PCA is a non-convex problem since the uncorrelated constraint requires the solution to be a specific element on the Stiefel manifold. Recently, some algorithms in the field of distributed optimization were proposed to deal with non-convex problems. While some of those deal with unconstrained problems [55], some are developed for non-convex objectives with convex constraints [56, 57], while some methods guarantee convergence only to a stationary point [58]. For these reasons, none of the existing distributed algorithms for non-convex problems are directly applicable for the PCA objective. A recent work based on perturbation theory for linear operators based on the Picard iteration was proposed for distributed optimization in [59]. The extension of this work in [60] demonstrated the application of the distributed Picard iteration (DPI) method to distributed PCA but it could only prove local convergence, i.e., if the estimate is already “close enough” to the optimal solution, then it converges to the optimal point at a linear rate. Furthermore, the DPI method suffers from two more limitations in terms of its theoretical analysis, namely it requires the covariance matrix to be full rank as well as the upper bound on step size required for convergence guarantees is not quantified in terms of problem parameters like eigengap, data dimension etc. Thus, many gaps still remain to be filled in distributed PCA.

Chapter 3

Distributed Sanger's Algorithm: A Linearly Converging Algorithm for Distributed PCA

This chapter considers the problem of computing the dominant eigenvectors of a covariance matrix in distributed settings. The goal is to estimate the true eigenvectors, not just any subspace spanned by them, of the covariance matrix of the data that is distributed across an arbitrarily connected network. Also, the focus is on providing a solution that is efficient in terms of communications between the interconnected nodes of an arbitrary network. One of the classical methods for PCA include Oja's rule [12] and generalized Hebbian method [13]. This chapter talks about a distributed algorithm called *Distributed Sanger's Algorithm* that is based on the generalized Hebbian algorithm (GHA) proposed by Sanger [13], wherein the nodes perform local computations along with information exchange with their directly connected neighbors. Detailed theoretical analysis and numerical experiments are provided to demonstrate the effectiveness of the proposed solution.

3.1 Introduction

Massive and high-dimensional datasets are becoming an increasingly essential part of the modern world ranging from healthcare to finance and from social media to the Internet-of-Things (IoT). In a related trend, machine learning algorithms are finding their applications in every possible domain because of their data-driven nature and the ability to generalize to new unseen

data. But these algorithms need a considerable amount of data preprocessing for their effective and efficient use. One of the major steps in this preprocessing is dimension reduction and feature learning for compression and extraction of useful features from raw data that can be used in downstream machine learning algorithms for classification, clustering, etc. Simultaneously, the enormity of the amount of available data makes it difficult to manage it at a single location. There are multiple and an increasing number of scenarios where data is distributed across different locations, either due to storage constraints or by its inherent nature like in the Internet-of-Things [61]. This aspect of the modern-world data have led researchers to explore distributed algorithms, which can process information across different locations/machines [62]. These aforementioned issues have motivated us to study and develop algorithms for distributed PCA that are efficient in terms of computations and communications among multiple machines, and that can also be proven to converge at a fast rate.

3.1.1 Our Contributions

The main contributions of this paper are (1) a novel algorithm for distributed PCA, (2) theoretical guarantees for the proposed distributed algorithm with a linear convergence rate to a small neighborhood of the true PCA solution, and (3) experimental results to further demonstrate the efficacy of the proposed algorithm.

Our focus in this paper is to solve the distributed PCA problem so as to find a solution that not only enables dimensionality reduction, but that also provides uncorrelated features of data distributed over a network. That is, our goal is to estimate the true eigenvectors, not just any subspace spanned by them, of the covariance matrix of the data that is distributed across an arbitrarily connected network. Also, we focus on providing a solution that is efficient in terms of communications between the interconnected nodes of an arbitrary network. To that end, we propose a distributed algorithm that is based on the generalized Hebbian algorithm (GHA) proposed by Sanger [13], wherein the nodes perform local computations along with information exchange with their directly connected neighbors, similar to the idea followed in the distributed gradient descent (DGD) approach in [63]. The local computations do not involve the calculation of any gradient, but we instead use a “psuedo gradient”, which we henceforth call *Sanger’s direction*. In our proposed solution, termed the *Distributed Sanger’s Algorithm (DSA)*, we have also done away with the need of explicit consensus iterations for making the nodes agree with each other, thereby making it a one time-scale solution that is more communications efficient. Theoretical guarantees are also provided for our proposed distributed PCA algorithm when using a constant step size. The analysis shows that, when using a constant step size α , the

DSA solution reaches within a $\mathcal{O}(\alpha)$ -neighborhood of the optimal solution at a linear rate when the error metric is the angles between the estimated vectors and the true eigenvectors¹. We also provide experimental results and comparisons with centralized orthogonal iteration [19], centralized GHA [13], a sequential distributed power method-based approach and distributed projected gradient descent. The results support our claims and analysis.

To the best of our knowledge, this is the first solution for distributed PCA that uses a Hebbian update, achieves network agreement without the use of explicit consensus iterations, and still provably reaches the globally optimum solution (within an error margin) at all nodes at a linear rate.

3.2 Problem Formulation

Principal Component Analysis (PCA) aims at finding the basis of a low-dimensional space that can decorrelate the features of data points and also retain maximum information. More formally, for a random vector $\mathbf{y} \in \mathbb{R}^d$ with $\mathbb{E}[\mathbf{y}] = \mathbf{0}$, PCA involves finding the top- K eigenvectors of the covariance matrix $\Sigma := \mathbb{E}[\mathbf{y}\mathbf{y}^T]$. The zero mean assumption is taken here without loss of generality as the mean can be subtracted in case data is not centered. Mathematically, PCA can be formulated as

$$\mathbf{X} = \arg \min_{\mathbf{X} \in \mathbb{R}^{d \times K}} \mathbb{E} \left[\|\mathbf{y} - \mathbf{X}\mathbf{X}^T\mathbf{y}\|_2^2 \right] \quad \text{such that} \quad \forall l \neq q, \left(\mathbb{E} \left[\mathbf{X}^T \mathbf{y} \mathbf{y}^T \mathbf{X} \right] \right)_{lq} = 0. \quad (3.1)$$

The constraint $\left(\mathbb{E} \left[\mathbf{X}^T \mathbf{y} \mathbf{y}^T \mathbf{X} \right] \right)_{lq} = 0, \forall l \neq q$, ensures that \mathbf{X} decorrelates the features of \mathbf{y} . Now, $\mathbb{E} \left[\mathbf{X}^T \mathbf{y} \mathbf{y}^T \mathbf{X} \right] = \mathbf{X}^T \mathbb{E} \left[\mathbf{y} \mathbf{y}^T \right] \mathbf{X} = \mathbf{X}^T \Sigma \mathbf{X}$ and it is straightforward to see that this quantity is diagonal only if \mathbf{X} contains the eigenvectors of Σ . This explains why the search for a solution of PCA ends with the eigenvectors and not the subspace spanned by them. In practice, we do not have access to Σ and so a covariance matrix estimated from the samples of \mathbf{y} is used instead. Specifically, for a dataset with N samples $\{\mathbf{y}_l\}_{l=1}^N$, or equivalently, for a data matrix $\mathbf{Y} := [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N]$, the sample covariance matrix can be written as $\mathbf{C} = \frac{1}{N} \mathbf{Y} \mathbf{Y}^T$ such that $\Sigma := \mathbb{E}[\mathbf{C}]$. The true solution for PCA is then obtained by finding the eigenvectors of the covariance matrix \mathbf{C} , which are also the left singular vectors of the data matrix \mathbf{Y} . The empirical form of (5.1) is thus

$$\mathbf{X} = \arg \min_{\mathbf{X} \in \mathbb{R}^{d \times K}} f(\mathbf{X}) = \arg \min_{\mathbf{X} \in \mathbb{R}^{d \times K}} \|\mathbf{Y} - \mathbf{X}\mathbf{X}^T\mathbf{Y}\|_F^2 \quad \text{such that} \quad \forall l \neq q, \left(\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X} \right)_{lq} = 0. \quad (3.2)$$

¹Our results can also be extrapolated to guarantee exact convergence with decaying step size, albeit at a slower than linear rate.

In the literature, however, PCA is usually posed with a ‘relaxed’ orthogonality constraint of $\mathbf{X}^T \mathbf{X} = \mathbf{I}$ instead of $\left(\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}\right)_{lq} = 0, \forall l \neq q$, as follows:

$$\mathbf{X} = \arg \min_{\mathbf{X} \in \mathbb{R}^{d \times K}, \mathbf{X}^T \mathbf{X} = \mathbf{I}} f(\mathbf{X}) = \arg \min_{\mathbf{X} \in \mathbb{R}^{d \times K}, \mathbf{X}^T \mathbf{X} = \mathbf{I}} \|\mathbf{Y} - \mathbf{X} \mathbf{X}^T \mathbf{Y}\|_F^2. \quad (3.3)$$

The optimization formulation in (3.3) with this constraint will only lead to a subspace spanned by the eigenvectors of \mathbf{C} as the solution, thus actually making it a Principal Subspace Analysis (PSA) formulation. In other words, although the formulation (3.3) gives a solution on the Stiefel manifold, the actual PCA formulation (3.2) requires the solution to be within a very specific subset of that manifold that corresponds to the eigenvectors of \mathbf{C} . The accuracy of the solutions given by the PCA and PSA formulations will be the same when measured in terms of the principal angles between the subspace estimates and the true subspace spanned by the eigenvectors of the covariance matrix. Specifically, if $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_K]$ is an estimate of the basis of the space spanned by the eigenvectors $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_K]$, then the principal angles between \mathbf{Q} and \mathbf{X} given by either (3.2) or (3.3) will be the same. But a more suitable measure of accuracy for any PCA solution should be the angles between \mathbf{x}_i and \mathbf{q}_i for all $i = 1, \dots, K$, which motivates us to judge the efficacy of any solution with respect to this metric instead of the principal subspace angles.

In the distributed setup considered in this paper, we consider a network of M nodes such that the undirected graph, $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, describing the network is connected. Here $\mathcal{V} = \{1, 2, \dots, M\}$ is the set of nodes and \mathcal{E} is the set of edges, i.e., $(i, j) \in \mathcal{E}$ if there is a direct path between i and j . The set of neighbors for any node i is denoted by \mathcal{N}_i . Under the setup of samples being distributed over the M nodes, let us assume that the i^{th} node has a data matrix \mathbf{Y}_i containing N_i samples such that $N = \sum_{i=1}^M N_i$. Thus each node has access to only a local covariance matrix $\mathbf{C}_i = \frac{1}{N_i} \mathbf{Y}_i \mathbf{Y}_i^T$ instead of the global covariance matrix but one can see that $N\mathbf{C} = \sum_{i=1}^M N_i \mathbf{C}_i$. In this setting, a straightforward approach might be that each node finds its own solution independent of the data at all the other nodes. While this might seem viable, this approach will have major drawbacks. Recall that the sample covariance \mathbf{C} approximates the population covariance $\mathbf{\Sigma}$ at a rate of $\mathcal{O}(f(N^{-1}))$, where f is some function (depending on the distribution) of the number of samples N [?]. Since the local data has smaller number of samples than the global data, working with the local covariance matrix \mathbf{C}_i alone instead of somehow using the whole data will lead to a larger error in estimation of the eigenvectors. Also, since uniform sampling from the underlying data distribution is not guaranteed in distributed setups, the samples at a node may end up being from a narrow part of the entire distribution, thus being more biased away from the true distribution. This invites the need for the nodes to

collaborate amongst themselves in a way that all the data is utilized to find estimates of the eigenvectors at each node while ensuring that all the nodes agree with each other. Thus, for a distributed setting, the PCA problem in (5.1) can be rewritten here as

$$\begin{aligned} \mathbf{X} = \arg \min_{\mathbf{X} \in \mathbb{R}^{d \times K}} \sum_{i=1}^M f_i(\mathbf{X}) &= \arg \min_{\mathbf{X} \in \mathbb{R}^{d \times K}} \sum_{i=1}^M \|\mathbf{Y}_i - \mathbf{X}\mathbf{X}^T\mathbf{Y}_i\|_F^2 \\ \text{such that } \forall l \neq q, &\left(\mathbf{X}^T \left(\sum_{i=1}^M \mathbf{Y}_i \mathbf{Y}_i^T\right) \mathbf{X}\right)_{lq} = 0. \end{aligned} \quad (3.4)$$

It is easy to see that $\sum_{i=1}^M f_i(\mathbf{X}) = f(\mathbf{X})$. Also, in a distributed setup, each node i maintains its own copy \mathbf{X}_i of the variable \mathbf{X} due to the difference in local information (local data) they carry. Thus, all nodes need to agree with each other to ensure the entire network reaches the same true solution. Hence, the true distributed PCA objective at the i^{th} node is written as

$$\arg \min_{\mathbf{X}_i \in \mathbb{R}^{d \times K}} \sum_{i=1}^M \|\mathbf{Y}_i - \mathbf{X}_i \mathbf{X}_i^T \mathbf{Y}_i\|_F^2 \quad \text{such that} \quad (3.5)$$

$$\forall j \in \mathcal{N}_i, \mathbf{X}_i = \mathbf{X}_j \quad \text{and} \quad \forall l \neq q, \left(\mathbf{X}_i^T \left(\sum_{i=1}^M \mathbf{Y}_i \mathbf{Y}_i^T\right) \mathbf{X}_i\right)_{lq} = 0. \quad (3.6)$$

Note that (5.1)–(3.6) are non-convex optimization problems due to the non-convexity of the constraint set. One possible solution to the PCA problem is to instead solve a convex relaxation of the original non-convex function [36, 64]. The issue with these solutions is that they require $O(d^2)$ memory and computation, which can be prohibitive in high-dimensional settings. In addition, due to $O(d^2)$ iterate size these solutions are not ideal for distributed settings. Also, these formulations, without any further constraints, will not necessarily give a basis that is the set of dominant eigenvectors. Instead, they might end up giving a rotated basis as explained earlier, thereby not completing the task of decorrelating features. Hence, in this paper we use an algebraic method based on GHA for neural network training, which has $O(dK)$ memory and computation requirements, to solve the distributed PCA problem. Our goal is to converge to the true eigenvectors of the global covariance matrix \mathbf{C} at every node of the network. As noted earlier, distributed variants of the power method exist in the literature [49–51] that can find the dominant eigenvector but these methods employ two time-scale approaches that involve several consensus averaging rounds for each iteration of the power method. Such two time-scale approaches can be expensive in terms of communications cost. In this chapter, we propose a one time-scale method that finds the top K eigenvectors of the global sample covariance matrix \mathbf{C} at each node through local computations and information exchange with neighbors. The proposed method also converges linearly up to a neighborhood of the true solution when the error metric considered is the angle between the estimates and the true eigenvectors.

3.3 The Proposed Algorithm

In [13], Sanger proposed a generalized Hebbian algorithm (GHA) to train a neural network and find the eigenvectors of the input autocorrelation matrix (same as the covariance matrix for zero-mean input). The outputs of such a network, when the weights are given by the eigenvectors, are the uncorrelated features of the input data that allow data reconstruction with minimal error, hence serving the true purpose of PCA. The algorithm was originally developed to tackle the centralized PCA problem in the case of streaming data, where a new data sample $\mathbf{y}_t, t = 1, 2, \dots$, arrives at each time instance.

In this paper we consider a batch setting, but the alignment of GHA with our basic goal of finding the eigenvectors motivates us to leverage it for our distributed setup. The rationale behind the idea of extrapolating the streaming case to a distributed batch setting is simple: since $\mathbb{E}[\mathbf{y}_t \mathbf{y}_t^T] = \mathbb{E}[\mathbf{Y}_i \mathbf{Y}_i^T] = \mathbf{\Sigma}$, the sample-wise distributed data setting can be seen as a *mini-batch* variant of the streaming data setting. In the context of neural network training, our approach can be viewed as training a network at each node with a mini-batch of samples in a way that all nodes end up with the same trained network whose weights are given by the eigenvectors of the autocorrelation matrix of the entire batch of samples.

The iterate for the GHA as given in [13] has the following update for the matrix of eigenvectors (i.e., the neural network weight matrix) \mathbf{X} when the t^{th} sample \mathbf{y}_t arrives at the input of the neural network:

$$\mathbf{X}^{(t+1)} = \mathbf{X}^{(t)} + \alpha_t \left[\mathbf{y}_t \mathbf{y}_t^T \mathbf{X}^{(t)} - \mathbf{X}^{(t)} \mathbf{U} \left((\mathbf{X}^{(t)})^T \mathbf{y}_t \mathbf{y}_t^T \mathbf{X}^{(t)} \right) \right], \quad (3.7)$$

where $\mathbf{U} : \mathbb{R}^{K \times K} \rightarrow \mathbb{R}^{K \times K}$ is an operator that sets all the elements below the diagonal to zero and α_t is the step size. For $K = 1$, and defining $\mathbf{\Sigma}_t = \mathbf{y}_t \mathbf{y}_t^T$, it was shown in [12] that the term $(\mathbf{X}^{(t)})^T \mathbf{y}_t \mathbf{y}_t^T \mathbf{X}^{(t)} = (\mathbf{X}^{(t)})^T \mathbf{\Sigma}_t \mathbf{X}^{(t)}$ is the consequence of a power series approximation of Oja's rule in lieu of the explicit normalization used in the case of the power method. In the case of $K > 1$, $\mathbf{U} \left((\mathbf{X}^{(t)})^T \mathbf{y}_t \mathbf{y}_t^T \mathbf{X}^{(t)} \right) = \mathbf{U} \left((\mathbf{X}^{(t)})^T \mathbf{\Sigma}_t \mathbf{X}^{(t)} \right)$ helps combine Oja's algorithm with

the Gram–Schmidt orthogonalization step as follows:

$$\begin{aligned}
\mathbf{X}^{(t)} \mathbf{U} \left((\mathbf{X}^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{X}^{(t)} \right) &= \mathbf{X}^{(t)} \mathbf{U} \left(\begin{bmatrix} (\mathbf{x}_1^{(t)})^T \\ \vdots \\ (\mathbf{x}_K^{(t)})^T \end{bmatrix} \boldsymbol{\Sigma}_t \begin{bmatrix} \mathbf{x}_1^{(t)} & \cdots & \mathbf{x}_K^{(t)} \end{bmatrix} \right) \\
&= \mathbf{X}^{(t)} \mathbf{U} \left(\begin{bmatrix} (\mathbf{x}_1^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_1^{(t)} & (\mathbf{x}_1^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_2^{(t)} & \cdots & (\mathbf{x}_1^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_K^{(t)} \\ (\mathbf{x}_2^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_1^{(t)} & (\mathbf{x}_2^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_2^{(t)} & \cdots & (\mathbf{x}_2^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_K^{(t)} \\ \vdots & \vdots & \ddots & \vdots \\ (\mathbf{x}_K^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_1^{(t)} & (\mathbf{x}_K^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_2^{(t)} & \cdots & (\mathbf{x}_K^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_K^{(t)} \end{bmatrix} \right) \\
&= \mathbf{X}^{(t)} \left(\begin{bmatrix} (\mathbf{x}_1^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_1^{(t)} & (\mathbf{x}_1^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_2^{(t)} & \cdots & (\mathbf{x}_1^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_K^{(t)} \\ 0 & (\mathbf{x}_2^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_2^{(t)} & \cdots & (\mathbf{x}_2^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_K^{(t)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (\mathbf{x}_K^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_K^{(t)} \end{bmatrix} \right) \\
&= \begin{bmatrix} (\mathbf{x}_1^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_1^{(t)} & \sum_{p=1}^2 (\mathbf{x}_p^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_2^{(t)} \mathbf{x}_p^{(t)} & \cdots & \sum_{p=1}^K (\mathbf{x}_p^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_K^{(t)} \mathbf{x}_p^{(t)} \end{bmatrix}.
\end{aligned} \tag{3.8}$$

Thus, for any $k = 1, \dots, K$, the term involving $\mathbf{U}(\cdot)$ in (3.7) includes an implicit normalization term $(\mathbf{x}_k^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_k^{(t)}$ as well an orthogonalization term $\sum_{p=1}^{k-1} (\mathbf{x}_p^{(t)})^T \boldsymbol{\Sigma}_t \mathbf{x}_k^{(t)} \mathbf{x}_p^{(t)}$, which—analagous to the Gram–Schmidt orthogonalization procedure—forces the estimate $\mathbf{x}_k^{(t)}$ to be orthogonal to all the estimates $\mathbf{x}_p^{(t)}, p = 1, \dots, k-1$. Another important thing to note about the GHA algorithm is that, in order to estimate the dominant K eigenvectors, it only requires the corresponding top K eigenvalues to be distinct (and nonzero). In other words, it does not require the covariance matrix to be non-singular.

In the deterministic setting, where we have the full-batch instead of new samples every instance, this iterate changes to

$$\mathbf{X}^{(t+1)} = \mathbf{X}^{(t)} + \alpha_t \left[\mathbf{C} \mathbf{X}^{(t)} - \mathbf{X}^{(t)} \mathbf{U} \left((\mathbf{X}^{(t)})^T \mathbf{C} \mathbf{X}^{(t)} \right) \right] = \mathbf{X}^{(t)} + \alpha_t \mathcal{H}(\mathbf{C}, \mathbf{X}^{(t)}). \tag{3.9}$$

Here, we term $\mathcal{H} : \mathbb{R}^{d \times d} \times \mathbb{R}^{d \times K} \rightarrow \mathbb{R}^{d \times K}$, $\mathcal{H}(\mathbf{C}, \mathbf{X}^t) := \left(\mathbf{C} \mathbf{X}^t - \mathbf{X}^t \mathbf{U} \left((\mathbf{X}^t)^T \mathbf{C} \mathbf{X}^t \right) \right)$ as the Sanger direction. An iterate similar to (3.9) has been proven to have global convergence in [24] for some very specific choice of the step sizes that are dependent on the iterate itself. Its straightforward extension to the distributed case is not possible as that would lead to different step sizes at different nodes of the network, making it difficult to talk about its convergence guarantees. Hence, to adapt this iterative method to our distributed setup, we use the typical combine and update strategy used quite richly in the literature for distributed algorithms such as [63, 65–67]. The main contributions of such works lie in showing that the resulting distributed

Algorithm 1: Distributed Sanger's Algorithm (DSA)

Input: $\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_M, [w_{ij}], \alpha, K$
Initialize: $\forall i, \mathbf{X}_i^{(0)} \leftarrow \mathbf{X}_{\text{init}} : \mathbf{X}_{\text{init}} \in \mathbb{R}^{d \times K}, \mathbf{X}_{\text{init}}^T \mathbf{X}_{\text{init}} = \mathbf{I}$

 1: **for** $t = 1, 2, \dots$ **do**

 2: Communicate $\mathbf{X}_i^{(t-1)}$ from each node i to its neighbors

 3: Estimate of eigenvectors at node i : $\mathbf{X}_i^{(t)} \leftarrow \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{X}_j^{(t-1)} + \alpha \mathcal{H}_i(\mathbf{X}_i^{(t-1)})$

 4: **end for**
Return: $\mathbf{X}_i^{(t)}, i = 1, 2, \dots, M$

algorithms achieve consensus (i.e., all nodes will have the same iterate values eventually) and, in addition, the consensus value is the same as the centralized solution. The convergence guarantees for these methods are mainly restricted to convex and strongly convex problems though. Our distributed version of (3.9) for PCA, which is non-convex, is based on similar principles of combine and update.

Specifically, the node i at iteration t carries a local copy $\mathbf{X}_i^{(t)}$ of the estimate of the eigenvectors of the global covariance matrix \mathbf{C} . In the combine step, each node i exchanges the iterate values with its immediate neighbors $j \in \mathcal{N}_i$, where \mathcal{N}_i denotes the neighborhood of node i , and then takes a weighted sum of the iterates received along with its local iterate. Then this sum is updated independently at all nodes using their respective local information. Since node i in the network only has access to its local sample covariance \mathbf{C}_i , the update is in the form of a local Sanger's direction given as

$$\mathcal{H}_i(\mathbf{C}_i, \mathbf{X}_i^{(t)}) = \mathbf{C}_i \mathbf{X}_i^{(t)} - \mathbf{X}_i^{(t)} \mathcal{U}\left((\mathbf{X}_i^{(t)})^T \mathbf{C}_i \mathbf{X}_i^{(t)}\right). \quad (3.10)$$

The details of the proposed distributed PCA algorithm, called the Distributed Sanger's Algorithm (DSA), are given in Algorithm 1. The weight matrix $\mathbf{W} = [w_{ij}]$ in this algorithm is a doubly stochastic matrix conforming to the network topology [52] in the sense that for $i \neq j$, $w_{ij} \neq 0$ when $(i, j) \in \mathcal{E}$ and $w_{ij} = 0$ otherwise. Also, $\forall i, w_{ii} \neq 0$, i.e., there is a self loop at each node. Note that connectivity of the network, as discussed in Section 3.2, is a necessary condition for convergence of DSA. The connectivity assumption, in turn, ensures the Markov chain underlying the graph \mathcal{G} is aperiodic and irreducible, which implies that the second-largest (in magnitude) eigenvalue of \mathbf{W} , $\beta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_M(\mathbf{W})|\}$, is strictly less than 1. While DSA shares algorithmic similarities with first-order distributed optimization methods [63, 68] in which the combine-and-update strategy is used, our challenge is characterizing its convergence behavior due to the non-convex and constrained nature of the distributed PCA problem. To this end, we first provide a general result in Section 3.4.1 where we prove the convergence of a modified form of GHA. Then we utilize that result, along with some linear algebraic tools and

additional lemmas provided in the appendices, to characterize the dynamics of the distributed setup in Section 3.4.2 and prove the convergence of the proposed algorithm.

3.4 Convergence Analysis

The convergence analysis of DSA algorithm is provided in this section. We first provide a general result where we prove the convergence of a modified form of GHA. Then we utilize that result, along with some linear algebraic tools and additional lemmas, to characterize the dynamics of the distributed setup and prove the convergence of the proposed algorithm.

3.4.1 Convergence Analysis of a Modified GHA

Let $\mathbf{X}^{(t)} = [\mathbf{x}_1^{(t)} \quad \mathbf{x}_2^{(t)} \quad \dots \quad \mathbf{x}_K^{(t)}] \in \mathbb{R}^{d \times K}$, $K \leq d$, be an estimate of the K -dimensional subspace spanned by the eigenvectors of the covariance matrix \mathbf{C} after t iterations and $\mathbf{q}_l, l = 1, \dots, d$, be the eigenvectors of \mathbf{C} with corresponding eigenvalues λ_l . On expanding (3.9) using (3.8), it is clear that the GHA update equation for estimation of the k^{th} eigenvector using a constant step size α is as follows:

$$\mathbf{x}_k^{(t+1)} = \mathbf{x}_k^{(t)} + \alpha (\mathbf{C}\mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \mathbf{x}_p^{(t)} (\mathbf{x}_p^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}). \quad (3.11)$$

We now slightly modify (4.1) by replacing $\mathbf{x}_p^{(t)}$ for $p < k$ by the true eigenvectors \mathbf{q}_p . We term the resulting update equation *modified GHA* and note that this is not an algorithm in the true sense of the term as it cannot be implemented because of its dependence on the true eigenvectors \mathbf{q}_p . The sole purpose of this modified GHA is to help in our ultimate goal of providing convergence guarantee for the DSA algorithm. The update equation of the modified GHA for “estimation” of the k^{th} eigenvector of \mathbf{C} , $k = 1, \dots, K$, has the form

$$\mathbf{x}_k^{(t+1)} = \mathbf{x}_k^{(t)} + \alpha (\mathbf{C}\mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C}\mathbf{x}_k^{(t)}). \quad (3.12)$$

Note that similar to the original GHA, this modified GHA assumes that \mathbf{C} has K distinct eigenvalues, i.e., $\lambda_1 > \lambda_2 > \dots > \lambda_K > \lambda_{K+1} \geq \dots \geq \lambda_d \geq 0$. Now, since $\mathbf{q}_l, l = 1, \dots, d$, are the eigenvectors of a real symmetric matrix, they form a basis for \mathbb{R}^d and can be used for expansion of any $\mathbf{x}_k^{(t)}$ as

$$\mathbf{x}_k^{(t)} = \sum_{l=1}^d z_{k,l}^{(t)} \mathbf{q}_l, \quad (3.13)$$

where $z_{k,l}^{(t)}$ is the coefficient corresponding to the eigenvector \mathbf{q}_l in the expansion of $\mathbf{x}_k^{(t)}$. Multiplying both sides of (3.12) by \mathbf{q}_l^T and using the fact that $\mathbf{q}_l^T \mathbf{q}_{l'} = 0$ for $l \neq l'$, we get

$$z_{k,l}^{(t+1)} = z_{k,l}^{(t)} + \alpha(\mathbf{q}_l^T \mathbf{C} \mathbf{x}_k^{(t)} - \mathbf{q}_l^T (\sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \mathbf{x}_k^{(t)}) - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} z_{k,l}^{(t)}).$$

This gives

$$z_{k,l}^{(t+1)} = z_{k,l}^{(t)} - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} z_{k,l}^{(t)}, \quad \text{for } l = 1, \dots, k-1, \quad (3.14)$$

$$\text{and } z_{k,l}^{(t+1)} = z_{k,l}^{(t)} + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}) z_{k,l}^{(t)}, \quad \text{for } l = k, \dots, d. \quad (3.15)$$

It has been shown in [23] that the update equation given by

$$\mathbf{x}_1^{(t+1)} = \mathbf{x}_1^{(t)} + \alpha(\mathbf{C} \mathbf{x}_1^{(t)} - (\mathbf{x}_1^{(t)})^T \mathbf{C} \mathbf{x}_1^{(t)} \mathbf{x}_1^{(t)})$$

for $k = 1$ converges to $\pm \mathbf{q}_1$ at a linear rate for a certain condition on the step size α . Specifically, it was proven that $(z_{1,1}^{(t)})^2 \rightarrow 1$ and $\sum_{l=2}^d (z_{1,l}^{(t)})^2 \leq b_1 \rho_1^t$, where $b_1 > 0$ is some constant and $\rho_1 = \left(\frac{1+\alpha\lambda_2}{1+\alpha\lambda_1}\right)^2 < 1$. Here, we extend the proof to a general k and show that the update equation given in the form of (3.12) for any $k = 1, \dots, K$, $K < d$, converges to the k^{th} dominant eigenvector.

Theorem 1. Suppose $\alpha \leq \frac{1}{3\lambda_1(2K-1)}$, where λ_1 is the largest eigenvalue of \mathbf{C} and K is the number of eigenvectors to be estimated, $\mathbf{q}_k^T \mathbf{x}_k^{(0)} \neq 0$, and $\|\mathbf{x}_k^{(0)}\| = 1$ for all k . Then the modified GHA iterate for $\mathbf{x}_k^{(t)}$ given by (3.12) converges at a linear rate to the eigenvector $\pm \mathbf{q}_k$ corresponding to the k^{th} largest eigenvalue λ_k of the covariance matrix \mathbf{C} .

Proof. The convergence of $\mathbf{x}_k^{(t)}$ to \mathbf{q}_k requires convergence of the lower-order coefficients $z_{k,1}^{(t)}, \dots, z_{k,k-1}^{(t)}$ and the higher-order coefficients $z_{k,k+1}^{(t)}, \dots, z_{k,d}^{(t)}$ to 0 and convergence of $z_{k,k}^{(t)}$ to ± 1 . Now,

$$\begin{aligned} |\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}| &= |\lambda_k - \sum_{l=1}^d \lambda_l (z_{k,l}^{(t)})^2| = |\lambda_k - \lambda_k (z_{k,k}^{(t)})^2 - \sum_{l=1}^{k-1} \lambda_l (z_{k,l}^{(t)})^2 - \sum_{l=k+1}^d \lambda_l (z_{k,l}^{(t)})^2| \\ &\geq |\lambda_k - \lambda_k (z_{k,k}^{(t)})^2| - \left| \sum_{l=1}^{k-1} \lambda_l (z_{k,l}^{(t)})^2 \right| - \left| \sum_{l=k+1}^d \lambda_l (z_{k,l}^{(t)})^2 \right| \\ \text{or, } \lambda_k |1 - (z_{k,k}^{(t)})^2| &\leq \left| \sum_{l=1}^{k-1} \lambda_l (z_{k,l}^{(t)})^2 \right| + \left| \sum_{l=k+1}^d \lambda_l (z_{k,l}^{(t)})^2 \right| + |\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}|. \end{aligned} \quad (3.16)$$

Thus, convergence of the lower-order and the higher-order coefficients to 0 along with convergence of the term $|\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}|$ will also imply the convergence of $z_{k,k}^{(t)}$ to ± 1 . To this end, Lemma 3 in the appendix proves linear convergence of the lower-order coefficients $z_{k,1}^{(t)}, \dots, z_{k,k-1}^{(t)}$ to 0 by showing $\sum_{l=1}^{k-1} (z_{k,l}^{(t+1)})^2 < a_1 \gamma^{t+1}$ for some constants $a_1 > 0, \gamma < 1$. Furthermore, Lemma 4 in the appendix shows that $\sum_{l=k+1}^d (z_{k,l}^{(t+1)})^2 \leq a_2 \rho_k^{t+1}$, where $a_1, a_2 > 0$

and $\gamma, \rho_k < 1$, thereby proving linear convergence of the higher-order coefficients to 0. Finally, Lemma 5 in the appendix shows that $|\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}| \leq ta_4(\delta^{t+1} + \max\{\delta^t, \gamma_1^t\})$, where $a_4 > 0$ and $\delta, \gamma_1 < 1$. The formal statements and proofs of Lemma 3, Lemma 4 and Lemma 5 are given in subsection 3.5.1.

Thus,

$$\begin{aligned}
\lambda_k |1 - (z_{k,k}^{(t)})^2| &\leq \left| \sum_{l=1}^{k-1} \lambda_l (z_{k,l}^{(t)})^2 \right| + \left| \sum_{l=k+1}^d \lambda_l (z_{k,l}^{(t)})^2 \right| + ta_4(\delta^{t+1} + \max\{\delta^t, \gamma_1^t\}) \\
&= \sum_{l=1}^{k-1} \lambda_l (z_{k,l}^{(t)})^2 + \sum_{l=k+1}^d \lambda_l (z_{k,l}^{(t)})^2 + ta_4(\delta^{t+1} + \max\{\delta^t, \gamma_1^t\}) \\
&< \lambda_1 \left(\sum_{l=1}^{k-1} (z_{k,l}^{(t)})^2 \right) + \sum_{l=k+1}^d (z_{k,l}^{(t)})^2 + ta_4(\delta^{t+1} + \max\{\delta^t, \gamma_1^t\}) \\
&< \lambda_1 (a_1 \gamma^t + a_2 \rho_k^t) + ta_4(\delta^{t+1} + \max\{\delta^t, \gamma_1^t\}).
\end{aligned}$$

Clearly, $\lim_{t \rightarrow \infty} |1 - (z_{k,k}^{(t)})^2| = 0$. Therefore, Theorem 1 shows that the iterates $\mathbf{x}_k^{(t)}$ of the form (3.12) converge linearly to eigenvectors \mathbf{q}_k of the covariance matrix \mathbf{C} . \square

3.4.2 Convergence Analysis of Distributed Sanger's Algorithm (DSA)

With the analysis of the modified GHA in hand, let us proceed to analyze the proposed DSA algorithm. The iterate of DSA at node i for the dominant K -dimensional eigenspace estimate ($K \leq d$) is given as

$$\mathbf{X}_i^{(t+1)} = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{X}_j^{(t)} + \alpha \mathcal{H}_i(\mathbf{C}_i, \mathbf{X}_i^{(t)}) = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{X}_j^{(t)} + \alpha \left(\mathbf{C}_i \mathbf{X}_i^{(t)} - \mathbf{X}_i^{(t)} \mathbf{U}((\mathbf{X}_i^{(t)})^T \mathbf{C}_i \mathbf{X}_i^{(t)}) \right), \quad (3.17)$$

where $\mathbf{X}_i^{(t)} = [\mathbf{x}_{i,1}^{(t)} \quad \mathbf{x}_{i,2}^{(t)} \quad \dots \quad \mathbf{x}_{i,K}^{(t)}] \in \mathbb{R}^{d \times K}$ is an estimate of the K -dimensional subspace of the global covariance matrix \mathbf{C} at the i^{th} node after t iterations, $\mathcal{H}_i(\mathbf{C}_i, \mathbf{X}_i^{(t)})$ is local Sanger's direction, and $w_{ij} \geq 0$ is a weight that node i assigns to $\mathbf{X}_j^{(t)}$ based on the connectivity between nodes i and j as mentioned before. The Sanger's direction and the update equation for an estimate of the k^{th} eigenvector is thus given as

$$\mathcal{H}_i(\mathbf{C}_i, \mathbf{x}_{i,k}^{(t)}) = \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)} - \sum_{p=1}^{k-1} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,p}^{(t)} \quad (3.18)$$

$$\text{and, } \mathbf{x}_{i,k}^{(t+1)} = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{x}_{j,k}^{(t)} + \alpha \left(\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)} - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \right). \quad (3.19)$$

Now, let the average of $\mathbf{x}_{1,k}^{(t)}, \mathbf{x}_{2,k}^{(t)}, \dots, \mathbf{x}_{M,k}^{(t)}$ after t^{th} iteration be denoted as $\bar{\mathbf{x}}_k^{(t)} = \frac{1}{M} \sum_{i=1}^M \mathbf{x}_{i,k}^{(t)}$ and given by taking average of (3.19) over all the nodes $i = 1, \dots, M$ as

$$\begin{aligned} \bar{\mathbf{x}}_k^{(t+1)} &= \bar{\mathbf{x}}_k^{(t)} + \frac{\alpha}{M} \sum_{i=1}^M \mathcal{H}_i(\mathbf{x}_{i,k}^{(t)}) \\ &= \bar{\mathbf{x}}_k^{(t)} + \frac{\alpha}{M} \sum_{i=1}^M \mathcal{H}_i(\bar{\mathbf{x}}_k^{(t)}) + \frac{\alpha}{M} \sum_{i=1}^M \mathcal{H}_i(\mathbf{x}_{i,k}^{(t)}) - \frac{\alpha}{M} \sum_{i=1}^M \mathcal{H}_i(\bar{\mathbf{x}}_k^{(t)}) \\ &= \bar{\mathbf{x}}_k^{(t)} + \frac{\alpha}{M} \sum_{i=1}^M \mathcal{H}_i(\bar{\mathbf{x}}_k^{(t)}) + \alpha \mathbf{h}_k^{(t)} \\ &= \bar{\mathbf{x}}_k^{(t)} + \frac{\alpha}{M} (\mathbf{C} \bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)} - \sum_{i=1}^M \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)}) + \alpha \mathbf{h}_k^{(t)}, \end{aligned}$$

where $\mathbf{h}_k^{(t)} = \frac{1}{M} \sum_{i=1}^M (\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)}) - \mathcal{H}_i(\bar{\mathbf{x}}_k^{(t)}))$. We present analysis of the DSA algorithm by first proving convergence of the average $\bar{\mathbf{x}}_k^{(t)}$ to a neighborhood of the eigenvector \mathbf{q}_k of the global covariance matrix \mathbf{C} while using a constant step size. Then with the help of Lemma 8, which proves that the deviation of the iterates $\mathbf{x}_{i,k}^{(t)}$ at each node from the average $\bar{\mathbf{x}}_k^{(t)}$ is upper bounded, we prove that the iterates at each node also converge to a neighborhood of the true solution. It is noteworthy that the analysis of DSA does not require additional constraints on eigenvalues of \mathbf{C}_i , i.e., similar to GHA, we only require the top K eigenvalues of \mathbf{C} to be distinct and non-zero.

The complete proof of convergence of DSA is done by induction. First, we show the convergence of $\mathbf{x}_{i,1}^{(t)}$ to a $\mathcal{O}(\alpha)$ neighborhood of \mathbf{q}_1 and then analyze the rest of the eigenvector estimates $\mathbf{x}_{i,k}^{(t)}, k = 2, \dots, K$, by assuming that the higher-order estimates have converged.

Case I for Induction – $k = 1$: The iterate for the dominant eigenvector is

$$\mathbf{x}_{i,1}^{(t+1)} = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{x}_{j,1}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,1}^{(t)} - ((\mathbf{x}_{i,1}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,1}^{(t)}) \mathbf{x}_{i,1}^{(t)}). \quad (3.20)$$

Theorem 2. Suppose $\alpha \leq \frac{\min_i w_{ii}}{3\lambda_1(2K-1)}$, where λ_1 is the largest eigenvalue of \mathbf{C} and K is the number of eigenvectors to be estimated, $\mathbf{q}_1^T \mathbf{x}_{i,1}^{(0)} \neq 0$, and $\|\mathbf{x}_{i,1}^{(0)}\| = 1$. Then the DSA iterate for $\mathbf{x}_{i,1}^{(t)}$ given by (3.20) converges at a linear rate to an $\mathcal{O}(\alpha)$ neighborhood of the eigenvector $\pm \mathbf{q}_1$ corresponding to the largest eigenvalue λ_1 of the global covariance matrix \mathbf{C} at every node of the network.

Proof. We know that

$$\|\mathbf{x}_{i,1}^{(t)} - \mathbf{x}_1^*\| \leq \|\mathbf{x}_{i,1}^{(t)} - \bar{\mathbf{x}}_1^{(t)}\| + \|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\|, \quad \text{where } \mathbf{x}_1^* = \pm \mathbf{q}_1. \quad (3.21)$$

The term $\|\mathbf{x}_{i,1}^{(t)} - \bar{\mathbf{x}}_1^{(t)}\|$ is a measure of consensus in the network and we prove in Lemma 8 that

this difference decreases linearly until it reaches a level of $\mathcal{O}(\alpha)$. More precisely,

$$\|\mathbf{x}_{i,1}^{(t)} - \bar{\mathbf{x}}_1^{(t)}\| \leq b_1 \left(\beta^t + \frac{\alpha}{1-\beta} \right), \quad (3.22)$$

where $\beta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_M(\mathbf{W})|\}$. In particular, it is well known that for a connected graph $\beta < 1$. Now, the average iterate of DSA for the estimate of the dominant eigenvector ($k = 1$) is

$$\bar{\mathbf{x}}_1^{(t)} = \bar{\mathbf{x}}_1^{(t-1)} + \frac{\alpha}{M} (\mathbf{C}\bar{\mathbf{x}}_1^{(t-1)} - (\bar{\mathbf{x}}_1^{(t-1)})^T \mathbf{C}\bar{\mathbf{x}}_1^{(t-1)} \bar{\mathbf{x}}_1^{(t-1)}) + \alpha \mathbf{h}_1^{(t-1)}.$$

Thus,

$$\begin{aligned} \bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^* &= \bar{\mathbf{x}}_1^{(t-1)} + \frac{\alpha}{M} (\mathbf{C}\bar{\mathbf{x}}_1^{(t-1)} - (\bar{\mathbf{x}}_1^{(t-1)})^T \mathbf{C}\bar{\mathbf{x}}_1^{(t-1)} \bar{\mathbf{x}}_1^{(t-1)}) - \mathbf{x}_1^* + \alpha \mathbf{h}_1^{(t-1)} \\ \text{or, } \|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\| &= \|\bar{\mathbf{x}}_1^{(t-1)} + \frac{\alpha}{M} (\mathbf{C}\bar{\mathbf{x}}_1^{(t-1)} - (\bar{\mathbf{x}}_1^{(t-1)})^T \mathbf{C}\bar{\mathbf{x}}_1^{(t-1)} \bar{\mathbf{x}}_1^{(t-1)}) - \mathbf{x}_1^* + \alpha \mathbf{h}_1^{(t-1)}\| \\ \text{or, } \|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\| &\leq \|\bar{\mathbf{x}}_1^{(t-1)} + \frac{\alpha}{M} (\mathbf{C}\bar{\mathbf{x}}_1^{(t-1)} - (\bar{\mathbf{x}}_1^{(t-1)})^T \mathbf{C}\bar{\mathbf{x}}_1^{(t-1)} \bar{\mathbf{x}}_1^{(t-1)}) - \mathbf{x}_1^*\| + \alpha \|\mathbf{h}_1^{(t-1)}\|. \end{aligned} \quad (3.23)$$

We saw in Section 3.4.1 that an iterate of the form

$$\bar{\mathbf{x}}_1^{(t)} = \bar{\mathbf{x}}_1^{(t-1)} + \frac{\alpha}{M} (\mathbf{C}\bar{\mathbf{x}}_1^{(t-1)} - (\bar{\mathbf{x}}_1^{(t-1)})^T \mathbf{C}\bar{\mathbf{x}}_1^{(t-1)} \bar{\mathbf{x}}_1^{(t-1)})$$

converges linearly to $\mathbf{x}_1^* = \pm \mathbf{q}_1$ for certain conditions on the step size and the initial point.

Thus,

$$\|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\| \leq \rho_1 \|\bar{\mathbf{x}}_1^{(t-1)} - \mathbf{x}_1^*\| + \alpha \|\mathbf{h}_1^{(t-1)}\|, \quad \text{where } \rho_1 = \frac{1 + \frac{\alpha}{M} \lambda_2}{1 + \frac{\alpha}{M} \lambda_1}.$$

The term $\mathbf{h}_1^{(t-1)}$ in the above equation appears due to the distributed nature of the algorithm and can be bounded separately. Specifically, we prove in Lemma 9, that

$$\|\mathbf{h}_1^{(t-1)}\| \leq 9\lambda_1 b_1 \left(\beta^{t-1} + \frac{\alpha}{1-\beta} \right).$$

Thus,

$$\begin{aligned} \|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\| &\leq \rho_1 \|\bar{\mathbf{x}}_1^{(t-1)} - \mathbf{x}_1^*\| + 9\alpha\lambda_1 b_1 \left(\beta^{t-1} + \frac{\alpha}{1-\beta} \right) \\ &\leq \rho_1 \left(\rho_1 \|\bar{\mathbf{x}}_1^{(t-2)} - \mathbf{x}_1^*\| + 9\alpha\lambda_1 b_1 \beta^{t-2} + 9\alpha\lambda_1 b_1 \left(\frac{\alpha}{1-\beta} \right) \right) + 9\alpha\lambda_1 b_1 \left(\beta^{t-1} + \frac{\alpha}{1-\beta} \right) \\ &\leq \rho_1^t \|\bar{\mathbf{x}}_1^{(0)} - \mathbf{x}_1^*\| + 9\alpha\lambda_1 b_1 \sum_{r=0}^{t-1} (\rho_1 \beta^{-1})^r \beta^{t-1} + \frac{1}{1-\rho_1} 9\alpha\lambda_1 b_1 \left(\frac{\alpha}{1-\beta} \right). \end{aligned}$$

Since $\rho_1, \beta < 1$, we have the following two cases:

1. $\rho_1 \leq \beta \implies \rho_1 \beta^{-1} \leq 1$. Then, $\sum_{r=0}^{t-1} (\rho_1 \beta^{-1})^r \beta^{t-1} \leq \sum_{r=0}^{t-1} \beta^{t-1} = t\beta^{t-1}$.
2. $\rho_1 > \beta$. Then $\sum_{r=0}^{t-1} (\rho_1 \beta^{-1})^r \beta^{t-1} = \beta^{t-1} + \rho_1 \beta^{t-2} + \dots + \rho_1^{t-1} < \rho_1^{t-1} + \dots + \rho_1^{t-1} = t\rho_1^{t-1}$.

Therefore,

$$\|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\| \leq \rho_1^t \|\bar{\mathbf{x}}_1^{(0)} - \mathbf{x}_1^*\| + c_1 t \max\{\rho_1^{t-1}, \beta^{t-1}\} + \frac{c_1}{1 - \rho_1} \left(\frac{\alpha}{1 - \beta} \right), \quad \text{where } c_1 = 9\alpha\lambda_1 b_1. \quad (3.24)$$

Consequently, from (3.22) and (3.24), we get

$$\begin{aligned} \|\mathbf{x}_{i,1}^{(t)} - \mathbf{x}_1^*\| &\leq b_1(\beta^t + \frac{\alpha}{1 - \beta}) + \rho_1^t \|\bar{\mathbf{x}}_1^{(0)} - \mathbf{x}_1^*\| + c_1 t \max\{\rho_1^{t-1}, \beta^{t-1}\} + \frac{c_1}{1 - \rho_1} \left(\frac{\alpha}{1 - \beta} \right) \\ &= \rho_1^t \|\bar{\mathbf{x}}_1^{(0)} - \mathbf{x}_1^*\| + b_1 \beta^t + c_1 t \max\{\rho_1^{t-1}, \beta^{t-1}\} + \left(\frac{c_1}{1 - \rho_1} + b_1 \right) \left(\frac{\alpha}{1 - \beta} \right). \end{aligned}$$

This proves that $\mathbf{x}_{i,1}^{(t)}$ converges to a neighborhood of $\mathbf{x}_1^* = \mathbf{q}_1$ or $\mathbf{x}_1^* = -\mathbf{q}_1$ at a linear rate. \square

Case II for Induction $-1 < k \leq K$: For the remainder of the eigenvectors, we proceed with the proof of convergence by induction. Since we have already proven the base case, we can assume there exist constants $c_{i,p} > 0$ and $\theta_{i,p} < 1$ such that

1. $\|\mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T - \mathbf{q}_p \mathbf{q}_p^T\| \leq c_{i,p}(\theta_{i,p}^t + \frac{\alpha}{1 - \beta}), \forall p = 1, \dots, k-1$, and
2. $\|\mathbf{x}_{i,p}^{(t)}\|^2 \leq 3, p = 1, \dots, k-1, i = 1, \dots, M$.

Using the inequality in 1) above, we can write $\mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T = \mathbf{q}_p \mathbf{q}_p^T + \phi_{i,p}^{(t)}, p = 1, \dots, k-1$ such that $\|\phi_{i,p}^{(t)}\| \leq c_{i,p}(\theta_{i,p}^t + \frac{\alpha}{1 - \beta})$. This implies

$$\begin{aligned} \frac{\alpha}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} &= \frac{\alpha}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} (\mathbf{q}_p \mathbf{q}_p^T + \phi_{i,p}^{(t)}) \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} \\ &= \frac{\alpha}{M} \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)} + \alpha \bar{\psi}_k^{(t)}, \quad \text{where } \bar{\psi}_k^{(t)} = \frac{1}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} \phi_{i,p}^{(t)} \mathbf{C} \bar{\mathbf{x}}_k^{(t)}. \end{aligned}$$

Thus, we have

$$\begin{aligned} \|\bar{\psi}_k^{(t)}\| &\leq \frac{1}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} \lambda_1 \|\phi_{i,p}^{(t)}\| \|\bar{\mathbf{x}}_k^{(t)}\| \leq \frac{1}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} \sqrt{3} \lambda_1 c_{i,p} (\theta_{i,p}^t + \frac{\alpha}{1 - \beta}) \\ &\leq \frac{1}{M} \sqrt{3} \lambda_1 (k-1) M \bar{c} (\bar{\theta}^t + \frac{\alpha}{1 - \beta}) = \sqrt{3} \lambda_1 (k-1) \bar{c} (\bar{\theta}^t + \frac{\alpha}{1 - \beta}), \quad (3.25) \end{aligned}$$

where $\bar{c} = \max_{i,p}\{c_{i,p}\}$ and $\bar{\theta} = \max_{i,p}\{\theta_{i,p}\} < 1$. Consequently,

$$\begin{aligned}
\bar{\mathbf{x}}_k^{(t+1)} &= \bar{\mathbf{x}}_k^{(t)} + \frac{\alpha}{M} (\mathbf{C}\bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C}\bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)}) - \sum_{i=1}^M \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)}) + \alpha \mathbf{h}_k^{(t)} \\
&= \bar{\mathbf{x}}_k^{(t)} + \frac{\alpha}{M} (\mathbf{C}\bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C}\bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)}) - \sum_{i=1}^M \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)}) + \\
&\quad \frac{\alpha}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} (\mathbf{q}_p \mathbf{q}_p^T - \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T) \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} + \alpha \mathbf{h}_k^{(t)} \\
&= \bar{\mathbf{x}}_k^{(t)} + \frac{\alpha}{M} (\mathbf{C}\bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C}\bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)}) - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)}) - \frac{\alpha}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} \phi_{i,p}^{(t)} \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} + \alpha \mathbf{h}_k^{(t)} \\
&= \bar{\mathbf{x}}_k^{(t)} + \frac{\alpha}{M} (\mathbf{C}\bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C}\bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)}) - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)}) - \alpha \bar{\boldsymbol{\psi}}_k^{(t)} + \alpha \mathbf{h}_k^{(t)}. \tag{3.26}
\end{aligned}$$

We can now proceed with the final theorem that characterizes the convergence behavior of DSA.

Theorem 3. Suppose $\alpha \leq \frac{\min_i w_{ii}}{3\lambda_1(2K-1)}$, where λ_1 is the largest eigenvalue of \mathbf{C} and K is the number of eigenvectors to be estimated, $\mathbf{q}_k^T \mathbf{x}_{i,k}^{(0)} \neq 0$ and $\|\mathbf{x}_{i,k}^{(0)}\| = 1, \forall k = 2, \dots, K$. Then the DSA iterate for $\mathbf{x}_{i,k}^{(t)}$ given by (3.19) converges at a linear rate to an $\mathcal{O}(\alpha)$ neighborhood of the eigenvector \mathbf{q}_k corresponding to the k^{th} largest eigenvalue λ_k of the global covariance matrix \mathbf{C} at each node of the network.

Proof. We know

$$\|\mathbf{x}_{i,k}^{(t)} - \mathbf{x}_k^*\| \leq \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| + \|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\|, \quad \text{where } \mathbf{x}_k^* = \pm \mathbf{q}_k. \tag{3.27}$$

Also, from Lemma 8 in the appendix we know that

$$\|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \leq b_k(\beta^t + \frac{\alpha}{1-\beta}).$$

Now, the average iterate of DSA for estimating the k^{th} eigenvector is

$$\begin{aligned}
\bar{\mathbf{x}}_k^{(t)} &= \bar{\mathbf{x}}_k^{(t-1)} + \frac{\alpha}{M} (\mathbf{C}\bar{\mathbf{x}}_k^{(t-1)} - ((\bar{\mathbf{x}}_k^{(t-1)})^T \mathbf{C}\bar{\mathbf{x}}_k^{(t-1)}) \bar{\mathbf{x}}_k^{(t-1)}) - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t-1)}) + \alpha \mathbf{h}_k^{(t-1)} + \alpha \bar{\boldsymbol{\psi}}_k^{(t-1)} \\
\text{or, } \|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\| &\leq \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| + \frac{\alpha}{M} (\mathbf{C}\bar{\mathbf{x}}_k^{(t-1)} - ((\bar{\mathbf{x}}_k^{(t-1)})^T \mathbf{C}\bar{\mathbf{x}}_k^{(t-1)}) \bar{\mathbf{x}}_k^{(t-1)}) - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t-1)}) - \mathbf{x}_k^*\| + \\
&\quad \alpha \|\mathbf{h}_k^{(t-1)}\| + \alpha \|\bar{\boldsymbol{\psi}}_k^{(t-1)}\|.
\end{aligned}$$

We know from the discussion in Section 3.4.1 that for an iterate of the form

$$\bar{\mathbf{x}}_k^{(t)} = \bar{\mathbf{x}}_k^{(t-1)} + \frac{\alpha}{M} (\mathbf{C}\bar{\mathbf{x}}_k^{(t-1)} - ((\bar{\mathbf{x}}_k^{(t-1)})^T \mathbf{C}\bar{\mathbf{x}}_k^{(t-1)}) \bar{\mathbf{x}}_k^{(t-1)}) - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t-1)}),$$

there exists a constant $\rho'_k < 1$ such that $\|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\| \leq \rho'_k \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\|$. Thus,

$$\|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\| \leq \rho'_k \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| + \alpha \|\mathbf{h}_k^{(t-1)}\| + \alpha \|\bar{\boldsymbol{\psi}}_k^{(t-1)}\|.$$

Now, the term $\|\mathbf{h}_k^{(t-1)}\|$ was bounded in Lemma 9 as

$$\|\mathbf{h}_k^{(t-1)}\| \leq 3(k+2)\lambda_1 b_k (\beta^{t-1} + \frac{\alpha}{1-\beta}). \quad (3.28)$$

Thus, using (3.25) and (3.28), we can write

$$\begin{aligned} \|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\| &\leq \rho'_k \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| + \alpha(3(k+2)\lambda_1 b_k (\beta^{t-1} + \frac{\alpha}{1-\beta})) + \alpha(\sqrt{3}\lambda_1(k-1)\bar{c}(\bar{\theta}^{t-1} + \frac{\alpha}{1-\beta})) \\ &\leq \rho'_k \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| + c_k \max\{\beta^{t-1}, \bar{\theta}^{t-1}\} + c_k \frac{\alpha}{1-\beta} \\ &\leq \rho'_k \left(\rho'_k \|\bar{\mathbf{x}}_k^{(t-2)} - \mathbf{x}_k^*\| + c_k \max\{\beta^{t-2}, \bar{\theta}^{t-2}\} + c_k \frac{\alpha}{1-\beta} \right) + c_k \max\{\beta^{t-1}, \bar{\theta}^{t-1}\} + c_k \frac{\alpha}{1-\beta} \\ &\leq \rho'_k \|\bar{\mathbf{x}}_k^{(0)} - \mathbf{x}_k^*\| + c_k \sum_{r=0}^{t-1} (\rho'_k \max\{\beta, \bar{\theta}\}^{-1})^r \max\{\beta, \bar{\theta}\}^{t-1} + \frac{c_k}{1-\rho'_k} \left(\frac{\alpha}{1-\beta} \right) \\ &\leq \rho'_k \|\bar{\mathbf{x}}_k^{(0)} - \mathbf{x}_k^*\| + c_k t \max\{\rho_k'^{t-1}, \beta^{t-1}, \bar{\theta}^{t-1}\} + \frac{c_k}{1-\rho'_k} \left(\frac{\alpha}{1-\beta} \right), \end{aligned}$$

where $c_k = \max\{\alpha(3(k+2)\lambda_1 b_k), \alpha(\sqrt{3}\lambda_1(k-1)\bar{c})\}$. Consequently, from (3.27) and Lemma 8 we get

$$\begin{aligned} \|\mathbf{x}_{i,k}^{(t)} - \mathbf{x}_k^*\| &\leq b_k \left(\beta^t + \frac{\alpha}{1-\beta} \right) + \rho_k'^t \|\bar{\mathbf{x}}_k^{(0)} - \mathbf{x}_k^*\| + c_k t \max\{\rho_k'^{t-1}, \beta^{t-1}, \bar{\theta}^{t-1}\} + \frac{c_k}{1-\rho'_k} \left(\frac{\alpha}{1-\beta} \right) \\ &= \rho_k^t \|\bar{\mathbf{x}}_k^{(0)} - \mathbf{x}_k^*\| + b_k \beta^t + c_k (t-1) \max\{\rho_k^{t-1}, \beta^{t-1}, \bar{\theta}^{t-1}\} + \left(\frac{c_k}{1-\rho_k} + b_k \right) \left(\frac{\alpha}{1-\beta} \right). \end{aligned}$$

This proves that $\mathbf{x}_{i,k}^{(t)}$ converges to a neighborhood of $\mathbf{x}_k^* = \mathbf{q}_k$ or $\mathbf{x}_k^* = -\mathbf{q}_k$ at a linear rate. \square

It is noteworthy that if decaying step sizes α_t are used such that $\alpha_t \rightarrow 0$ as $t \rightarrow \infty$ (instead of constant α), the convergence will be exact but not linear. The rate in that case will be dominated by the rate of decay of α_t .

3.5 Statements and Proofs of Auxiliary Lemmas

3.5.1 Statement and Proof of Supporting Lemma for Modified GHA

In order to prove the convergence of modified GHA, we first need to prove that the iterates remain bounded for certain condition on the step size. This is done in the following Lemma 1.

Lemma 1. Assume $\|\mathbf{x}_k^{(0)}\| = 1, \forall k$. If the step size is bounded above as $\alpha \leq \frac{1}{3\lambda_1(2K-1)}$, where λ_1 is the largest eigenvalue of \mathbf{C} and K is the number of eigenvectors to be estimated, then

$$\forall t, \quad \|\mathbf{x}_k^{(t)}\| < \sqrt{3} \quad \text{and} \quad (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} < \frac{1}{\alpha}. \quad (3.29)$$

Proof. From (3.12), we know the iterate for k^{th} eigenvector estimate is

$$\begin{aligned}
\mathbf{x}_k^{(t+1)} &= \mathbf{x}_k^{(t)} + \alpha \left(\mathbf{C}\mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C}\mathbf{x}_k^{(t)} \right) \\
&= \mathbf{x}_k^{(t)} + \alpha \left(\mathbf{C}\mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T \mathbf{x}_k^{(t)} \right) \\
&= \mathbf{x}_k^{(t)} + \alpha \left(\tilde{\mathbf{C}}_k \mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)} \right),
\end{aligned}$$

where $\tilde{\mathbf{C}}_k = \mathbf{C} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T$. Notice that $\tilde{\mathbf{C}}_k^2 = \mathbf{C}^2 - \sum_{p=1}^{k-1} \lambda_p^2 \mathbf{q}_p \mathbf{q}_p^T$. Hence,

$$\begin{aligned}
\|\mathbf{x}_k^{(t+1)}\|^2 &= \|\mathbf{x}_k^{(t)} + \alpha(\tilde{\mathbf{C}}_k \mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)})\|^2 \\
&= \|\mathbf{x}_k^{(t)}\|^2 + \alpha^2 \|\tilde{\mathbf{C}}_k \mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)}\|^2 + 2\alpha (\mathbf{x}_k^{(t)})^T (\tilde{\mathbf{C}}_k \mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)}) \\
&= \|\mathbf{x}_k^{(t)}\|^2 + \alpha^2 ((\mathbf{x}_k^{(t)})^T \tilde{\mathbf{C}}_k^2 \mathbf{x}_k^{(t)} + ((\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})^2 \|\mathbf{x}_k^{(t)}\|^2 - 2(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} (\mathbf{x}_k^{(t)})^T \tilde{\mathbf{C}}_k \mathbf{x}_k^{(t)}) \\
&\quad + 2\alpha ((\mathbf{x}_k^{(t)})^T \tilde{\mathbf{C}}_k \mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} \|\mathbf{x}_k^{(t)}\|^2) \\
&= \|\mathbf{x}_k^{(t)}\|^2 + \alpha^2 ((\mathbf{x}_k^{(t)})^T (\mathbf{C}^2 - \sum_{p=1}^{k-1} \lambda_p^2 \mathbf{q}_p \mathbf{q}_p^T) \mathbf{x}_k^{(t)} + ((\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})^2 \|\mathbf{x}_k^{(t)}\|^2 - 2(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} \times \\
&\quad (\mathbf{x}_k^{(t)})^T (\mathbf{C} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T) \mathbf{x}_k^{(t)}) + 2\alpha ((\mathbf{x}_k^{(t)})^T (\mathbf{C} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T) \mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} \|\mathbf{x}_k^{(t)}\|^2) \\
&= \|\mathbf{x}_k^{(t)}\|^2 + \alpha^2 ((\mathbf{x}_k^{(t)})^T \mathbf{C}^2 \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \lambda_p^2 (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2 + ((\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})^2 (\|\mathbf{x}_k^{(t)}\|^2 - 2) \\
&\quad + 2(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2) + 2\alpha ((\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} (1 - \|\mathbf{x}_k^{(t)}\|^2) - \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2).
\end{aligned} \tag{3.30}$$

We now split our analysis into three cases based on the range of values of $\|\mathbf{x}_k^{(t)}\|^2$.

Case I: Let $\|\mathbf{x}_k^{(t)}\|^2 \leq 1$. Then we see from (3.30) that

$$\begin{aligned}
\|\mathbf{x}_k^{(t+1)}\|^2 &\leq 1 + \alpha^2 (\lambda_1^2 + 2\lambda_1 \sum_{p=1}^{k-1} \lambda_p) + 2\alpha \lambda_1 \leq 1 + \alpha^2 (\lambda_1^2 + 2\lambda_1 \sum_{p=1}^{k-1} \lambda_1) + 2\alpha \lambda_1 \\
&\leq 1 + \alpha^2 \lambda_1^2 (2K - 1) + 2\alpha \lambda_1 \sqrt{2K - 1} = (1 + \alpha \lambda_1 \sqrt{2K - 1})^2 \\
&\leq 2(1 + \alpha^2 \lambda_1^2 (2K - 1)) \leq 2(1 + \frac{1}{9(2K - 1)}) \leq 2(1 + \frac{1}{9}) < 3.
\end{aligned}$$

Case II: Now suppose $1 < \|\mathbf{x}_k^{(t)}\|^2 \leq 2$. Then from (3.30) we have

$$\begin{aligned}
\|\mathbf{x}_k^{(t+1)}\|^2 &\leq 2 + \alpha^2 (2\lambda_1^2 + 2\lambda_1 \sum_{p=1}^{k-1} 2\lambda_p) \leq 2 + \alpha^2 (2\lambda_1^2 + 2\lambda_1 \sum_{p=1}^{k-1} 2\lambda_1) \\
&\leq 2(1 + \frac{1}{9(2K - 1)}) \leq 2(1 + \frac{1}{9}) < 3, \quad \text{using similar steps as Case I.}
\end{aligned}$$

Case III: Finally suppose $2 < \|\mathbf{x}_k^{(t)}\|^2 < 3$. Then from (3.30) we get

$$\begin{aligned} \|\mathbf{x}_k^{(t+1)}\|^2 &< 3 + \alpha^2 ((\mathbf{x}_k^{(t)})^T \mathbf{C}^2 \mathbf{x}_k^{(t)}) - \sum_{p=1}^{k-1} \lambda_p^2 (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2 + ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (\|\mathbf{x}_k^{(t)}\|^2 - 2) \\ &\quad + 2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2 + 2\alpha ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (1 - \|\mathbf{x}_k^{(t)}\|^2) - \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2). \end{aligned}$$

To show that $\|\mathbf{x}_k^{(t+1)}\|^2 < 3$, we have to show

$$\begin{aligned} &\alpha^2 ((\mathbf{x}_k^{(t)})^T \mathbf{C}^2 \mathbf{x}_k^{(t)}) - \sum_{p=1}^{k-1} \lambda_p^2 (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2 + ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (\|\mathbf{x}_k^{(t)}\|^2 - 2) \\ &2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2 + 2\alpha ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (1 - \|\mathbf{x}_k^{(t)}\|^2) - \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2) \leq 0 \\ \Leftrightarrow \alpha &\leq \frac{2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1) + 2 \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2}{(\mathbf{x}_k^{(t)})^T \mathbf{C}^2 \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \lambda_p^2 (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2 + ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (\|\mathbf{x}_k^{(t)}\|^2 - 2) + 2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2}. \end{aligned} \quad (3.31)$$

We now find a lower bound of the right hand side of (3.31). Note that

$$2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1) + 2 \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2 \geq 2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1) \quad (3.32)$$

$$\begin{aligned} \text{and } &(\mathbf{x}_k^{(t)})^T \mathbf{C}^2 \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \lambda_p^2 (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2 + ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (\|\mathbf{x}_k^{(t)}\|^2 - 2) + 2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2 \\ &\leq (\mathbf{x}_k^{(t)})^T \mathbf{C}^2 \mathbf{x}_k^{(t)} + ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (\|\mathbf{x}_k^{(t)}\|^2 - 2) + 2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2. \end{aligned} \quad (3.33)$$

Now, $\frac{(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}}{(\mathbf{x}_k^{(t)})^T \mathbf{C}^2 \mathbf{x}_k^{(t)}}$ is a generalized Rayleigh quotient whose maximum and minimum values are the largest and smallest eigenvalues of the generalized eigenvalue problem $\mathbf{C}\mathbf{y} = \lambda \mathbf{C}^2 \mathbf{y}$. Since the eigenvectors of \mathbf{C} and \mathbf{C}^2 are the same, the largest and smallest eigenvalues of the generalized problems are $\frac{1}{\lambda_d}$ and $\frac{1}{\lambda_1}$, respectively, where λ_1 and λ_d are the largest and smallest eigenvalues of \mathbf{C} . Thus, $(\mathbf{x}_k^{(t)})^T \mathbf{C}^2 \mathbf{x}_k^{(t)} \leq \lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}$. Also, since $\mathbf{q}_p^T \mathbf{x}_k^{(t)} \leq \|\mathbf{q}_p\| \|\mathbf{x}_k^{(t)}\|$, we

have the right hand side of (3.33)

$$\begin{aligned}
& \lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} + ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (\|\mathbf{x}_k^{(t)}\|^2 - 2) + 2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \sum_{p=1}^{k-1} \lambda_p \|\mathbf{x}_k^{(t)}\|^2 \\
&= (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\lambda_1 + (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 2) + 2 \sum_{p=1}^{k-1} \lambda_p \|\mathbf{x}_k^{(t)}\|^2) \\
&\leq (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\lambda_1 + \lambda_1 \|\mathbf{x}_k^{(t)}\|^2 (\|\mathbf{x}_k^{(t)}\|^2 - 2) + 2 \sum_{p=1}^{k-1} \lambda_1 \|\mathbf{x}_k^{(t)}\|^2) \\
&= \lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (1 + \|\mathbf{x}_k^{(t)}\|^4 - 2\|\mathbf{x}_k^{(t)}\|^2 + 2(k-1)\|\mathbf{x}_k^{(t)}\|^2) \\
&= \lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} ((\|\mathbf{x}_k^{(t)}\|^2 - 1)^2 + 2(k-1)\|\mathbf{x}_k^{(t)}\|^2) \\
&= \lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1)((\|\mathbf{x}_k^{(t)}\|^2 - 1) + 2(k-1)\|\mathbf{x}_k^{(t)}\|^2) \\
&< \lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1)((3-1) + 2(k-1)2), \quad \text{since } \frac{\|\mathbf{x}_k^{(t)}\|^2}{(\|\mathbf{x}_k^{(t)}\|^2 - 1)} < 2 \\
&= 2\lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1)(2k-1) \leq 2\lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1)(2K-1).
\end{aligned}$$

Hence, we have that the right hand side of (3.31) exceeds

$$\frac{2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1)}{2\lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1)(2K-1)} = \frac{1}{\lambda_1(2K-1)} > \frac{1}{3\lambda_1(2K-1)}.$$

Thus, if $\alpha \leq \frac{1}{3\lambda_1(2K-1)}$, then $\|\mathbf{x}_k^{(t)}\|^2 < 3$.

Next,

$$0 \leq (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \leq \lambda_1 \|\mathbf{x}_k^{(t)}\|^2 < 3\lambda_1 \leq 3(2K-1)\lambda_1 \leq \frac{1}{\alpha}. \quad (3.34)$$

Hence, $(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} < \frac{1}{\alpha}$. \square

Using Lemma 1, we can now prove the following Lemma 2.

Lemma 2. Suppose $\mathbf{q}_k^T \mathbf{x}_k^{(0)} = z_{k,k}^{(0)} \neq 0$ and $(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} < \frac{1}{\alpha}$, then

$$(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} > \min\{(1 - 3\alpha\lambda_1)^2 \lambda_m, (\tilde{\mathbf{x}}_k^{(0)})^T \mathbf{C} \tilde{\mathbf{x}}_k^{(0)}\}, \quad \forall t.$$

Proof. We know $0 \leq (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \leq \lambda_1 \|\mathbf{x}_k^{(t)}\|^2 < 3\lambda_1$ using Lemma 1. Let $\lambda_m, m > K$ be the

smallest non-zero eigenvalue of \mathbf{C} . Now, if $\lambda_m \leq (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} < 3\lambda_1$, then

$$\begin{aligned}
(\mathbf{x}_k^{(t+1)})^T \mathbf{C} \mathbf{x}_k^{(t+1)} &= \sum_{l=1}^d \lambda_l (z_{k,l}^{(t+1)})^2 \\
&= \sum_{l=1}^{k-1} \lambda_l (z_{k,l}^{(t+1)})^2 + \sum_{l=k}^d \lambda_l (z_{k,l}^{(t+1)})^2 \\
&= \sum_{l=1}^{k-1} \lambda_l (1 - \alpha (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (z_{k,l}^{(t)})^2 + \sum_{l=k}^d \lambda_l (1 + \alpha (\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 \\
&\geq (1 - \alpha (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 \sum_{l=1}^{k-1} \lambda_l (z_{k,l}^{(t)})^2 + (1 + \alpha (\lambda_m - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 \sum_{l=k}^d \lambda_l (z_{k,l}^{(t)})^2 \\
&> (1 - \alpha (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 \sum_{l=1}^{k-1} \lambda_l (z_{k,l}^{(t)})^2 + (1 - \alpha (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 \sum_{l=k}^d \lambda_l (z_{k,l}^{(t)})^2 \\
&= (1 - \alpha (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 \sum_{l=1}^d \lambda_l (z_{k,l}^{(t)})^2 \\
&> (1 - 3\alpha \lambda_1)^2 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \geq (1 - 3\alpha \lambda_1)^2 \lambda_m.
\end{aligned} \tag{3.35}$$

Also, from (4.7), we have $\mathbf{x}_k^{(t)} = \sum_{l=1}^d z_{k,l}^{(t)} \mathbf{q}_l = \sum_{l=1}^{k-1} z_{k,l}^{(t)} \mathbf{q}_l + \sum_{l=k}^d z_{k,l}^{(t)} \mathbf{q}_l$. Let $\sum_{l=1}^{k-1} z_{k,l}^{(t)} \mathbf{q}_l = \mathbf{x}_k'^{(t)}$ and $\sum_{l=k}^d z_{k,l}^{(t)} \mathbf{q}_l = \tilde{\mathbf{x}}_k^{(t)}$. Thus, $(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} = (\tilde{\mathbf{x}}_k^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_k^{(t)} + (\mathbf{x}_k'^{(t)})^T \mathbf{C} \mathbf{x}_k'^{(t)}$.

Now, if $(\tilde{\mathbf{x}}_k^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_k^{(t)} \leq (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} < \lambda_m$ then

$$\begin{aligned}
(\mathbf{x}_k^{(t+1)})^T \mathbf{C} \mathbf{x}_k^{(t+1)} &\geq (\tilde{\mathbf{x}}_k^{(t+1)})^T \mathbf{C} \tilde{\mathbf{x}}_k^{(t+1)} = \sum_{l=k}^d \lambda_l (z_{k,l}^{(t+1)})^2 \\
&= \sum_{l=k}^d \lambda_l (1 + \alpha (\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 \\
&\geq (1 + \alpha (\lambda_m - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 \sum_{l=k}^d \lambda_l (z_{k,l}^{(t)})^2 > (\tilde{\mathbf{x}}_k^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_k^{(t)}.
\end{aligned} \tag{3.36}$$

Combining (3.35) and (3.36), we have

$$(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} > \min\{(1 - 3\alpha \lambda_1)^2 \lambda_m, (\tilde{\mathbf{x}}_k^{(0)})^T \mathbf{C} \tilde{\mathbf{x}}_k^{(0)}\}. \tag{3.37}$$

□

Using the above two lemmas, we now prove the Lemma 3, Lemma 4 and Lemma 5 that is required to prove Theorem 1.

Lemma 3. Let $\eta = \min\{(1 - 3\alpha \lambda_1)^2 \lambda_m, (\tilde{\mathbf{x}}_k^{(0)})^T \mathbf{C} \tilde{\mathbf{x}}_k^{(0)}\}$. Now, suppose $\eta < (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} < \frac{1}{\alpha}$, then the following is true for $\gamma = 1 - \alpha\eta$, and some constant $a_1 > 0$:

$$\sum_{l=1}^{k-1} (z_{k,l}^{(t+1)})^2 < a_1 \gamma^{t+1}. \tag{3.38}$$

Proof. For $l = 1, \dots, k-1$, we know from (4.10)

$$\begin{aligned} z_{k,l}^{(t+1)} &= (1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}) z_{k,l}^{(t)} \\ \text{or, } (z_{k,l}^{(t+1)})^2 &= (1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (z_{k,l}^{(t)})^2. \end{aligned}$$

Let $\min\{(1 - 3\alpha\lambda_1)^2 \lambda_m, (\tilde{\mathbf{x}}_k^{(0)})^T \mathbf{C} \tilde{\mathbf{x}}_k^{(0)}\} = \eta$. Since $0 < \eta < (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} < \frac{1}{\alpha}$ (from (3.34) and (3.37)), we have $0 < 1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} < 1 - \alpha\eta < 1$. Therefore,

$$\sum_{l=1}^{k-1} (z_{k,l}^{(t+1)})^2 < \sum_{l=1}^{k-1} \gamma (z_{k,l}^{(t)})^2 < \gamma^{t+1} \sum_{l=1}^{k-1} (z_{k,l}^{(0)})^2 = a_1 \gamma^{t+1}, \quad \text{where } \gamma = (1 - \alpha\eta)^2. \quad (3.39)$$

□

Lemma 4. Suppose $z_{k,k}^{(0)} \neq 0$ and $(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} < \frac{1}{\alpha}$. Then the following is true for $\rho_k = \left(\frac{1+\alpha\lambda_{k+1}}{1+\alpha\lambda_k}\right)^2 < 1$ and some constant $a_2 > 0$:

$$\sum_{l=k+1}^d (z_{k,l}^{(t+1)})^2 \leq a_2 \rho_k^{t+1}. \quad (3.40)$$

Proof. For $l = k, \dots, d$ we know from (4.11) that $z_{k,l}^{(t+1)} = (1 + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})) z_{k,l}^{(t)}$. If $(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} < \frac{1}{\alpha}$, we have $1 + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}) > \alpha\lambda_l \geq 0, \forall l = k, \dots, d$.

Thus, we have for $l = k+1, \dots, d$,

$$\begin{aligned} \left(\frac{z_{k,l}^{(t+1)}}{z_{k,k}^{(t+1)}}\right)^2 &= \left(\frac{1 + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})}{1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})}\right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2 \\ &= \left(1 - \frac{\alpha(\lambda_k - \lambda_l)}{1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})}\right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2 \\ &\leq \left(1 - \frac{\alpha(\lambda_k - \lambda_l)}{1 + \alpha\lambda_k}\right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2 \\ &= \left(\frac{1 + \alpha\lambda_l}{1 + \alpha\lambda_k}\right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2 \leq \left(\frac{1 + \alpha\lambda_{k+1}}{1 + \alpha\lambda_k}\right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2 \\ &= \rho_k \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2, \quad \rho_k = \left(\frac{1 + \alpha\lambda_{k+1}}{1 + \alpha\lambda_k}\right)^2 < 1. \end{aligned}$$

Therefore, for $l = k+1, \dots, d$, $(z_{k,l}^{(t+1)})^2 \leq \rho_k^{t+1} \left(\frac{z_{k,l}^{(0)}}{z_{k,k}^{(0)}}\right)^2 (z_{k,k}^{(t+1)})^2$. Since $\|\mathbf{x}_k^{(t+1)}\|^2 \leq 3$ and $\|\mathbf{x}_k^{(0)}\| = 1$, hence $(z_{k,k}^{(t+1)})^2 \leq 3$ and $z_{k,l}^{(0)} \leq 1$. Also, because of the assumption $z_{k,k}^{(0)} \neq 0$, let us assume $(z_{k,k}^{(0)})^2 > \tilde{\eta}$. Thus, we can write

$$\sum_{l=k+1}^d (z_{k,l}^{(t+1)})^2 \leq \rho_k^{t+1} \sum_{l=k+1}^d \frac{3}{\tilde{\eta}} = a_2 \rho_k^{t+1}. \quad (3.41)$$

□

Lemma 5. Suppose $(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} < \frac{1}{\alpha}$ and $(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} > \min\{(1 - 3\alpha\lambda_1)^2 \lambda_m, (\tilde{\mathbf{x}}_k^{(0)})^T \mathbf{C} \tilde{\mathbf{x}}_k^{(0)}\}$.

Then there exists constants $0 < \delta, \gamma_1 < 1, a_4 > 0$ such that

$$|\lambda_k - (\mathbf{x}_k^{(t+1)})^T \mathbf{C} \mathbf{x}_k^{(t+1)}| \leq ta_4(\delta^{t+1} + \max\{\delta^t, \gamma_1^t\}). \quad (3.42)$$

Proof.

$$\begin{aligned} (\mathbf{x}_k^{(t+1)})^T \mathbf{C} \mathbf{x}_k^{(t+1)} &= \sum_{l=1}^{k-1} \lambda_l (1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (z_{k,l}^{(t)})^2 + \sum_{l=k}^d \lambda_l (1 + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 \\ &= \sum_{l=1}^{k-1} \lambda_l (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 + \sum_{l=k}^d \lambda_l (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 \\ &\quad + \sum_{l=1}^{k-1} \lambda_l (1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (z_{k,l}^{(t)})^2 - \sum_{l=1}^{k-1} \lambda_l (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 + \\ &\quad \sum_{l=k+1}^d \lambda_l (1 + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 - \sum_{l=k+1}^d \lambda_l (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 \\ &= \sum_{l=1}^d \lambda_l (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 + P^{(t)} \\ &= (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} + P^{(t)}, \end{aligned}$$

where

$$\begin{aligned} P^{(t)} &= \sum_{l=1}^{k-1} \lambda_l (1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (z_{k,l}^{(t)})^2 - \sum_{l=1}^{k-1} \lambda_l (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 \\ &\quad + \sum_{l=k+1}^d \lambda_l (1 + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 - \sum_{l=k+1}^d \lambda_l (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2. \end{aligned}$$

Now,

$$\begin{aligned} &\lambda_k - (\mathbf{x}_k^{(t+1)})^T \mathbf{C} \mathbf{x}_k^{(t+1)} \\ &= \lambda_k - (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} - P^{(t)} \\ &= \lambda_k - (1 + \alpha^2(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 + 2\alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} - P^{(t)} \\ &= \lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} - (\alpha^2(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 + 2\alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} - P^{(t)} \\ &= \lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} - (\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})(\alpha^2(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}) + 2\alpha)(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} - P^{(t)} \\ &= (\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})(1 - \alpha^2(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} - 2\alpha(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}) - P^{(t)}. \end{aligned}$$

Let us denote $V^{(t)} = |\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}|$. Then,

$$\begin{aligned} V^{(t+1)} &\leq V^{(t)} |1 - \alpha^2(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} - 2\alpha(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}| + |P^{(t)}| \\ &\leq V^{(t)} \max\{(1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2, \alpha^2 \lambda_k (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}\} + |P^{(t)}|. \end{aligned}$$

Also from (3.34) and (3.37), $0 < \alpha\eta < \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} < 1$ and $\alpha^2\lambda_k(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} < \alpha\lambda_k$. Denote $\delta = \max\{(1 - \alpha\eta)^2, \alpha\lambda_k\}$. Since $\alpha\lambda_k < \alpha\lambda_1 < 1$, hence $0 < \delta < 1$. Thus,

$$V^{(t+1)} \leq \delta V^{(t)} + |P^{(t)}|. \quad (3.43)$$

Next, we bound $|P^{(t)}|$ as follows:

$$\begin{aligned} |P^{(t)}| &= \left| \sum_{l=1}^{k-1} \lambda_l ((1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})^2 - (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2) (z_{k,l}^{(t)})^2 \right. \\ &\quad \left. + \sum_{l=k+1}^d \lambda_l ((1 + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2 - (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2) (z_{k,l}^{(t)})^2 \right| \\ &= \left| \sum_{l=1}^{k-1} \lambda_l (-\alpha\lambda_k) (2 + \alpha\lambda_k - 2\alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}) (z_{k,l}^{(t)})^2 \right. \\ &\quad \left. + \sum_{l=k+1}^d \lambda_l \alpha(\lambda_l - \lambda_k) (2 + \alpha(\lambda_k + \lambda_l) - 2\alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}) (z_{k,l}^{(t)})^2 \right| \\ &\leq \sum_{l=1}^{k-1} \lambda_l |(-\alpha\lambda_k) (2 + \alpha\lambda_k - 2\alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}) (z_{k,l}^{(t)})^2| \\ &\quad + \sum_{l=k+1}^d \lambda_l |\alpha(\lambda_l - \lambda_k) (2 + \alpha(\lambda_k + \lambda_l) - 2\alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}) (z_{k,l}^{(t)})^2| \\ &\leq \sum_{l=1}^{k-1} \lambda_l \alpha\lambda_k (2 + \alpha\lambda_k) (z_{k,l}^{(t)})^2 + \sum_{l=k+1}^d \lambda_l \alpha(\lambda_k - \lambda_l) (2 + \alpha(\lambda_k + \lambda_l)) (z_{k,l}^{(t)})^2 \\ &< \sum_{l=1}^{k-1} \lambda_l \alpha\lambda_k (2 + \alpha\lambda_k) (z_{k,l}^{(t)})^2 + \sum_{l=k+1}^d \lambda_l (2\alpha\lambda_k + \alpha^2\lambda_k^2) (z_{k,l}^{(t)})^2 \\ &< \sum_{l=1}^{k-1} 3\lambda_1 (z_{k,l}^{(t)})^2 + \sum_{l=k+1}^d 3\lambda_1 (z_{k,l}^{(t)})^2, \quad \text{since } \alpha\lambda_k < 1 \text{ and } \lambda_l < \lambda_1 \\ &= 3\lambda_1 \left(\sum_{l=1}^{k-1} (z_{k,l}^{(t)})^2 + \sum_{l=k+1}^d (z_{k,l}^{(t)})^2 \right) < 3\lambda_1 (a_1\gamma^t + a_2\rho_k^t) \quad \text{using Lemma 3 and 4} \\ &\leq a_3\gamma_1^t, \quad \text{where } a_3 = \max\{3\lambda_1 a_1, 3\lambda_1 a_2\} \quad \text{and } \gamma_1 = \max\{\gamma, \rho_k\}. \end{aligned}$$

So from (3.43), we have $V^{(t+1)} \leq \delta V^{(t)} + a_3\gamma_1^t \leq \delta^{t+1}V^{(0)} + a_3 \sum_{r=0}^t (\delta\gamma_1^{-1})^r \gamma_1^t$. Since $\gamma_1, \delta < 1$, we have the following two cases:

1. $\delta \leq \gamma_1 \implies \delta\gamma_1^{-1} \leq 1$. Then, $\sum_{r=0}^t (\delta\gamma_1^{-1})^r \gamma_1^t \leq \sum_{r=0}^t \gamma_1^t = t\gamma_1^t$.
2. $\delta > \gamma_1$. Then $\sum_{r=0}^t (\delta\gamma_1^{-1})^r \gamma_1^t = \gamma_1^t + \delta\gamma_1^{t-1} + \dots + \delta^t < \delta^t + \dots + \delta^t = t\delta^t$.

Thus,

$$V^{(t+1)} \leq \delta^{t+1}V^{(0)} + ta_3 \max\{\delta^t, \gamma_1^t\} \leq ta_4(\delta^{t+1} + \max\{\delta^t, \gamma_1^t\}),$$

where $a_4 = \max\{V^{(0)}, a_3\}$. □

3.5.2 Statement and Proof of supporting Lemma for DSA

Lemma 6. Assume $\|\mathbf{x}_{i,k}^{(0)}\| = 1$. If the step size is bounded above as $\alpha \leq \frac{w_{ii}}{3\lambda_1(2K-1)}$, where λ_1 is the largest eigenvalue of \mathbf{C} and K is the number of eigenvectors to be estimated, then

$$\|\mathbf{x}_{i,k}^{(t)}\| < \sqrt{3} \quad \text{and} \quad (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_k^{(t)} < \frac{1}{\alpha}, \quad \forall k, t. \quad (3.44)$$

Proof. We have

$$\mathbf{x}_{i,k}^{(t+1)} = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{x}_{j,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)}) - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)}) \quad (3.45)$$

Hence,

$$\begin{aligned} \|\mathbf{x}_{i,k}^{(t+1)}\| &\leq \|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\| + \alpha \sum_{p=1}^{k-1} \|\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)}\| + \sum_{j \neq i} \|w_{ij} \mathbf{x}_{j,k}^{(t)}\| \\ &\leq \|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\| + \alpha \sum_{p=1}^{k-1} \lambda_1 \|\mathbf{x}_{i,p}^{(t)}\| \|\mathbf{x}_{i,k}^{(t)}\| \|\mathbf{x}_{i,p}^{(t)}\| + \sum_{j \neq i} w_{ij} \|\mathbf{x}_{j,k}^{(t)}\| \\ &= \|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\| + \alpha \sum_{p=1}^{k-1} \lambda_1 \|\mathbf{x}_{i,p}^{(t)}\|^2 \|\mathbf{x}_{i,k}^{(t)}\| + \sum_{j \neq i} w_{ij} \|\mathbf{x}_{j,k}^{(t)}\| \\ &\leq \|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\| + 3\alpha \lambda_1 \sum_{p=1}^{k-1} \|\mathbf{x}_{i,k}^{(t)}\| + \sum_{j \neq i} w_{ij} \|\mathbf{x}_{j,k}^{(t)}\| \\ &= \|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\| + 3(k-1)\alpha \lambda_1 \|\mathbf{x}_{i,k}^{(t)}\| + \sum_{j \neq i} w_{ij} \|\mathbf{x}_{j,k}^{(t)}\|. \end{aligned}$$

Now,

$$\begin{aligned} &\|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)}) \mathbf{x}_{i,k}^{(t)})\|^2 \\ &= w_{ii}^2 \|\mathbf{x}_{i,k}^{(t)}\|^2 + \alpha^2 \|\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)}) \mathbf{x}_{i,k}^{(t)}\|^2 + 2\alpha w_{ii} (\mathbf{x}_{i,k}^{(t)})^T (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)}) \\ &= w_{ii}^2 \|\mathbf{x}_{i,k}^{(t)}\|^2 + 2\alpha w_{ii} (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (1 - \|\mathbf{x}_{i,k}^{(t)}\|^2) + \alpha^2 (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i^2 \mathbf{x}_{i,k}^{(t)} + \alpha^2 ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2). \end{aligned}$$

Case I: Let us assume $\|\mathbf{x}_{i,k}^{(t)}\|^2 \leq 1, \forall i$. Then, we have

$$\|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\|^2 \leq (w_{ii} + \alpha \lambda_1)^2 \leq \left(w_{ii} + \frac{w_{ii}}{3(2K-1)}\right)^2.$$

Thus,

$$\begin{aligned} \|\mathbf{x}_{i,k}^{t+1}\| &\leq w_{ii} \left(1 + \frac{1}{3(2K-1)}\right) + \frac{3(k-1)}{3(2K-1)} + (1 - w_{ii}) \\ &< \frac{1}{3(2K-1)} + \frac{k-1}{2K-1} + 1 = \frac{k-0.67}{2(K-0.5)} + 1 \\ &\leq \frac{K-0.67}{2(K-0.5)} + 1 < 1.5 < \sqrt{3}. \end{aligned}$$

Case II: Now, suppose $1 \leq \|\mathbf{x}_{i,k}^{(t)}\|^2 < 2, \forall i$. Then, we get

$$\|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\|^2 \leq 2w_{ii}^2 + 2\alpha^2 \lambda_1^2 < 2(w_{ii} + \alpha \lambda_1)^2.$$

Thus, if we need $\|\mathbf{x}_{i,k}^{(t+1)}\| \leq \sqrt{3}$, the following condition should be met:

$$\begin{aligned}
\|\mathbf{x}_{i,k}^{t+1}\| &\leq \sqrt{2}w_{ii}(1 + \alpha\lambda_1) + 3(k-1)\alpha\lambda_1\sqrt{2} + (1 - w_{ii})\sqrt{2} \leq \sqrt{3} \\
&\Leftrightarrow \sqrt{2} + \sqrt{2}w_{ii}\alpha\lambda_1 + 3(k-1)\alpha\lambda_1\sqrt{2} \leq \sqrt{3} \\
&\Leftrightarrow \sqrt{2}\alpha\lambda_1 + 3(k-1)\alpha\lambda_1\sqrt{2} \leq \sqrt{3} - \sqrt{2} \\
&\Leftrightarrow \sqrt{2}\alpha\lambda_1(3k-2) \leq \sqrt{3} - \sqrt{2} \Leftrightarrow \sqrt{2}\alpha\lambda_1(3K-2) \leq \sqrt{3} - \sqrt{2} \\
&\Leftrightarrow \alpha \leq \frac{\sqrt{3} - \sqrt{2}}{\sqrt{2}\lambda_1(3K-2)} = \frac{\sqrt{1.5} - 1}{\lambda_1(3K-2)} = \frac{0.225}{\lambda_1(3K-2)}.
\end{aligned}$$

Since $\frac{0.225}{\lambda_1(3(2K-1))} < \frac{0.225}{\lambda_1(3K-2)}$, if $\alpha \leq \frac{0.225}{3\lambda_1(2K-1)}$, then $\|\mathbf{x}_{i,k}^{(t+1)}\| \leq \sqrt{3}$.

Case III: Finally, suppose $2 \leq \|\mathbf{x}_{i,k}^{(t)}\|^2 \leq 3, \forall i$. We then have the following: $\sum_{j \neq i} w_{ij} \|\mathbf{x}_{j,k}^t\| \leq \sum_{j \neq i} w_{ij} \sqrt{3} = (1 - w_{ii})\sqrt{3}$.

Now, if we desire $\|\mathbf{x}_{i,k}^{(t+1)}\| \leq \sqrt{3}$, then we need

$$\begin{aligned}
&\|w_{ii}\mathbf{x}_{i,k}^{(t)} + \alpha(\mathbf{C}_i\mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i\mathbf{x}_{i,k}^{(t)}\mathbf{x}_{i,k}^{(t)})\| + 3(k-1)\alpha\lambda_1\|\mathbf{x}_{i,k}^{(t)}\| + \sum_{j \neq i} w_{ij}\sqrt{3} \leq \sqrt{3} \\
&\Leftrightarrow \|w_{ii}\mathbf{x}_{i,k}^{(t)} + \alpha(\mathbf{C}_i\mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i\mathbf{x}_{i,k}^{(t)}\mathbf{x}_{i,k}^{(t)})\| + 3(k-1)\alpha\lambda_1\|\mathbf{x}_{i,k}^{(t)}\| + (1 - w_{ii})\sqrt{3} \leq \sqrt{3} \\
&\Leftrightarrow \|w_{ii}\mathbf{x}_{i,k}^{(t)} + \alpha(\mathbf{C}_i\mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i\mathbf{x}_{i,k}^{(t)}\mathbf{x}_{i,k}^{(t)})\| \leq \sqrt{3} - 3(k-1)\alpha\lambda_1\|\mathbf{x}_{i,k}^{(t)}\| - (1 - w_{ii})\sqrt{3} \\
&\Leftrightarrow \|w_{ii}\mathbf{x}_{i,k}^{(t)} + \alpha(\mathbf{C}_i\mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i\mathbf{x}_{i,k}^{(t)}\mathbf{x}_{i,k}^{(t)})\|^2 \leq 3w_{ii}^2 - 6\sqrt{3}(k-1)w_{ii}\alpha\lambda_1\|\mathbf{x}_{i,k}^{(t)}\| + 9\alpha^2\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2.
\end{aligned}$$

Therefore, we need

$$\begin{aligned}
&w_{ii}^2\|\mathbf{x}_{i,k}^{(t)}\|^2 + 2\alpha w_{ii}(\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i\mathbf{x}_{i,k}^{(t)}(1 - \|\mathbf{x}_{i,k}^{(t)}\|^2) + \alpha^2(\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i^2\mathbf{x}_{i,k}^{(t)} + \alpha^2((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i\mathbf{x}_{i,k}^{(t)})^2(\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) \\
&\leq 3w_{ii}^2 - 6\sqrt{3}(k-1)w_{ii}\alpha\lambda_1\|\mathbf{x}_{i,k}^{(t)}\| + 9\alpha^2\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2 \\
&\Leftrightarrow 3w_{ii}^2 + 2\alpha w_{ii}(\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i\mathbf{x}_{i,k}^{(t)}(1 - \|\mathbf{x}_{i,k}^{(t)}\|^2) + \alpha^2(\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i^2\mathbf{x}_{i,k}^{(t)} + \alpha^2((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i\mathbf{x}_{i,k}^{(t)})^2(\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) \\
&\leq 3w_{ii}^2 - 6\sqrt{3}(k-1)w_{ii}\alpha\lambda_1\|\mathbf{x}_{i,k}^{(t)}\| + 9\alpha^2\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2 \\
&\Leftrightarrow 2\alpha w_{ii}(\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i\mathbf{x}_{i,k}^{(t)}(1 - \|\mathbf{x}_{i,k}^{(t)}\|^2) + \alpha^2(\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i^2\mathbf{x}_{i,k}^{(t)} + \alpha^2((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i\mathbf{x}_{i,k}^{(t)})^2(\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) \\
&\leq -6\sqrt{3}(k-1)w_{ii}\alpha\lambda_1\|\mathbf{x}_{i,k}^{(t)}\| + 9\alpha^2\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2 \\
&\Leftrightarrow \alpha^2(\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i^2\mathbf{x}_{i,k}^{(t)} + \alpha^2((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i\mathbf{x}_{i,k}^{(t)})^2(\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) - 9\alpha^2\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2 \\
&\leq 2\alpha w_{ii}(\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i\mathbf{x}_{i,k}^{(t)}(\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) - 6\sqrt{3}(k-1)w_{ii}\alpha\lambda_1\|\mathbf{x}_{i,k}^{(t)}\| \\
&\Leftrightarrow \alpha \leq \frac{2w_{ii}(\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i\mathbf{x}_{i,k}^{(t)}(\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) - 6\sqrt{3}(k-1)w_{ii}\alpha\lambda_1\|\mathbf{x}_{i,k}^{(t)}\|}{(\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i^2\mathbf{x}_{i,k}^{(t)} + ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i\mathbf{x}_{i,k}^{(t)})^2(\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) - 9\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2}. \tag{3.46}
\end{aligned}$$

We now find the lower bound of the right-hand side of (3.46). Note that

$$\begin{aligned}
& (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i^2 \mathbf{x}_{i,k}^{(t)} + ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) - 9\lambda_1^2 (k-1)^2 \|\mathbf{x}_{i,k}^{(t)}\|^2 \\
& \stackrel{\zeta_1}{\leq} \lambda_1 (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} + ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) - 9\lambda_1^2 (k-1)^2 \|\mathbf{x}_{i,k}^{(t)}\|^2, \\
& \leq \lambda_1 (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} + \lambda_1 (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \|\mathbf{x}_{i,k}^{(t)}\|^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) - 9\lambda_1^2 (k-1)^2 \|\mathbf{x}_{i,k}^{(t)}\|^2 \\
& = \lambda_1 (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1)^2 - 9\lambda_1^2 (k-1)^2 \|\mathbf{x}_{i,k}^{(t)}\|^2 \\
& \leq \lambda_1 (k-1) (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1)^2 - 9\lambda_1^2 (k-1)^2 \|\mathbf{x}_{i,k}^{(t)}\|^2 \\
& < \stackrel{\zeta_2}{\leq} \lambda_1 (k-1) (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) \left((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) - 9\lambda_1 (k-1) \right),
\end{aligned}$$

where ζ_1 is true since $(\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i^2 \mathbf{x}_{i,k}^{(t)} \leq \lambda_1 (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)}$ and ζ_2 is true since $\frac{\|\mathbf{x}_{i,k}^{(t)}\|^2}{\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1} > 1$. Also,

$$\begin{aligned}
& 2w_{ii} (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) - 6\sqrt{3}(k-1)w_{ii}\lambda_1 \|\mathbf{x}_{i,k}^{(t)}\| \\
& \geq 2w_{ii} (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) - 18(k-1)w_{ii}\lambda_1 = 2w_{ii} ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) - 9(k-1)\lambda_1).
\end{aligned}$$

Thus, we have that the right hand side of (3.46) exceeds

$$\frac{2w_{ii} ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) - 9(k-1)\lambda_1)}{\lambda_1 (k-1) (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) \left((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) - 9\lambda_1 (k-1) \right)} = \frac{2w_{ii}}{\lambda_1 (k-1)} > \frac{w_{ii}}{3\lambda_1 (2K-1)}.$$

This proves if $\alpha \leq \min\{\frac{w_{ii}}{3\lambda_1 (2K-1)}, \frac{0.225}{3\lambda_1 (2K-1)}\}$ then $\|\mathbf{x}_{i,k}^{(t+1)}\| \leq \sqrt{3}$. \square

Lemma 7. *The norm of Sanger's direction $\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)})$ is bounded as*

$$\|\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)})\|^2 \leq 3\lambda_{i,1}^2 (3k-2)(3k+1), \forall k = 1, \dots, K. \quad (3.47)$$

Proof. We know

$$\begin{aligned}
\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)}) &= \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)} - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \\
&= (\mathbf{I} - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T) \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)} \\
&= \tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)}
\end{aligned}$$

Thus,

$$\begin{aligned}
\|\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)})\|^2 &= \|\tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)}\|^2 \\
&= (\mathbf{x}_{i,k}^{(t)})^T (\tilde{\mathbf{C}}_i^{(t)})^T \tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)} + ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) + \\
&\quad (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \sum_{p=1}^{k-1} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)}.
\end{aligned}$$

Next, notice that $\|\tilde{\mathbf{C}}_i^{(t)}\| = \|\mathbf{C}_i - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i\|$. Thus,

$$\|\tilde{\mathbf{C}}_i^{(t)}\| \leq \|\mathbf{C}_i\| + \sum_{p=1}^{k-1} \|\mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T\| \|\mathbf{C}_i\| \leq \lambda_{i,1} + \sum_{p=1}^{k-1} 3\lambda_{i,1} = \lambda_{i,1} + 3(k-1)\lambda_{i,1} = \lambda_{i,1}(3k-2).$$

Therefore, we get

$$\begin{aligned} (\mathbf{x}_{i,k}^{(t)})^T (\tilde{\mathbf{C}}_i^{(t)})^T \tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)} &\leq \lambda_{\max}((\tilde{\mathbf{C}}_i^{(t)})^T) (\mathbf{x}_{i,k}^{(t)})^T \tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)} \\ &= \|(\tilde{\mathbf{C}}_i^{(t)})\| (\mathbf{x}_{i,k}^{(t)})^T \tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)} \leq \lambda_{i,1}(3k-2) (\mathbf{x}_{i,k}^{(t)})^T \tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)}. \end{aligned}$$

Thus,

$$\begin{aligned} \|\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)})\|^2 &\leq \lambda_{i,1}(3k-2) (\mathbf{x}_{i,k}^{(t)})^T \tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)} + ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) + \\ &\quad (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \sum_{p=1}^{k-1} (\mathbf{x}_{i,k}^{(t)})^T \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \\ &\leq \lambda_{i,1}(3k-2) (\mathbf{x}_{i,k}^{(t)})^T \tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)} + ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) + \\ &\quad (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \sum_{p=1}^{k-1} \|\mathbf{x}_{i,k}^{(t)}\|^2 \|\mathbf{x}_{i,p}^{(t)}\|^2 \lambda_{i,1} \\ &\leq \lambda_{i,1}(3k-2) (\mathbf{x}_{i,k}^{(t)})^T (\mathbf{I} - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T) \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} + ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) \\ &\quad + (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \sum_{p=1}^{k-1} \|\mathbf{x}_{i,k}^{(t)}\|^2 \|\mathbf{x}_{i,p}^{(t)}\|^2 \lambda_{i,1} \\ &\leq \lambda_{i,1}(3k-2) (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} + \lambda_{i,1}(3k-2) \sum_{p=1}^{k-1} \|\mathbf{x}_{i,k}^{(t)}\|^2 \|\mathbf{x}_{i,p}^{(t)}\|^2 \lambda_{i,1} + \\ &\quad ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) + (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \sum_{p=1}^{k-1} \|\mathbf{x}_{i,k}^{(t)}\|^2 \|\mathbf{x}_{i,p}^{(t)}\|^2 \lambda_{i,1} \\ &\leq \lambda_{i,1}(3k-2) 3\lambda_{i,1} + \lambda_{i,1}(3k-2) \sum_{p=1}^{k-1} 9\lambda_{i,1} + 9\lambda_{i,1}^2 + \lambda_{i,1} 3 \sum_{p=1}^{k-1} 9\lambda_{i,1} \\ &= 3\lambda_{i,1}^2(3k-2)(3k+1). \end{aligned}$$

□

Lemma 8. *The deviation of an iterate at a node from the average is bounded from above as*

$$\|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \leq b_k(\beta^t + \frac{\alpha}{1-\beta}), \forall k = 1, \dots, K, \quad (3.48)$$

where β is the second largest magnitude of the eigenvalues of \mathbf{W} given as $\beta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_M(\mathbf{W})|\} < 1$ and $b_k > 0$ is some constant.

Proof. We stack the iterates $\mathbf{x}_{i,k}^{(t)}$ and $\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)})$ as

$$\mathbf{x}_k^{(t)} = \begin{bmatrix} \mathbf{x}_{1,k}^{(t)} \\ \mathbf{x}_{2,k}^{(t)} \\ \vdots \\ \mathbf{x}_{M,k}^{(t)} \end{bmatrix} \in \mathbb{R}^{Md} \quad \mathcal{H}(\mathbf{x}_k^{(t)}) = \begin{bmatrix} \mathcal{H}_1(\mathbf{x}_{1,k}^{(t)}) \\ \mathcal{H}_2(\mathbf{x}_{2,k}^{(t)}) \\ \vdots \\ \mathcal{H}_M(\mathbf{x}_{M,k}^{(t)}) \end{bmatrix} \in \mathbb{R}^{Md} \quad \mathbf{x}_{avg,k}^{(t)} = \begin{bmatrix} \bar{\mathbf{x}}_k^{(t)} \\ \bar{\mathbf{x}}_k^{(t)} \\ \vdots \\ \bar{\mathbf{x}}_k^{(t)} \end{bmatrix} \in \mathbb{R}^{Md}.$$

The next network-wide iterate (as a stacked vector) can then be written as $\mathbf{x}_k^{(t)} = (\mathbf{W} \otimes \mathbf{I})\mathbf{x}_k^{(t-1)} + \alpha\mathcal{H}(\mathbf{x}_k^{(t-1)})$, where \otimes denotes the Kronecker product. The t^{th} iterate can thus be written as

$$\mathbf{x}_k^{(t)} = (\mathbf{W}^t \otimes \mathbf{I})\mathbf{x}_k^{(0)} + \alpha \sum_{s=0}^{t-1} (\mathbf{W}^{t-1-s} \otimes \mathbf{I})\mathcal{H}(\mathbf{x}_k^{(s)}).$$

Since $\mathbf{W} = [w_{ij}]$ is a symmetric and doubly stochastic mixing matrix, its largest eigenvalue is 1 corresponding to the eigenvector $\mathbf{1}_M$, a column vector of all 1's. It is also the left eigenvector of \mathbf{W} . That is, $\mathbf{W}\mathbf{1}_M = \mathbf{1}_M$ and $\mathbf{1}_M^T \mathbf{W} = \mathbf{1}_M^T$. Also, since the squared norm of Sanger's direction at every node is bounded, it is easy to see $\|\mathcal{H}(\mathbf{x}_k^{(t)})\|^2 = 3M\lambda_1^2(3k-2)(3k+1)$.

Now,

$$\begin{aligned}
\|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| &\leq \|\mathbf{x}_k^{(t)} - \mathbf{x}_{avg,k}^{(t)}\| = \|\mathbf{x}_k^{(t)} - \frac{1}{M}((\mathbf{1}_M \mathbf{1}_M^T) \otimes \mathbf{I})\mathbf{x}_k^{(t)}\| \\
&= \|(\mathbf{W}^t \otimes \mathbf{I})\mathbf{x}_k^{(0)} + \alpha \sum_{s=0}^{t-1} (\mathbf{W}^{t-1-s} \otimes \mathbf{I})\mathcal{H}(\mathbf{x}_k^{(s)}) - \frac{1}{M}((\mathbf{1}_M \mathbf{1}_M^T) \otimes \mathbf{I})(\mathbf{W}^t \otimes \mathbf{I})\mathbf{x}_k^{(0)} \\
&\quad + \alpha \sum_{s=0}^{t-1} (\mathbf{W}^{t-1-s} \otimes \mathbf{I})\mathcal{H}(\mathbf{x}_k^{(s)})\| \\
&= \|(\mathbf{W}^t \otimes \mathbf{I})\mathbf{x}_k^{(0)} + \alpha \sum_{s=0}^{t-1} (\mathbf{W}^{t-1-s} \otimes \mathbf{I})\mathcal{H}(\mathbf{x}_k^{(s)}) - \frac{1}{M}((\mathbf{1}_M \mathbf{1}_M^T) \otimes \mathbf{I})\mathbf{x}_k^{(0)} \\
&\quad - \alpha \sum_{s=0}^{t-1} (\frac{1}{M}((\mathbf{1}_M \mathbf{1}_M^T) \otimes \mathbf{I})\mathcal{H}(\mathbf{x}_k^{(s)}))\| \\
&= \|((\mathbf{W}^t - \frac{1}{M}(\mathbf{1}_M \mathbf{1}_M^T)) \otimes \mathbf{I})\mathbf{x}_k^{(0)} + \alpha \sum_{s=0}^{t-1} ((\mathbf{W}^{t-1-s} - \frac{1}{M}(\mathbf{1}_M \mathbf{1}_M^T)) \otimes \mathbf{I})\mathcal{H}(\mathbf{x}_k^{(s)})\| \\
&\leq \|((\mathbf{W}^t - \frac{1}{M}(\mathbf{1}_M \mathbf{1}_M^T)) \otimes \mathbf{I})\mathbf{x}_k^{(0)}\| + \alpha \sum_{s=0}^{t-1} \|(\mathbf{W}^{t-1-s} - \frac{1}{M}(\mathbf{1}_M \mathbf{1}_M^T)) \otimes \mathbf{I}\| \|\mathcal{H}(\mathbf{x}_k^{(s)})\| \\
&\leq \|((\mathbf{W}^t - \frac{1}{M}(\mathbf{1}_M \mathbf{1}_M^T)) \otimes \mathbf{I})\mathbf{x}_k^{(0)}\| + \alpha \sum_{s=0}^{t-1} \|(\mathbf{W}^{t-1-s} - \frac{1}{M}(\mathbf{1}_M \mathbf{1}_M^T)) \otimes \mathbf{I}\| \|\mathcal{H}(\mathbf{x}_k^{(s)})\| \\
&= \beta^t \|\mathbf{x}_k^{(0)}\| + \alpha \sum_{s=0}^{t-1} \beta^{t-1-s} \|\mathcal{H}(\mathbf{x}_k^{(s)})\| \leq \beta^t \sqrt{3M} + \alpha \sqrt{3M\lambda_1^2(3k-2)(3k+1)} \sum_{s=0}^{t-1} \beta^{t-1-s} \\
&\leq \beta^t \sqrt{3M} + \frac{\alpha \sqrt{3M\lambda_1^2(3k-2)(3k+1)}}{1-\beta} \\
&\leq \sqrt{3M}\lambda_1 \sqrt{(3k-2)(3k+1)} \left(\beta^t + \frac{\alpha}{1-\beta} \right) \\
&= b_k \left(\beta^t + \frac{\alpha}{1-\beta} \right), \quad \text{where } b_k = \lambda_1 \sqrt{3M} \sqrt{(3k-2)(3k+1)}.
\end{aligned}$$

□

Lemma 9. Suppose $\|\mathbf{x}_{i,k}^{(t)}\|^2 \leq 3$ and $\|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \leq b_k(\beta^t + \frac{\alpha}{1-\beta})$, then the following is true $\forall k = 1, \dots, K$:

$$\|\mathbf{h}_k^{(t)}\| \leq 3(k+2)\lambda_1 b_k (\beta^t + \frac{\alpha}{1-\beta}). \quad (3.49)$$

Proof. We have

$$\begin{aligned}
&\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)}) - \mathcal{H}_i(\bar{\mathbf{x}}_k^{(t)}) \\
&= \mathbf{C}_i(\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}) - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)} + (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)} - \sum_{p=1}^{k-1} (\mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - \mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)}) \\
&= (\mathbf{C}_i - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{I})(\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}) - ((\mathbf{x}_{i,k}^{(t)} + \bar{\mathbf{x}}_k^{(t)})^T \mathbf{C}_i (\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)})) \bar{\mathbf{x}}_k^{(t)} - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i (\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)})
\end{aligned}$$

$$\begin{aligned}
\|\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)}) - \mathcal{H}_i(\bar{\mathbf{x}}_k^{(t)})\| &\leq \|\mathbf{C}_i - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{I}\| \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| + |(\mathbf{x}_{i,k}^{(t)} + \bar{\mathbf{x}}_k^{(t)})^T \mathbf{C}_i (\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)})| \|\bar{\mathbf{x}}_k^{(t)}\| \\
&\quad + \sum_{p=1}^{k-1} \|\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i (\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)})\| \\
&\leq \|\mathbf{C}_i - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{I}\| \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| + \|\mathbf{x}_{i,k}^{(t)} + \bar{\mathbf{x}}_k^{(t)}\| \|\mathbf{C}_i\| \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \|\bar{\mathbf{x}}_k^{(t)}\| \\
&\quad + \sum_{p=1}^{k-1} \|\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i\| \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \\
&\leq \|\mathbf{C}_i - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{I}\| \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| + \|\bar{\mathbf{x}}_k^{(t)}\| (\|\mathbf{x}_{i,k}^{(t)}\| + \|\bar{\mathbf{x}}_k^{(t)}\|) \|\mathbf{C}_i\| \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \\
&\quad + \sum_{p=1}^{k-1} \|\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T\| \|\mathbf{C}_i\| \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \\
&\leq 3\lambda_1 \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| + \sqrt{3}(2\sqrt{3})\lambda_1 \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| + \sum_{p=1}^{k-1} 3\lambda_1 \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \\
&= 3(k+2)\lambda_1 \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \leq 3(k+2)\lambda_1 b_k (\beta^t + \frac{\alpha}{1-\beta}).
\end{aligned}$$

Thus,

$$\|\mathbf{h}_k^{(t)}\| \leq \frac{1}{M} \sum_{i=1}^M \|\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)}) - \mathcal{H}_i(\bar{\mathbf{x}}_k^{(t)})\| \leq 3(k+2)\lambda_1 b_k (\beta^t + \frac{\alpha}{1-\beta}). \quad (3.50)$$

□

3.6 Experimental Results

In this section, we provide results that demonstrate the efficacy of the proposed DSA algorithm. The need for collaboration between the nodes of a network is a vital part of any distributed algorithm, as already pointed out in Section 3.2. We first verify that necessity along with the effect of step size on DSA by performing some experiments. In these experiments, the weight matrix \mathbf{W} that conforms to the underlying graph topology is generated using the Metropolis constant edge-weight approach [69]. The performance of DSA in comparison to some baseline methods is also evaluated in additional experiments. We provide experimental results for DSA on synthetic and real data and compare the results with centralized generalized Hebbian algorithm (GHA) [13], centralized orthogonal iteration (OI) [19], distributed projected gradient descent (DPGD) and sequential distributed power method (SeqDistPM). For both the centralized methods, all the data is assumed to be at a single location with the difference being that GHA uses the Hebbian update whereas OI uses the well-known orthogonal iterations to estimate the top K eigenvectors of the covariance matrix \mathbf{C} . DPGD involves two significant steps per iteration. The first is a distributed gradient descent step at every node i given by

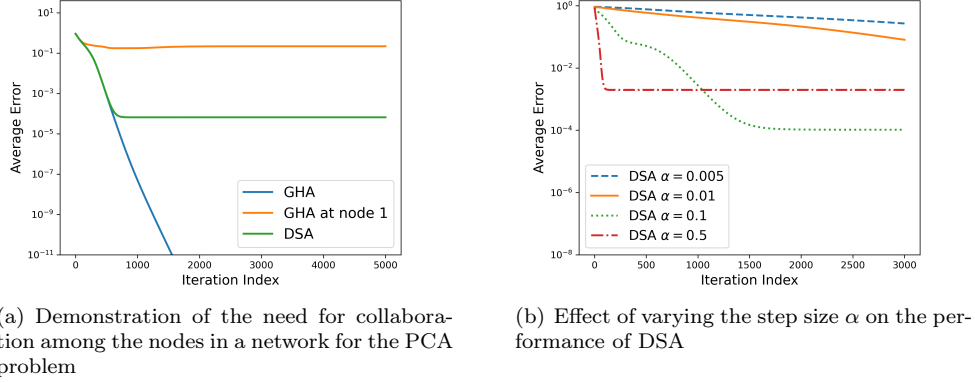


Figure 3.1: The role of collaboration in the distributed PCA problem and the effect of changing the step size on the performance of DSA. The distributed setup corresponds to an Erdos–Renyi graph ($p = 0.5$) with $M = 10$ nodes, while the dimension of data is $d = 10$ and the number of estimated eigenvectors is $K = 3$.

$\sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{X}_j + \alpha \nabla f_i(\mathbf{X}_i)$ as in [63] using trace maximization $f_i(\mathbf{X}_i) = \max \text{Trace}(\mathbf{X}_i^T \mathbf{C}_i \mathbf{X}_i)$ as the objective. This is followed by a projection step to ensure the orthogonality constraint $\mathbf{X}_i^T \mathbf{X}_i = \mathbf{I}$. The orthogonalization is accomplished using QR decomposition, an approach that ensures projection onto the Stiefel manifold [70] and whose computational complexity is $\mathcal{O}(K^2 d)$, at each node in each iteration. In contrast, SeqDistPM involves implementing the distributed power method [49, 50] K times, estimating one eigenvector at a time and subtracting its impact on the covariance matrix for the estimation of subsequent eigenvectors. Note that SeqDistPM requires a finite T_c number of consensus iterations per iteration of the power method. Assuming the cost of communicating one $\mathbb{R}^{d \times K}$ matrix across the network from nodes to their neighbors to be one unit, the communication cost of SeqDistPM is T_c/K per iteration of the power method. The error metric used for comparison and reporting of the results is the average of the angles between the estimated and true eigenvectors, i.e., if $\mathbf{x}_{i,k}$ is the estimate of the k^{th} eigenvector at i^{th} node and \mathbf{q}_k is the true k^{th} eigenvector then the average error across all nodes is calculated as follows:

$$E = \frac{1}{MK} \sum_{i=1}^M \sum_{k=1}^K \left(1 - \left(\frac{\mathbf{x}_{i,k}^T \mathbf{q}_k}{\|\mathbf{x}_{i,k}\|} \right)^2 \right). \quad (3.51)$$

3.6.1 Synthetic Data

We first show results that emphasize on the need for collaboration among the nodes. To that end, we generate $N = 10,000$ independent and identically distributed (i.i.d.) samples drawn from a multivariate Gaussian distribution with an eigengap $\Delta_K = \frac{\lambda_{K+1}}{\lambda_K} = 0.8$ and dimension $d = 10$. These samples are distributed equally among the $M = 10$ nodes of an Erdos–Renyi

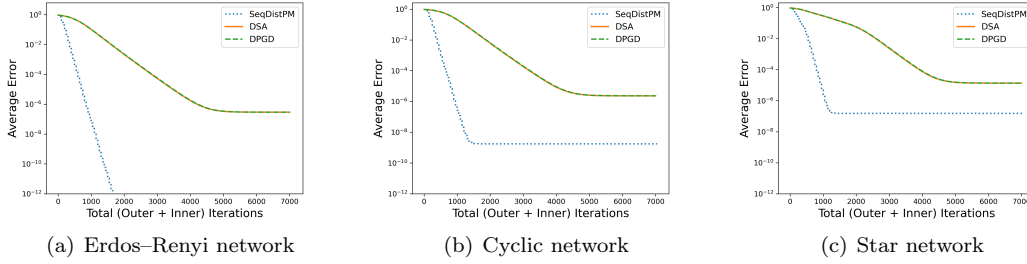


Figure 3.2: Comparison between the performances of DSA, DPGD and SeqDistPM for $K = 1$ and $\Delta_K = 0.8$ in terms of communications efficiency, i.e., decrease in average estimation error as a function of the number of data units communicated throughout the network.

network (with connectivity probability $p = 0.5$), implying that each node has 1,000 samples. The number of eigenvectors estimated is $K = 3$ and a constant step size of $\alpha = 0.1$ is used for this experiment. Figure 3.1(a) shows the effect of using the GHA at a node without collaboration with other nodes versus DSA, which in simple terms embodies GHA + collaboration in the network. The blue line indicating GHA in the figure is the result of using all the data in a centralized manner. It is clear that the lack of any communication between nodes increases the error in estimation of the eigenvectors by a significant factor. In Figure 3.1(b), we use the same setup and parameters to show the effect of different step sizes on our proposed DSA algorithm. It is evident that if the step size is too low, the convergence becomes significantly slow, while if its high, the final error is larger. Hence, careful choice of the step size is required for DSA, as characterized by its convergence analysis.

Next, we compare DSA with the distributed methods of DPGD and SeqDistPM to demonstrate its communication efficiency. For that purpose, we generate synthetic data with different eigengaps $\Delta_K \in \{0.6, 0.8\}$. We simulate the distributed setup for Erdos-Renyi ($p = 0.5$), star and cycle graph topologies with $M = 10$ nodes. The data is generated so that each node has 1,000 i.i.d samples ($N_i = 1000$) drawn from a multivariate Gaussian distribution for $d = 20$, i.e., the total samples generated are 10,000. The dimension of the subspace to be estimated is taken to be $K \in \{1, 5\}$. We use $T_c = 50$ as the number of consensus iterations per power iteration for SeqDistPM throughout our experiments. The results reported are an average of 10 Monte-Carlo trials. Figure 3.2 shows the performance of different algorithms for the estimation of the most dominant eigenvector for different network topologies. It is clear that for $K = 1$ SeqDistPM outperforms both DSA and DPGD in terms of communications efficiency because it is basically distributed power method, which is shown in [49, 50] to have good performance for $K = 1$. Even though DSA and DPGD have the same performance in terms of communications cost, it is

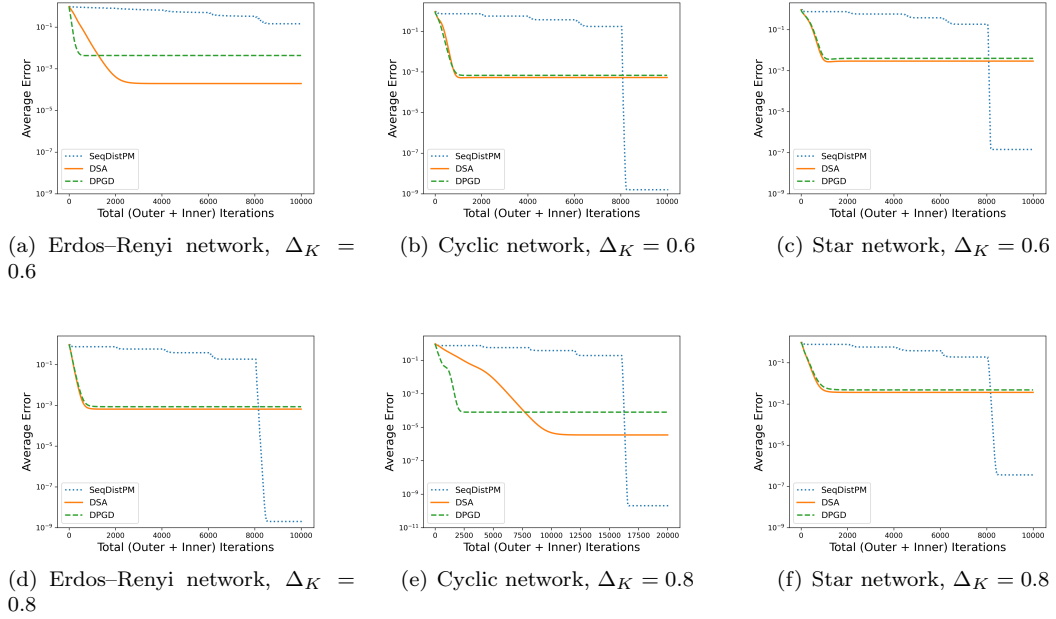


Figure 3.3: Comparison between DSA, DPGD, and SeqDistPM for $K = 5$ in terms of communications efficiency.

important to remember that DPGD requires an additional QR normalization step per communications round. Next, Figure 3.3 shows a comparison between the three algorithms when the top-5 eigenvectors are estimated i.e., $K = 5$. It is clear that while estimating higher-order eigenvectors, DSA slightly outperforms DPGD without performing explicit QR normalization and it also has much better communications efficiency than SeqDistPM. The error for SeqDistPM is significantly high in the beginning because of the sequential estimation, which means that when the first (higher-order) eigenvector(s) is (are) being estimated, the lower-order estimates are still at their initial values and hence those contribute significant error even when the first or higher order terms have low error. After a sufficiently large number of communications rounds, SeqDistPM eventually does reach a lower final error compared to DSA. But this comes at the expense of slower convergence as a function of communications costs. It should also be noted that SeqDistPM lacks a formal convergence analysis and has two time scales that need to be adjusted as both contribute to the final error. Finally, the benefits of DSA over DPGD are twofold. First, DSA reaches similar or better error floor without explicit QR normalization, thus saving $\mathcal{O}(K^2d)$ computations per iteration; and second, the convergence guarantees for gradient descent-based algorithms for non-convex problems like the PCA have limitations. The guarantees usually exist for convergence to a stationary solution with a sub-linear rate.

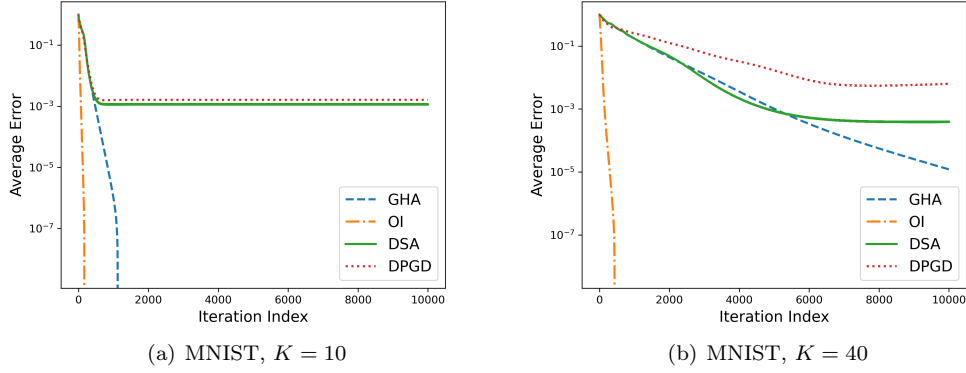


Figure 3.4: Comparison between DSA, OI, GHA, and DPGD for MNIST dataset as a function of the number of algorithmic iterations.

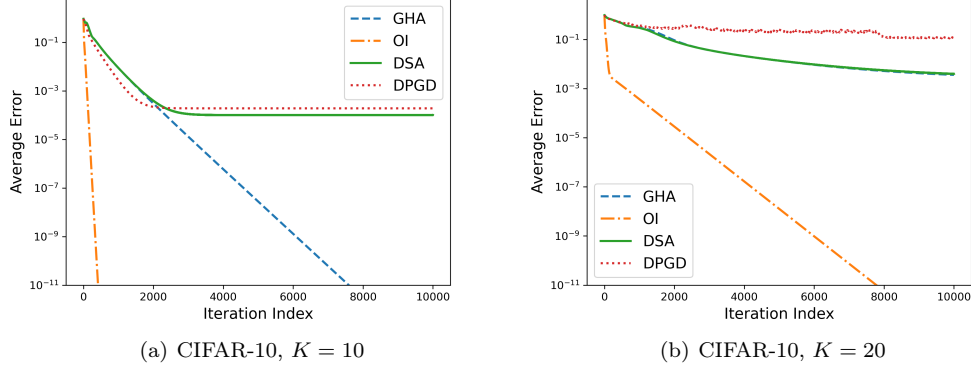


Figure 3.5: Comparison between DSA, OI, GHA, and DPGD for CIFAR-10 dataset as a function of the number of algorithmic iterations.

3.6.2 Real-World Data

Along with the synthetic data experiments, we provide some experiments with real-world datasets of MNIST [71] and CIFAR-10 [72]. For the distributed setup in this case, we use an Erdos–Renyi graph with $M = 20$ nodes and $p = 0.5$. Both the datasets have 60,000 samples, thereby making the number of samples per node to be $N_i = 3000$. The data dimension for MNIST is $d = 784$ and a constant step size of $\alpha = 0.1$ was used. The plots in Figure 3.4(a) and Figure 3.4(b) show the results for $K \in \{10, 40\}$ for MNIST. Similar plots are shown for CIFAR-10 in Figure 3.5(a) and Figure 3.5(b), where the dimension d for CIFAR-10 is 1024, the number of estimated eigenvectors $K \in \{10, 20\}$ and a constant step size of $\alpha = 0.7$ is used. For these real-world data sets, we exclude the comparison with SeqDistPM as it is evident this method requires much higher cost of communications for estimating larger number of eigenvectors.

3.7 Conclusion

In this chapter , we proposed and analyzed a new distributed Principal Component Analysis (PCA) algorithm that, as opposed to distributed subspace learning methods, facilitates both dimensionality reduction and data decorrelation in a distributed setup. Our main contribution in this regard was a detailed convergence analysis to prove that the proposed distributed method linearly converges to a neighborhood of the eigenvectors of the global covariance matrix. We also provided numerical results to demonstrate the communications efficiency and overall effectiveness of the proposed algorithm.

Chapter 4

FAST-PCA: A Fast and Exact Algorithm for Distributed PCA

In the previous chapter a distributed algorithm for PCA based on generalized Hebbian algorithm using a combine-and-adapt strategy called distributed Sanger's algorithm (DSA) [16] was developed and analyzed. But it only reaches to a neighborhood of the optimal solution for a fixed step size. The algorithm, however, does converge exactly in the case of decreasing step sizes but with a slower rate of convergence. To overcome this limitation, this chapter introduces a new method that uses a gradient-tracking idea to develop a novel algorithm for distributed PCA called *Fast and exAct diSTributed PCA* (FAST-PCA). Two variants of the algorithm are proposed in this chapter that converge linearly and exactly to the optimal solution.

4.1 Introduction

We focus on finding exact solutions for principal component analysis when data samples are distributed across a network. As evident from our previous solution, a simple combine-and-adapt strategy using a generalized Hebbian method can only reach a neighborhood of the true eigenvectors of the covariance matrix. To overcome the limitations of simple gradient descent-based distributed algorithms, new methods have been proposed recently that deploy a technique called "gradient-tracking" [65, 73, 74]. In this chapter, we use this gradient-tracking idea to develop a one-time scale algorithm called *Fast and exAct diSTributed PCA* (FAST-PCA) that *linearly and exactly* converges to the eigenvectors of the covariance matrix, not just the subspace spanned by them. Although this strategy has mainly been used in distributed

optimization literature for convex and strongly convex problems, we showed using extensive analysis that even for the non-convex PCA problem, each node converges linearly and globally, i.e., starting from any random initial point inspite of being a one time-scale algorithm. We propose two versions of the FAST-PCA algorithm. One is based on the generalized Hebbian algorithm and can be viewed as an extension of our DSA algorithm. The other version is based on Krasulina’s method [21], which is a stochastic approximation method for estimation of the dominant eigenvector of covariance matrix in streaming data. Along with adapting Krasulina to the distributed case, we also generalize it for estimation of multiple leading eigenvectors. Although Hebbian and Krasulina’s method are similar, the question of which (if any) is better still remains open. This chapter answers this question in the context of distributed PCA.

4.1.1 Our Contributions

The main contributions of this chapter are 1) two versions of a novel algorithm for distributed PCA called *Fast and exAct diSTributed PCA* (FAST-PCA) based on a gradient-tracking technique, 2) theoretical guarantees that show that the estimates given by our method converge exactly and globally at a linear rate to the eigenvectors of the global covariance matrix, and 3) experimental results that further demonstrate the efficiency of our solution for both synthetic and real-world datasets.

Our primary focus in this chapter is to develop a solution for distributed PCA when the data samples are scattered across an arbitrarily connected network with no central node. While PCA is often reduced to dimension reduction, we focus on the dual goal of PCA that requires dimensionality reduction as well as feature decorrelation. To that end, we propose two versions algorithm based on a gradient-tracking approach called FAST-PCA. First of an version called FAST-PCA-O is based on the Hebbian (Oja’s) rule for the estimation of multiple eigenvectors in a distributed setting. The second version is based on the Krasulina’s method and is called FAST-PCA-K. Since the original Krasulina’s method only finds the dominant eigenvector, we also generalize it to the distributed setting for the estimation of top K eigenvectors. Our proposed FAST-PCA method is an iterative update algorithm and its main attributes are that it is fast since it lacks any explicit consensus loop and hence reduces the communication overhead, and it converges exactly to the true eigenvectors of the global covariance matrix at a linear rate. We provide detailed convergence analysis to support our claims as well as extensive numerical experiments where we compare our method to centralized orthogonal iteration (OI) as the centralized baseline, as well as distributed PCA algorithms of sequential distributed power method (SeqDistPM), DeEPCA and DSA. We provide the results for different network

topologies as well as eigengaps to further solidify our claims.

To the best of our knowledge, this is a first novel algorithm for distributed PCA which achieves fast and exact convergence to the true eigenvectors of the global covariance matrix at every node of an arbitrarily connected network.

4.2 Proposed Algorithm

Iterative methods like power method [19] have proven to be very effective solutions for PCA because of their effectiveness and ease of implementations. For estimating the top eigenvector ($K = 1$) of data covariance matrix in centralized setting for the streaming data case, two elegant, similar and widely studied algorithms were proposed by Oja [12] and Krasulina [21]. Let $\mathbf{y}_t, t = 1, 2, \dots$ be data sample drawn from a zero-mean distribution at time t . Then the update equations of these two methods in centralized setting go as follows:

$$\begin{aligned} \text{Oja: } \mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} + \alpha_t (\mathbf{C}_t \mathbf{x}^{(t)} - (\mathbf{x}^{(t)})^T \mathbf{C}_t \mathbf{x}^{(t)} \mathbf{x}^{(t)}) \\ \text{Krasulina: } \mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} + \alpha_t \left(\mathbf{C}_t \mathbf{x}^{(t)} - \frac{(\mathbf{x}^{(t)})^T \mathbf{C}_t \mathbf{x}^{(t)}}{\|\mathbf{x}^{(t)}\|^2} \mathbf{x}^{(t)} \right), \end{aligned}$$

where $\mathbf{C}_t = \mathbf{y}_t \mathbf{y}_t^T$ is the sample covariance matrix and α_t is the step size at time t . Both these methods were shown to converge to the dominant eigenvector of $\mathbf{\Sigma} = \mathbb{E} [\mathbf{C}_t]$ under the requirement that $\sum_t \alpha_t^2 \rightarrow 0$. From an autoencoder training point of view, the simple update based rules can be very easily implemented in neural networks where the \mathbf{x} 's denote the weights of the network. The resultant ‘‘encoding’’ or representation learned from such network will have uncorrelated features. Oja’s rule was generalized for the case of multiple eigenvector estimation ($K > 1$) by Sanger [13] using the generalized Hebbian rule as follows:

$$\mathbf{x}_k^{(t+1)} = \mathbf{x}_k^{(t)} + \alpha (\mathbf{C} \mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \mathbf{x}_p^{(t)} (\mathbf{x}_p^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}). \quad (4.1)$$

Krasulina’s method was also extended but only for the estimation of a higher dimensional ($K > 1$) subspace spanned by the eigenvectors of $\mathbf{\Sigma}$ [25]. Furthermore, since Matrix Krasulina [25] only estimates the dominant subspace, Krasulina’s method also needs to be generalized for the estimation of $K > 1$ dominant *eigenvectors*.

In the distributed setup considered in this paper, samples are not streaming but distributed across a connected network of M nodes, where node i has access to a local covariance matrix \mathbf{C}_i such that $\sum_{i=1}^M \mathbf{C}_i = \mathbf{C}$, the global covariance matrix. It is noteworthy that $\mathbb{E} [\mathbf{C}_t] = \mathbb{E} [\mathbf{C}_i] = \mathbf{\Sigma}$ and this similarity between streaming and distributed setting motivates the extrapolation of Oja’s and Krasulina’s method for the distributed setting. As mentioned before, the data is fixed

at each node and the goal is to find the top K eigenvectors of \mathbf{C} at each node i.e., have consensus in the network, and thus naive implementation of generalized Hebbian rule (GHA) or Krasulina's method would not accomplish our goal. Instead, we propose a gradient-tracking [73, 74] based solution called *Fast and exAct diSTributed PCA* (FAST-PCA) that converges exactly and at a linear rate to the top K eigenvectors of the global covariance matrix \mathbf{C} . We propose two variants of FAST-PCA based on the Hebbian (Oja's) and Krasulina's rule and call them FAST-PCA-O and FAST-PCA-K, respectively.

Let $\mathbf{x}_{i,k}^{(t)}$ be the estimate of the k^{th} , $k = 1, \dots, K$, eigenvector at node i after t iterations of the algorithm. Then we define the pseudo gradient $\mathbf{h}_i^O(\mathbf{x}_{i,k}^{(t)})$ motivated from the Oja's rule as follows:

$$\mathbf{h}_i^O(\mathbf{x}_{i,k}^{(t)}) = \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)} - \sum_{p=1}^{k-1} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,p}^{(t)}$$

Also, for the estimation of the dominant eigenvector at node i after the t^{th} iteration, we define a pseudo-gradient $\mathbf{h}_i^K(\mathbf{x}_{i,1}^{(t)})$ motivated from Krasulina's rule at node i as follows:

$$\mathbf{h}_i^K(\mathbf{x}_{i,1}^{(t)}) = \mathbf{C}_i \mathbf{x}_{i,1}^{(t)} - \frac{(\mathbf{x}_{i,1}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,1}^{(t)}}{\|\mathbf{x}_{i,1}^{(t)}\|^2} \mathbf{x}_{i,1}^{(t)}. \quad (4.2)$$

Additionally, for the estimation of k^{th} , $k = 2, \dots, K$, eigenvector we propose to generalize Krasulina's update rule along the lines of the generalized Hebbian algorithm [13] and combine Krasulina's method with Gram-Schmidt orthogonalization to define a general pseudo-gradient as:

$$\mathbf{h}_i^K(\mathbf{x}_{i,k}^{(t)}) = \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - \frac{(\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)}}{\|\mathbf{x}_{i,k}^{(t)}\|^2} \mathbf{x}_{i,k}^{(t)} - \sum_{p=1}^{k-1} \frac{(\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)}}{\|\mathbf{x}_{i,p}^{(t)}\|^2} \mathbf{x}_{i,p}^{(t)}. \quad (4.3)$$

Here, the term $\frac{(\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)}}{\|\mathbf{x}_{i,p}^{(t)}\|^2} \mathbf{x}_{i,p}^{(t)}$ is analogous to Gram-Schmidt orthogonalization and enforces the orthogonality of $\mathbf{x}_{i,k}^{(t)}$ to $\mathbf{x}_{i,p}^{(t)}$, $p = 1, \dots, k-1$.

Let $\mathbf{X}_i^{(t)} = [\mathbf{x}_{i,1}^{(t)}, \dots, \mathbf{x}_{i,K}^{(t)}] \in \mathbb{R}^{d \times K}$ be the estimate of the K eigenvectors of the global covariance matrix \mathbf{C} after t iterations. Along with $\mathbf{X}_i^{(t)}$, FAST-PCA also updates a second variable in every iteration that essentially tracks the average of the pseudo-gradients at the nodes. Let us define a *pseudo-gradient tracker* matrix $\mathbf{S}_i^{(t)} = [\mathbf{s}_{i,1}^{(t)}, \dots, \mathbf{s}_{i,K}^{(t)}] \in \mathbb{R}^{d \times K}$ that tracks the average of the pseudo-gradients at each node. These $\mathbf{S}_i^{(t)}$ are updated along with the eigenvector estimates $\mathbf{X}_i^{(t)}$ in each iteration of FAST-PCA. The entities $\mathbf{h}_i^O(\mathbf{X}_i^{(t)})$ and $\mathbf{h}_i^K(\mathbf{X}_i^{(t)})$ in the algorithm are the matrices of the psuedo-gradients, i.e., $\mathbf{h}_i^{O/K}(\mathbf{X}_i^{(t)}) = [\mathbf{h}_i^{O/K}(\mathbf{x}_{i,1}^{(t)}), \dots, \mathbf{h}_i^{O/K}(\mathbf{x}_{i,K}^{(t)})] \in \mathbb{R}^{d \times K}$. Both versions of the algorithm, namely FAST-PCA-O and FAST-PCA-K, involve same steps except the difference in the pseudo-gradients as explained earlier and are formally described

Algorithm 2: Fast and exAct diSTributed PCA (FAST-PCA)

Input: $\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_M, \mathbf{W}, \alpha, K$
Initialize: $\forall i, \mathbf{X}_i^{(0)} \leftarrow \mathbf{X}_{\text{init}} : \mathbf{X}_{\text{init}} \in \mathbb{R}^{d \times K}, \mathbf{X}_{\text{init}}^T \mathbf{X}_{\text{init}} = \mathbf{I}; \mathbf{S}_i^{(0)} \leftarrow \mathbf{h}_i^{O/K}(\mathbf{X}_i^{(0)})$

 1: **for** $t = 0, 1, \dots$ **do**

 2: Communicate $\mathbf{X}_i^{(t)}$ from each node i to its neighbors

 3: Subspace estimate at node i : $\mathbf{X}_i^{(t+1)} \leftarrow \frac{1}{2} \mathbf{X}_i^{(t)} + \sum_{j \in \mathcal{N}_i} \frac{w_{ij}}{2} \mathbf{X}_j^{(t)} + \alpha \mathbf{S}_i^{(t)}$

 4: Psuedo-gradient estimate at node i :

$$\mathbf{S}_i^{(t+1)} \leftarrow \frac{1}{2} \mathbf{S}_i^{(t)} + \sum_{j \in \mathcal{N}_i} \frac{w_{ij}}{2} \mathbf{S}_j^{(t)} + \mathbf{h}_i^{O/K}(\mathbf{X}_i^{(t+1)}) - \mathbf{h}_i^{O/K}(\mathbf{X}_i^{(t)})$$

 5: **end for**
Return: $\mathbf{X}_i^{(t+1)}, i = 1, 2, \dots, M$

in Algorithm 2. The weight matrix $\mathbf{W} = [w_{ij}]$ is a doubly stochastic matrix that conforms to the underlying graph topology [69], i.e., $w_{ij} \neq 0$ if $(i, j) \in \mathcal{E}$ or $i = j$ and 0 otherwise. A necessary assumption for convergence of the algorithm here is the graph connectivity, which ensures that the magnitude of the second largest eigenvalue of \mathbf{W} is strictly less than 1. The gradient-tracking based solutions are recently being very popular in distributed optimization literature because of their fast and exact convergence guarantees. Our main challenge here was providing theoretical convergence guarantees inspite of the non-convex nature of the problem. In the next section, we provide convergence results of both versions of the proposed algorithm FAST-PCA.

4.3 Convergence Analysis of Auxiliary Results

This section entails detailed analysis for some auxiliary results that aid the proof of convergence of both versions of our proposed FAST-PCA algorithm. In the first subsection, we restate the auxiliary result from previous chapter which gives the proof of convergence of a *modified GHA* (Subsection 3.4.1). This auxiliary result will be a key component in proving the convergence of FAST-PCA-O. The second subsection provides detailed analysis of another auxiliary result for a modified version of *modified Krasulina* that will aid the proof of convergence of FAST-PCA-K.

4.3.1 Convergence Analysis of a Modified GHA

Let $\mathbf{C} \in \mathbb{R}^{d \times d}$ be a covariance matrix whose eigenvectors are $\mathbf{q}_l, l = 1, \dots, d$, with corresponding eigenvalues λ_l . We define a general update rule for the “estimation” of k^{th} eigenvector, termed as *modified GHA* as the following:

$$\mathbf{x}_{g,k}^{(t+1)} = \mathbf{x}_{g,k}^{(t)} + \alpha (\mathbf{C} \mathbf{x}_{g,k}^{(t)} - (\mathbf{x}_{g,k}^{(t)})^T \mathbf{C} \mathbf{x}_{g,k}^{(t)} \mathbf{x}_{g,k}^{(t)} - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \mathbf{x}_{g,k}^{(t)}). \quad (4.4)$$

where α is a constant step size. We saw in Subsection 3.4.1, the update equation (4.4) converges to the eigenvector $\pm \mathbf{q}_k$ of \mathbf{C} corresponding to the k^{th} largest eigenvalue.

Note that this modified GHA not an algorithm in the true sense of the term as it cannot be implemented because of its dependence on the true eigenvectors \mathbf{q}_p . The sole purpose of this update equation is to help in our ultimate goal of providing convergence guarantee for the FAST-PCA-O algorithm.

4.3.2 Convergence Analysis of a Modified Krasulina

Let $\mathbf{C} \in \mathbb{R}^{d \times d}$ be a covariance matrix whose eigenvectors are $\mathbf{q}_l, l = 1, \dots, d$, with corresponding eigenvalues λ_l . With an aim to estimate the first K eigenvectors of \mathbf{C} , we define a general update rule, called *modified Krasulina*, of the following form:

$$\mathbf{x}_{g,k}^{(t+1)} = \mathbf{x}_{g,k}^{(t)} + \alpha \left(\mathbf{C} \mathbf{x}_{g,k}^{(t)} - \frac{(\mathbf{x}_{g,k}^{(t)})^T \mathbf{C} \mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|^2} \mathbf{x}_{g,k}^{(t)} - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \mathbf{x}_{g,k}^{(t)} \right) \quad (4.5)$$

$$= \mathbf{x}_{g,k}^{(t)} + \alpha \left(\mathbf{C} \mathbf{x}_{g,k}^{(t)} - \frac{(\mathbf{x}_{g,k}^{(t)})^T \mathbf{C} \mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|^2} \mathbf{x}_{g,k}^{(t)} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T \mathbf{x}_{g,k}^{(t)} \right), \quad (4.6)$$

where α is a constant step size. Again, note that this modified Krasulina is not an algorithm in the true sense of the term as it cannot be implemented because of its dependence on the true eigenvectors \mathbf{q}_p . The sole purpose of this update equation is to help in our ultimate goal of providing convergence guarantee for the FAST-PCA-K algorithm.

Since $\mathbf{q}_l, l = 1, \dots, d$, are the eigenvectors of a real symmetric matrix, they form a basis for d -dimensional space and can be used for expansion of any vector $\mathbf{x} \in \mathbb{R}^d$. Let

$$\tilde{\mathbf{x}}_{g,k}^{(t)} = \frac{\mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|} = \sum_{l=1}^d z_{k,l}^{(t)} \mathbf{q}_l, \quad (4.7)$$

where $z_{k,l}^{(t)}$ is the coefficient corresponding to the eigenvector \mathbf{q}_l in the expansion of $\tilde{\mathbf{x}}_{g,k}^{(t)}$. The update equation (3.12) can be re-written as:

$$\frac{\mathbf{x}_{g,k}^{(t+1)}}{\|\mathbf{x}_{g,k}^{(t+1)}\|} = \left(\frac{\mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|} + \alpha \left(\mathbf{C} \frac{\mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|} - \frac{(\mathbf{x}_{g,k}^{(t)})^T \mathbf{C} \mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|^2} \frac{\mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T \frac{\mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|} \right) \right) \frac{\|\mathbf{x}_{g,k}^{(t)}\|}{\|\mathbf{x}_{g,k}^{(t+1)}\|} \quad (4.8)$$

$$\tilde{\mathbf{x}}_{g,k}^{(t+1)} = \left(\tilde{\mathbf{x}}_{g,k}^{(t)} + \alpha \left(\mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)} - \frac{(\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)}}{\|\tilde{\mathbf{x}}_{g,k}^{(t)}\|^2} \tilde{\mathbf{x}}_{g,k}^{(t)} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T \tilde{\mathbf{x}}_{g,k}^{(t)} \right) \right) a_k^{(t)}, \quad (4.9)$$

where $a_k^{(t)} = \frac{\|\mathbf{x}_{g,k}^{(t)}\|}{\|\mathbf{x}_{g,k}^{(t+1)}\|}$. Multiplying both sides of (4.9) by \mathbf{q}_l^T and using the fact that $\mathbf{q}_l^T \mathbf{q}_{l'} = 0$ for $l \neq l'$, we get

$$z_{k,l}^{(t+1)} = a_k^{(t)} (z_{k,l}^{(t)} + \alpha (\mathbf{q}_l^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)} - \mathbf{q}_l^T (\sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T \tilde{\mathbf{x}}_{g,k}^{(t)}) - (\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)} z_{k,l}^{(t)})).$$

This gives

$$z_{k,l}^{(t+1)} = a_k^{(t)} (z_{k,l}^{(t)} - \alpha (\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)} z_{k,l}^{(t)}), \quad \text{for } l = 1, \dots, k-1, \quad (4.10)$$

$$\text{and } z_{k,l}^{(t+1)} = a_k^{(t)} (z_{k,l}^{(t)} + \alpha (\lambda_l - (\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)}) z_{k,l}^{(t)}), \quad \text{for } l = k, \dots, d. \quad (4.11)$$

In the following theorem, we show that $\mathbf{x}_{g,k}^{(t)}$ converges to a multiple of the true eigenvector \mathbf{q}_k by proving convergence of the coefficients $z_{k,l}^{(t)}$ for $l = 1, \dots, d$.

Theorem 4. Suppose \mathbf{C} has K distinct eigenvalues, i.e., $\lambda_1 > \lambda_2 > \dots > \lambda_K > \lambda_{K+1} \geq \dots \geq \lambda_d \geq 0$ and $\alpha < \frac{1}{\lambda_1}$, $\mathbf{q}_k^T \mathbf{x}_{g,k}^{(0)} \neq 0$, and $\|\mathbf{x}_{g,k}^{(0)}\| = 1$ for all $k = 1, \dots, K$. Then the update equation for $\mathbf{x}_{g,k}^{(t)}$ given by (3.12) converges at a linear rate to a multiple of the eigenvector $\pm \mathbf{q}_k$ corresponding to the k^{th} largest eigenvalue λ_k of the covariance matrix \mathbf{C} for $k = 1, \dots, K$.

Proof. We prove the linear convergence of $\mathbf{x}_{g,k}^{(t)}$ to a multiple of \mathbf{q}_k by proving that $\tilde{\mathbf{x}}_{g,k}^{(t)}$ converges to \mathbf{q}_k at a linear rate. The convergence of $\tilde{\mathbf{x}}_{g,k}^{(t)}$ to \mathbf{q}_k requires convergence of the lower-order coefficients $z_{k,1}^{(t)}, \dots, z_{k,k-1}^{(t)}$ and the higher-order coefficients $z_{k,k+1}^{(t)}, \dots, z_{k,d}^{(t)}$ to 0 and convergence of $z_{k,k}^{(t)}$ to ± 1 . To this end, Lemma 10 in the appendix proves linear convergence of the lower-order coefficients $z_{k,1}^{(t)}, \dots, z_{k,k-1}^{(t)}$ to 0 by showing $\sum_{l=1}^{k-1} (z_{k,l}^{(t)})^2 \leq a_1 \gamma_k^t$ for some constants $a_1 > 0$, $\gamma_k = \left(\frac{1}{1+\alpha\lambda_k}\right)^2 < 1$. Furthermore, Lemma 11 in the appendix shows that $\sum_{l=k+1}^d (z_{k,l}^{(t)})^2 \leq a_2 \delta_k^t$, where $a_2 > 0$ and $\delta_k = \left(\frac{1+\alpha\lambda_{k+1}}{1+\alpha\lambda_k}\right)^2 < 1$, thereby proving linear convergence of the higher-order coefficients to 0. The formal statements and proofs of Lemma 10 and Lemma 11 are given in the next subsection. Finally, since $\|\tilde{\mathbf{x}}_{g,k}^{(t)}\|^2 = 1$, we have

$$\begin{aligned} \sum_{l=1}^d (z_{k,l}^{(t)})^2 &= 1 \\ \text{or, } 1 - (z_{k,k}^{(t)})^2 &= \sum_{l=1}^{k-1} (z_{k,l}^{(t)})^2 + \sum_{l=k+1}^d (z_{k,l}^{(t)})^2 \\ &\leq a_1 \gamma_k^t + a_2 \delta_k^t \\ &< a_3 \delta_k^t, \quad \text{where } a_3 = \max\{a_1, a_2\} \quad \text{and} \quad \delta_k = \max\{\gamma_k, \delta_k\}. \end{aligned}$$

This shows $(z_{k,k}^{(t)})^2$ converges to 1 and $(z_{k,l}^{(t)})^2, l \neq k$, converges to 0 at a linear rate of $\mathcal{O}(\delta_k^t)$ where $\delta_k = \left(\frac{1+\alpha\lambda_{k+1}}{1+\alpha\lambda_k}\right)^2$. Thus, $\tilde{\mathbf{x}}_{g,k}^{(t)} \rightarrow \pm \mathbf{q}_k$ at a linear rate of $\left(\frac{1+\alpha\lambda_{k+1}}{1+\alpha\lambda_k}\right)^t$ and $(\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)} \rightarrow \lambda_k$.

We also know from (3.12) that

$$\begin{aligned} \mathbf{x}_{g,k}^{(t+1)} &= \mathbf{x}_{g,k}^{(t)} + \alpha \left(\mathbf{C} \mathbf{x}_{g,k}^{(t)} - \frac{(\mathbf{x}_{g,k}^{(t)})^T \mathbf{C} \mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|^2} \mathbf{x}_{g,k}^{(t)} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T \mathbf{x}_{g,k}^{(t)} \right) \\ &= \mathbf{x}_{g,k}^{(t)} + \alpha \left(\left(\mathbf{C} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T \right) \mathbf{x}_{g,k}^{(t)} - \frac{(\mathbf{x}_{g,k}^{(t)})^T \mathbf{C} \mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|^2} \mathbf{x}_{g,k}^{(t)} \right) \end{aligned}$$

Thus,

$$\begin{aligned}
\|\mathbf{x}_{g,k}^{(t+1)}\|^2 &= \|\mathbf{x}_{g,k}^{(t)}\|^2 + \alpha^2 \left\| \left(\mathbf{C} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T \right) \mathbf{x}_{g,k}^{(t)} - \frac{(\mathbf{x}_{g,k}^{(t)})^T \mathbf{C} \mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|^2} \mathbf{x}_{g,k}^{(t)} \right\|^2 \\
&\quad - 2\alpha (\mathbf{x}_{g,k}^{(t)})^T \left(\left(\mathbf{C} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T \right) \mathbf{x}_{g,k}^{(t)} - \frac{(\mathbf{x}_{g,k}^{(t)})^T \mathbf{C} \mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|^2} \mathbf{x}_{g,k}^{(t)} \right) \\
&= \|\mathbf{x}_{g,k}^{(t)}\|^2 + \alpha^2 \left\| \left(\mathbf{C} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T \right) \mathbf{x}_{g,k}^{(t)} - \frac{(\mathbf{x}_{g,k}^{(t)})^T \mathbf{C} \mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|^2} \mathbf{x}_{g,k}^{(t)} \right\|^2 \\
&\quad - 2\alpha \left((\mathbf{x}_{g,k}^{(t)})^T \mathbf{C} \mathbf{x}_{g,k}^{(t)} - \sum_{p=1}^{k-1} (\mathbf{x}_{g,k}^{(t)})^T \lambda_p \mathbf{q}_p \mathbf{q}_p^T \mathbf{x}_{g,k}^{(t)} - \frac{(\mathbf{x}_{g,k}^{(t)})^T \mathbf{C} \mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|^2} (\mathbf{x}_{g,k}^{(t)})^T \mathbf{x}_{g,k}^{(t)} \right) \\
&= \|\mathbf{x}_{g,k}^{(t)}\|^2 + \alpha^2 \left\| \left(\mathbf{C} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T \right) \mathbf{x}_{g,k}^{(t)} - \frac{(\mathbf{x}_{g,k}^{(t)})^T \mathbf{C} \mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|^2} \mathbf{x}_{g,k}^{(t)} \right\|^2 + 2\alpha \sum_{p=1}^{k-1} \lambda_p (\mathbf{x}_{g,k}^{(t)})^T \mathbf{q}_p \mathbf{q}_p^T \mathbf{x}_{g,k}^{(t)} \\
&= \|\mathbf{x}_{g,k}^{(t)}\|^2 + \alpha^2 \left\| \left(\mathbf{C} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T \right) \tilde{\mathbf{x}}_{g,k}^{(t)} \|\mathbf{x}_{g,k}^{(t)}\| - \frac{(\mathbf{x}_{g,k}^{(t)})^T \mathbf{C} \mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|^2} \tilde{\mathbf{x}}_{g,k}^{(t)} \|\mathbf{x}_{g,k}^{(t)}\| \right\|^2 \\
&\quad + 2\alpha \|\mathbf{x}_{g,k}^{(t)}\|^2 \sum_{p=1}^{k-1} \lambda_p (\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{q}_p \mathbf{q}_p^T \tilde{\mathbf{x}}_{g,k}^{(t)} \quad (4.12)
\end{aligned}$$

As $\tilde{\mathbf{x}}_{g,k}^{(t)} \rightarrow \pm \mathbf{q}_k$ and $\frac{(\mathbf{x}_{g,k}^{(t)})^T \mathbf{C} \mathbf{x}_{g,k}^{(t)}}{\|\mathbf{x}_{g,k}^{(t)}\|^2} \rightarrow \lambda_k$, we have

$$\left(\mathbf{C} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T \right) \tilde{\mathbf{x}}_{g,k}^{(t)} \|\mathbf{x}_{g,k}^{(t)}\| \rightarrow \pm \mathbf{C} \mathbf{q}_k \|\mathbf{x}_{g,k}^{(t)}\| = \pm \lambda_k \mathbf{q}_k \|\mathbf{x}_{g,k}^{(t)}\|.$$

and,

$$\sum_{p=1}^{k-1} \lambda_p (\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{q}_p \mathbf{q}_p^T \tilde{\mathbf{x}}_{g,k}^{(t)} \rightarrow 0.$$

Thus from (4.12), we get

$$\|\mathbf{x}_{g,k}^{(t+1)}\|^2 - \|\mathbf{x}_{g,k}^{(t)}\|^2 \rightarrow 0,$$

which implies $\|\mathbf{x}_{g,k}^{(t)}\|$ converges to some constant $c_k > 0$ which further implies $\mathbf{x}_{g,k}^{(t)} \rightarrow \pm c_k \mathbf{q}_k$. \square

4.4 Main Results

In this section, we provide a detailed analysis proving that both versions of the FAST-PCA algorithm converge at a linear rate to the true eigenvectors $\mathbf{q}_k, k = 1, \dots, K$, of the global covariance matrix \mathbf{C} . Specifically, if $\mathbf{X}_i^{(t)} = [\mathbf{x}_{i,1}^{(t)}, \dots, \mathbf{x}_{i,K}^{(t)}] \in \mathbb{R}^{d \times K}$ is the estimate of the K eigenvectors at node i derived from Algorithm 2, then we show that square of the sine of the angle between $\mathbf{x}_{i,k}^{(t)}, \forall i = 1, \dots, M$, and \mathbf{q}_k for $k = 1, \dots, K$ converges to 0 at the rate of $\mathcal{O}(\rho^t)$ for some $\rho \in (0, 1)$.

If $\mathbf{W} = [w_{ij}]$ is the weight matrix underlying the graph representing the network, then the iterates of FAST-PCA-O/K for the estimation of the k^{th} eigenvector are given as follows:

$$\mathbf{x}_{i,k}^{(t+1)} = \frac{1}{2}\mathbf{x}_{i,k}^{(t)} + \sum_{j \in \mathcal{N}_i} \frac{w_{ij}}{2}\mathbf{x}_{j,k}^{(t)} + \alpha \mathbf{s}_{i,k}^{(t)} \quad (4.13)$$

$$\mathbf{s}_{i,k}^{(t+1)} = \frac{1}{2}\mathbf{s}_{i,k}^{(t)} + \sum_{j \in \mathcal{N}_i} \frac{w_{ij}}{2}\mathbf{s}_{j,k}^{(t)} + \mathbf{h}_i^{O/K}(\mathbf{x}_{i,k}^{(t+1)}) - \mathbf{h}_i^{O/K}(\mathbf{x}_{i,k}^{(t)}), \quad (4.14)$$

where $\mathbf{x}_{i,k}^{(t)}$ is the estimate of the k^{th} eigenvector, $\mathbf{s}_{i,k}^{(t)}$ is the estimate of the average pseudo-gradients $\mathbf{h}_i^{O/K}(\mathbf{x}_{i,k}^{(t+1)})$ given as :

$$\begin{aligned} \text{FAST-PCA-O : } \mathbf{h}_i^O(\mathbf{x}_{i,k}^{(t)}) &= \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)} - \sum_{p=1}^{k-1} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,p}^{(t)} \\ \text{FAST-PCA-K : } \mathbf{h}_i^K(\mathbf{x}_{i,k}^{(t)}) &= \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - \frac{(\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)}}{\|\mathbf{x}_{i,k}^{(t)}\|^2} \mathbf{x}_{i,k}^{(t)} - \sum_{p=1}^{k-1} \frac{\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i}{\|\mathbf{x}_{i,p}^{(t)}\|^2} \mathbf{x}_{i,k}^{(t)} \end{aligned}$$

Let us define the following stacked versions of the quantities $\mathbf{x}_{i,k}^{(t)}$, $\mathbf{s}_{i,k}^{(t)}$, and $\mathbf{h}_i(\mathbf{x}_{i,k}^{(t)})$ for $i = 1, \dots, M$ as

$$\mathbf{x}_k^{(t)} = \begin{bmatrix} \mathbf{x}_{1,k}^{(t)} \\ \mathbf{x}_{2,k}^{(t)} \\ \vdots \\ \mathbf{x}_{M,k}^{(t)} \end{bmatrix}, \quad \mathbf{h}^{O/K}(\mathbf{x}_k^{(t)}) = \begin{bmatrix} \mathbf{h}_1^{O/K}(\mathbf{x}_{1,k}^{(t)}) \\ \mathbf{h}_2^{O/K}(\mathbf{x}_{2,k}^{(t)}) \\ \vdots \\ \mathbf{h}_M^{O/K}(\mathbf{x}_{M,k}^{(t)}) \end{bmatrix}, \quad \mathbf{s}_k^{(t)} = \begin{bmatrix} \mathbf{s}_{1,k}^{(t)} \\ \mathbf{s}_{2,k}^{(t)} \\ \vdots \\ \mathbf{s}_{M,k}^{(t)} \end{bmatrix}$$

. Using these definitions (4.13) and (4.14) can be re-written as

$$\mathbf{x}_k^{(t+1)} = \frac{1}{2}((\mathbf{I}_M + \mathbf{W}) \otimes \mathbf{I}_d) \mathbf{x}_k^{(t)} + \alpha \mathbf{s}_k^{(t)}, \quad (4.15)$$

$$\mathbf{s}_k^{(t+1)} = \frac{1}{2}((\mathbf{I}_M + \mathbf{W}) \otimes \mathbf{I}_d) \mathbf{s}_k^{(t)} + \mathbf{h}^{O/K}(\mathbf{x}_k^{(t+1)}) - \mathbf{h}^{O/K}(\mathbf{x}_k^{(t)}). \quad (4.16)$$

Combining (4.15) and (4.16),

$$\mathbf{x}_k^{(t+2)} = ((\mathbf{I}_M + \mathbf{W}) \otimes \mathbf{I}_d) \mathbf{x}_k^{(t+1)} - \frac{1}{4}((\mathbf{I}_M + \mathbf{W}) \otimes \mathbf{I}_d)^2 \mathbf{x}_k^{(t)} + \alpha \mathbf{h}^{O/K}(\mathbf{x}_k^{(t+1)}) - \alpha \mathbf{h}^{O/K}(\mathbf{x}_k^{(t)}). \quad (4.17)$$

Next, let $\bar{\mathbf{x}}_k^{(t)}$ and $\bar{\mathbf{s}}_k^{(t)}$ denote the average of $\{\mathbf{x}_{i,k}^{(t)}\}_{i=1}^M$ and $\{\mathbf{s}_{i,k}^{(t)}\}_{i=1}^M$, respectively. Taking the average of (4.13) and (4.14) over all nodes $i = 1, \dots, M$, we get

$$\frac{1}{M} \sum_{i=1}^M \mathbf{x}_{i,k}^{(t+1)} = \bar{\mathbf{x}}_k^{(t+1)} = \bar{\mathbf{x}}_k^{(t)} + \alpha \bar{\mathbf{s}}_k^{(t)} \quad (4.18)$$

$$\frac{1}{M} \sum_{i=1}^M \mathbf{s}_{i,k}^{(t+1)} = \bar{\mathbf{s}}_k^{(t+1)} = \bar{\mathbf{s}}_k^{(t)} + \mathbf{g}(\mathbf{x}_k^{(t+1)}) - \mathbf{g}(\mathbf{x}_k^{(t)}) = \mathbf{g}(\mathbf{x}_k^{(t+1)}), \quad (4.19)$$

where

$$\mathbf{g}(\mathbf{x}_k^{(t)}) = \begin{cases} \frac{1}{M} \sum_{i=1}^M \mathbf{h}_i^O(\mathbf{x}_{i,k}^{(t)}), & \text{for FAST-PCA-O} \\ \frac{1}{M} \sum_{i=1}^M \mathbf{h}_i^K(\mathbf{x}_{i,k}^{(t)}), & \text{for FAST-PCA-K.} \end{cases}$$

Additionally, we also define the following stacked versions (denoted by subscript ‘s’) such that all these are in \mathbb{R}^{Md} .

$$\bar{\mathbf{x}}_{s,k}^{(t)} = \begin{bmatrix} \bar{\mathbf{x}}_k^{(t)} \\ \bar{\mathbf{x}}_k^{(t)} \\ \vdots \\ \bar{\mathbf{x}}_k^{(t)} \end{bmatrix}, \quad \bar{\mathbf{s}}_{s,k}^{(t)} = \begin{bmatrix} \bar{\mathbf{s}}_k^{(t)} \\ \bar{\mathbf{s}}_k^{(t)} \\ \vdots \\ \bar{\mathbf{s}}_k^{(t)} \end{bmatrix}, \quad \mathbf{g}_s(\mathbf{x}_k^{(t)}) = \begin{bmatrix} \mathbf{g}(\mathbf{x}_k^{(t)}) \\ \mathbf{g}(\mathbf{x}_k^{(t)}) \\ \vdots \\ \mathbf{g}(\mathbf{x}_k^{(t)}) \end{bmatrix}.$$

Thus,

$$\bar{\mathbf{s}}_{s,k}^{(t+1)} = \bar{\mathbf{s}}_{s,k}^{(t)} + \mathbf{g}_s(\mathbf{x}_k^{(t+1)}) - \mathbf{g}_s(\mathbf{x}_k^{(t)}) = \mathbf{g}_s(\mathbf{x}_k^{(t+1)}) \quad (4.20)$$

$$\bar{\mathbf{x}}_{s,k}^{(t+1)} = \bar{\mathbf{x}}_{s,k}^{(t)} + \alpha \bar{\mathbf{s}}_{s,k}^{(t)} = \bar{\mathbf{x}}_{s,k}^{(t)} + \alpha \mathbf{g}_s(\mathbf{x}_k^{(t)}) \quad (4.21)$$

$$\mathbf{g}(\bar{\mathbf{x}}_{s,k}^{(t)}) = \begin{cases} \frac{1}{M} \left(\mathbf{C} \bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)} - \sum_{i=1}^M \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} \right) & \text{for FAST-PCA-O} \\ \frac{1}{M} \left(\mathbf{C} \bar{\mathbf{x}}_k^{(t)} - \frac{(\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)}}{\|\bar{\mathbf{x}}_k^{(t)}\|^2} \bar{\mathbf{x}}_k^{(t)} - \sum_{i=1}^M \sum_{p=1}^{k-1} \frac{\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i}{\|\mathbf{x}_{i,p}^{(t)}\|^2} \bar{\mathbf{x}}_k^{(t)} \right) & \text{for FAST-PCA-K.} \end{cases} \quad (4.22)$$

Now, we first show that convergence of $\mathbf{x}_{i,1}^{(t)}$ at a linear rate and then proceed with the proof for $k = 2, \dots, K$ through induction.

Case I for Induction – $k = 1$: The iterates of FAST-PCA for estimation of the dominant eigenvector are

$$\mathbf{x}_{i,1}^{(t+1)} = \frac{1}{2} \mathbf{x}_{i,1}^{(t)} + \sum_{j \in \mathcal{N}_i} \frac{w_{ij}}{2} \mathbf{x}_{j,1}^{(t)} + \alpha \mathbf{s}_{i,1}^{(t)} \quad (4.23)$$

$$\mathbf{s}_{i,1}^{(t+1)} = \frac{1}{2} \mathbf{s}_{i,1}^{(t)} + \sum_{j \in \mathcal{N}_i} \frac{w_{ij}}{2} \mathbf{s}_{j,1}^{(t)} + \mathbf{h}_i^{O/K}(\mathbf{x}_{i,1}^{(t+1)}) - \mathbf{h}_i^{O/K}(\mathbf{x}_{i,1}^{(t)}), \quad (4.24)$$

where

$$\begin{aligned} \mathbf{h}_i^O(\mathbf{x}_{i,1}^{(t)}) &= \mathbf{C}_i \mathbf{x}_{i,1}^{(t)} - (\mathbf{x}_{i,1}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,1}^{(t)} \mathbf{x}_{i,1}^{(t)} \\ \text{and } \mathbf{h}_i^K(\mathbf{x}_{i,1}^{(t)}) &= \mathbf{C}_i \mathbf{x}_{i,1}^{(t)} - \frac{(\mathbf{x}_{i,1}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,1}^{(t)}}{\|\mathbf{x}_{i,1}^{(t)}\|^2} \mathbf{x}_{i,1}^{(t)} \end{aligned}$$

We first prove the Lipschitz continuity of both these pseudo-gradients. Lemma 12 shows $\mathbf{h}_i^O(\mathbf{x}_{i,1}^{(t)})$ is Lipschitz continuous if $\|\mathbf{x}_{i,1}^{(t)}\|$ is bounded $\forall t$. While Lemma 13 shows $\mathbf{h}_i^K(\mathbf{x}_{i,1}^{(t)})$ is also Lipschitz continuous. The proofs of these lemmas are given in Subsection 4.5.2. The Lipschitz continuity of \mathbf{g} can be proved by simple extension as shown in Lemma 14. We now

present our first main theorem for FAST-PCA-O that proves the convergence of the iterate $\mathbf{x}_{i,1}^{(t)}$ at node i to $\mathbf{x}_1^* = \pm \mathbf{q}_1$, where c_1 is a constant.

Theorem 5. *Suppose the estimate $\mathbf{x}_{i,1}^{(t)}$ from FAST-PCA-O remains bounded i.e., $\|\mathbf{x}_{i,1}^{(t)}\|^2 \leq \mu$, $\alpha \leq \frac{\lambda_1 - \lambda_2}{2\lambda_1^2(1+3\mu)^2}(\frac{1-\beta}{9})^2$ where λ_1, λ_2 are the largest and second largest eigenvalues of \mathbf{C} and $\beta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_M(\mathbf{W})|\}$, $\mathbf{q}_k^T \mathbf{x}_{i,k}^{(0)} \neq 0$, and the graph underlying the network is connected. Then the estimate $\mathbf{x}_{i,k}^{(t)}$ from FAST-PCA-O converges to the eigenvector $\pm \mathbf{q}_k$ corresponding to the largest eigenvalue λ_k of \mathbf{C} at each node $i = 1, \dots, M$ at a linear rate.*

Proof. For proving the convergence of $\mathbf{x}_{i,1}^{(t)}$ to $\mathbf{x}_1^* = \pm \mathbf{q}_1$, $\forall i = 1, \dots, M$, we prove that the distance of average $\bar{\mathbf{x}}_1^{(t)}$ from \mathbf{x}_1^* , the consensus error as well as the distance of $\mathbf{s}_{i,1}^{(t)}$ from the average pseudo-gradient $\mathbf{g}(\mathbf{x}_1^{(t)}) = \frac{1}{M} \sum_{i=1}^M \mathbf{h}_i^O(\mathbf{x}_{i,1}^{(t)})$ decay to zero at a linear rate. From (4.16), we have

$$\mathbf{s}_1^{(t)} - \mathbf{g}_s(\mathbf{x}_1^{(t)}) = \frac{1}{2}((\mathbf{I}_M + \mathbf{W}) \otimes \mathbf{I}_d) \mathbf{s}_1^{(t-1)} - \mathbf{g}_s(\mathbf{x}_1^{(t-1)}) + \mathbf{h}(\mathbf{x}_1^{(t)}) - \mathbf{h}(\mathbf{x}_1^{(t-1)}) - (\mathbf{g}_s(\mathbf{x}_1^{(t)}) - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})).$$

From the definitions of $\bar{\mathbf{s}}_1^{(t)}$, $\mathbf{g}(\mathbf{x}_1^{(t-1)})$ and $\mathbf{g}_s(\mathbf{x}_1^{(t-1)})$, it is obvious that $(\frac{1}{M} \mathbf{1} \mathbf{1}^T \otimes \mathbf{I}_d) \mathbf{s}_1^{(t-1)} = \mathbf{1} \bar{\mathbf{s}}_1^{(t-1)} = \bar{\mathbf{s}}_{s,1}^{(t-1)} = \mathbf{g}_s(\mathbf{x}_1^{(t-1)})$. Thus,

$$\begin{aligned} & \|\mathbf{s}_1^{(t)} - \mathbf{g}_s(\mathbf{x}_1^{(t)})\| \\ &= \left\| \frac{1}{2}((\mathbf{I}_M + \mathbf{W}) \otimes \mathbf{I}_d) \mathbf{s}_1^{(t-1)} - \mathbf{g}_s(\mathbf{x}_1^{(t-1)}) + \mathbf{h}(\mathbf{x}_1^{(t)}) - \mathbf{h}(\mathbf{x}_1^{(t-1)}) - (\mathbf{g}_s(\mathbf{x}_1^{(t)}) - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})) \right\| \\ &= \left\| \frac{1}{2}((\mathbf{I}_M + \mathbf{W}) \otimes \mathbf{I}_d) \mathbf{s}_1^{(t-1)} - \left(\frac{1}{M} \mathbf{1} \mathbf{1}^T \otimes \mathbf{I}_d \right) \mathbf{s}_1^{(t-1)} + \mathbf{g}_s(\mathbf{x}_1^{(t-1)}) - \mathbf{g}_s(\mathbf{x}_1^{(t-1)}) + \right. \\ & \quad \left. \mathbf{h}(\mathbf{x}_1^{(t)}) - \mathbf{h}(\mathbf{x}_1^{(t-1)}) - (\mathbf{g}_s(\mathbf{x}_1^{(t)}) - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})) \right\| \\ &= \left\| \left(\frac{1}{2}((\mathbf{I}_M + \mathbf{W}) \otimes \mathbf{I}_d) - \frac{1}{M} \mathbf{1} \mathbf{1}^T \otimes \mathbf{I}_d \right) (\mathbf{s}_1^{(t-1)} - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})) + \right. \\ & \quad \left. \mathbf{h}(\mathbf{x}_1^{(t)}) - \mathbf{h}(\mathbf{x}_1^{(t-1)}) - (\mathbf{g}_s(\mathbf{x}_1^{(t)}) - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})) \right\| \\ &\leq \left\| \left(\frac{1}{2}(\mathbf{I}_M + \mathbf{W}) - \frac{1}{M} \mathbf{1} \mathbf{1}^T \right) \otimes \mathbf{I}_d (\mathbf{s}_1^{(t-1)} - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})) \right\| + \\ & \quad \left\| \mathbf{h}(\mathbf{x}_1^{(t)}) - \mathbf{h}(\mathbf{x}_1^{(t-1)}) - (\mathbf{g}_s(\mathbf{x}_1^{(t)}) - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})) \right\|. \end{aligned} \tag{4.25}$$

Next, we simplify the second term of the above inequality (4.25) as follows:

$$\begin{aligned} & \|\mathbf{h}(\mathbf{x}_1^{(t)}) - \mathbf{h}(\mathbf{x}_1^{(t-1)}) - (\mathbf{g}_s(\mathbf{x}_1^{(t)}) - \mathbf{g}_s(\mathbf{x}_1^{(t-1)}))\|^2 \\ &= \|\mathbf{h}(\mathbf{x}_1^{(t)}) - \mathbf{h}(\mathbf{x}_1^{(t-1)})\|^2 + \|\mathbf{g}_s(\mathbf{x}_1^{(t)}) - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})\|^2 - 2\langle \mathbf{h}(\mathbf{x}_1^{(t)}) - \mathbf{h}(\mathbf{x}_1^{(t-1)}), \mathbf{g}_s(\mathbf{x}_1^{(t)}) - \mathbf{g}_s(\mathbf{x}_1^{(t-1)}) \rangle \\ &= \|\mathbf{h}(\mathbf{x}_1^{(t)}) - \mathbf{h}(\mathbf{x}_1^{(t-1)})\|^2 + \|\mathbf{g}_s(\mathbf{x}_1^{(t)}) - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})\|^2 - 2 \sum_{i=1}^M \langle \mathbf{h}_i(\mathbf{x}_{i,1}^{(t)}) - \mathbf{h}_i(\mathbf{x}_{i,1}^{(t-1)}), \mathbf{g}(\mathbf{x}_1^{(t)}) - \mathbf{g}(\mathbf{x}_1^{(t-1)}) \rangle \\ &= \|\mathbf{h}(\mathbf{x}_1^{(t)}) - \mathbf{h}(\mathbf{x}_1^{(t-1)})\|^2 + M \|\mathbf{g}(\mathbf{x}_1^{(t)}) - \mathbf{g}(\mathbf{x}_1^{(t-1)})\|^2 - 2M \langle \mathbf{g}(\mathbf{x}_1^{(t)}) - \mathbf{g}(\mathbf{x}_1^{(t-1)}), \mathbf{g}(\mathbf{x}_1^{(t)}) - \mathbf{g}(\mathbf{x}_1^{(t-1)}) \rangle \\ &= \|\mathbf{h}(\mathbf{x}_1^{(t)}) - \mathbf{h}(\mathbf{x}_1^{(t-1)})\|^2 - M \|\mathbf{g}(\mathbf{x}_1^{(t)}) - \mathbf{g}(\mathbf{x}_1^{(t-1)})\|^2 \\ &\leq \|\mathbf{h}(\mathbf{x}_1^{(t)}) - \mathbf{h}(\mathbf{x}_1^{(t-1)})\|^2. \end{aligned}$$

Thus,

$$\|\mathbf{h}(\mathbf{x}_1^{(t)}) - \mathbf{h}(\mathbf{x}_1^{(t-1)}) - (\mathbf{g}_s(\mathbf{x}_1^{(t)}) - \mathbf{g}_s(\mathbf{x}_1^{(t-1)}))\| \leq \|\mathbf{h}(\mathbf{x}_1^{(t)}) - \mathbf{h}(\mathbf{x}_1^{(t-1)})\| \leq L_1 \|\mathbf{x}_1^{(t)} - \mathbf{x}_1^{(t-1)}\| \quad (4.24)$$

From (4.25) and (4.26), we have the following

$$\begin{aligned} \|\mathbf{s}_1^{(t)} - \mathbf{g}_s(\mathbf{x}_1^{(t)})\| &\leq \|((\frac{1}{2}(\mathbf{I}_M + \mathbf{W}) - \frac{1}{M}\mathbf{1}\mathbf{1}^T) \otimes \mathbf{I}_d)(\mathbf{s}_1^{(t-1)} - \mathbf{g}_s(\mathbf{x}_1^{(t-1)}))\| + L_1 \|\mathbf{x}_1^{(t)} - \mathbf{x}_1^{(t-1)}\| \\ &\leq \frac{1+\beta}{2} \|\mathbf{s}_1^{(t-1)} - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})\| + L_1 \|\mathbf{x}_1^{(t)} - \mathbf{x}_1^{(t-1)}\|, \end{aligned} \quad (4.27)$$

where β is absolute value of the second largest eigenvalue of the weight matrix \mathbf{W} i.e., $\beta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_M(\mathbf{W})|\}$. As pointed out before, network connectivity ensures that $\beta < 1$.

Next, from (4.15) and (4.21), we have

$$\begin{aligned} \mathbf{x}_1^{(t)} - \bar{\mathbf{x}}_{s,1}^{(t)} &= \frac{1}{2}((\mathbf{I}_M + \mathbf{W}) \otimes \mathbf{I}_d)\mathbf{x}_1^{(t-1)} - \bar{\mathbf{x}}_{s,1}^{(t-1)} + \alpha(\mathbf{s}_1^{(t-1)} - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})) \\ &= ((\frac{1}{2}(\mathbf{I}_M + \mathbf{W}) - \frac{1}{M}\mathbf{1}\mathbf{1}^T) \otimes \mathbf{I}_d)(\mathbf{x}_1^{(t-1)} - \bar{\mathbf{x}}_{s,1}^{(t-1)}) + \alpha(\mathbf{s}_1^{(t-1)} - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})). \end{aligned}$$

Thus,

$$\|\mathbf{x}_1^{(t)} - \bar{\mathbf{x}}_{s,1}^{(t)}\| \leq \frac{1+\beta}{2} \|\mathbf{x}_1^{(t-1)} - \bar{\mathbf{x}}_{s,1}^{(t-1)}\| + \alpha \|\mathbf{s}_1^{(t-1)} - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})\|. \quad (4.28)$$

Next, we bound $\|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\|$. We know from (4.18)

$$\begin{aligned} \bar{\mathbf{x}}_1^{(t)} &= \bar{\mathbf{x}}_1^{(t-1)} + \alpha \bar{\mathbf{s}}_1^{(t-1)} = \bar{\mathbf{x}}_1^{(t-1)} + \alpha \mathbf{g}(\mathbf{x}_1^{(t-1)}) \\ &= \bar{\mathbf{x}}_1^{(t-1)} + \alpha \mathbf{g}(\bar{\mathbf{x}}_{s,1}^{(t-1)}) + \alpha(\mathbf{g}(\mathbf{x}_1^{(t-1)}) - \mathbf{g}(\bar{\mathbf{x}}_{s,1}^{(t-1)})) \\ &= \bar{\mathbf{x}}_1^{(t-1)} + \frac{\alpha}{M}(\mathbf{C}\bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C}\bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)}) + \alpha(\mathbf{g}(\mathbf{x}_1^{(t-1)}) - \mathbf{g}(\bar{\mathbf{x}}_{s,1}^{(t-1)})). \end{aligned}$$

We know from Theorem 1 in the previous chapter that any general iterate of the form

$$\mathbf{x}_{g,1}^{(t)} = \mathbf{x}_{g,1}^{(t-1)} + \alpha(\mathbf{C}\mathbf{x}_{g,1}^{(t-1)} - (\mathbf{x}_{g,1}^{(t-1)})^T \mathbf{C}\mathbf{x}_{g,1}^{(t-1)} \mathbf{x}_{g,1}^{(t-1)}) \quad (4.29)$$

converges at a linear rate to the eigenvector $\pm \mathbf{q}_1$ corresponding to the largest eigenvalue λ_1 of \mathbf{C} if the top two eigenvalues of \mathbf{C} are distinct, i.e., $\lambda_1 > \lambda_2$. Mathematically,

$$\|\mathbf{x}_{g,1}^{(t)} - \mathbf{x}_1^*\| \leq \delta_1 \|\mathbf{x}_{g,1}^{(t-1)} - \mathbf{x}_1^*\| \quad \text{for } 0 < \delta_1 = \frac{1 + \alpha\lambda_2}{1 + \alpha\lambda_1} < 1 \quad \text{and } \mathbf{x}_1^* = \mathbf{q}_1 \quad \text{or} \quad -\mathbf{q}_1. \quad (4.30)$$

Thus,

$$\begin{aligned} \|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\| &\leq \delta_1 \|\bar{\mathbf{x}}_1^{(t-1)} - \mathbf{x}_1^*\| + \alpha \|\mathbf{g}(\mathbf{x}_1^{(t-1)}) - \mathbf{g}(\bar{\mathbf{x}}_{s,1}^{(t-1)})\| \\ &\leq \delta_1 \|\bar{\mathbf{x}}_1^{(t-1)} - \mathbf{x}_1^*\| + \alpha \frac{L_1}{\sqrt{M}} \|\mathbf{x}_1^{(t-1)} - \bar{\mathbf{x}}_{s,1}^{(t-1)}\|. \end{aligned} \quad (4.31)$$

Now, we will bound $\|\mathbf{x}_1^{(t)} - \mathbf{x}_1^{(t-1)}\|$. We know from (4.22)

$$\mathbf{g}(\bar{\mathbf{x}}_{s,1}^{(t)}) = \frac{1}{M}(\mathbf{C}\bar{\mathbf{x}}_1^{(t)} - (\bar{\mathbf{x}}_1^{(t)})^T \mathbf{C}\bar{\mathbf{x}}_1^{(t)} \bar{\mathbf{x}}_1^{(t)}).$$

Thus $\mathbf{g}(\mathbf{x}_{s,1}^*) = \frac{1}{M}(\mathbf{C}\mathbf{x}_1^* - (\mathbf{x}_1^*)^T \mathbf{C}\mathbf{x}_1^* \mathbf{x}_1^*) = 0$, where $\mathbf{x}_{s,1}^* = [(\mathbf{x}_1^*)^T, \dots, (\mathbf{x}_1^*)^T]^T$. Hence,

$$\|\mathbf{g}_s(\bar{\mathbf{x}}_{s,1}^{(t-1)})\| = \sqrt{M}\|\mathbf{g}(\bar{\mathbf{x}}_{s,1}^{(t-1)})\| = \sqrt{M}\|\mathbf{g}(\bar{\mathbf{x}}_{s,1}^{(t-1)}) - \mathbf{g}(\mathbf{x}_{s,1}^*)\| \leq L_1\sqrt{M}\|\bar{\mathbf{x}}_1^{(t-1)} - \mathbf{x}_1^*\|.$$

Using the above inequality and Lemma 14, we get

$$\begin{aligned} \|\mathbf{s}_1^{(t-1)}\| &= \|\mathbf{s}_1^{(t-1)} - \mathbf{g}_s(\mathbf{x}_1^{(t-1)}) + \mathbf{g}_s(\mathbf{x}_1^{(t-1)}) - \mathbf{g}_s(\bar{\mathbf{x}}_{s,1}^{(t-1)}) + \mathbf{g}_s(\bar{\mathbf{x}}_{s,1}^{(t-1)})\| \\ &\leq \|\mathbf{s}_1^{(t-1)} - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})\| + \|\mathbf{g}_s(\mathbf{x}_1^{(t-1)}) - \mathbf{g}_s(\bar{\mathbf{x}}_{s,1}^{(t-1)})\| + \|\mathbf{g}_s(\bar{\mathbf{x}}_{s,1}^{(t-1)})\| \\ &\leq \|\mathbf{s}_1^{(t-1)} - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})\| + L_1\|\mathbf{x}_1^{(t-1)} - \bar{\mathbf{x}}_{s,1}^{(t-1)}\| + L_1\sqrt{M}\|\bar{\mathbf{x}}_1^{(t-1)} - \mathbf{x}_1^*\|, \end{aligned} \quad (4.32)$$

where $L_1 = \lambda_1(1 + 3\mu)$. Thus

$$\begin{aligned} \|\mathbf{x}_1^{(t)} - \mathbf{x}_1^{(t-1)}\| &= \left\| \frac{1}{2}((\mathbf{I}_M + \mathbf{W}) \otimes \mathbf{I}_d)\mathbf{x}_1^{(t-1)} - \mathbf{x}_1^{(t-1)} + \alpha\mathbf{s}_1^{(t-1)} \right\| \\ &= \left\| \left(\frac{1}{2}((\mathbf{I}_M + \mathbf{W}) \otimes \mathbf{I}_d) - \mathbf{I}_{Md} \right)(\mathbf{x}_1^{(t-1)} - \bar{\mathbf{x}}_{s,1}^{(t-1)}) + \alpha\mathbf{s}_1^{(t-1)} \right\| \\ &\leq 2\|\mathbf{x}_1^{(t-1)} - \bar{\mathbf{x}}_{s,1}^{(t-1)}\| + \alpha\|\mathbf{s}_1^{(t-1)}\| \\ &\leq \alpha\|\mathbf{s}_1^{(t-1)} - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})\| + (2 + \alpha L_1)\|\mathbf{x}_1^{(t-1)} - \bar{\mathbf{x}}_{s,1}^{(t-1)}\| + \alpha L_1\sqrt{M}\|\bar{\mathbf{x}}_1^{(t-1)} - \mathbf{x}_1^*\|, \end{aligned} \quad (4.33)$$

where the second last inequality is because $\|\frac{1}{2}((\mathbf{I}_M + \mathbf{W}) \otimes \mathbf{I}_d) - \mathbf{I}_{Md}\| \leq \|\frac{1}{2}(\mathbf{I}_M + \mathbf{W})\| + \|\mathbf{I}_{Md}\| = 2$ and last inequality is using (??). Using the above inequality in (4.27), we get

$$\begin{aligned} \|\mathbf{s}_1^{(t)} - \mathbf{g}_s(\mathbf{x}_1^{(t)})\| &\leq \left(\frac{1+\beta}{2} + \alpha L_1 \right) \|\mathbf{s}_1^{(t-1)} - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})\| + L_1(2 + \alpha L_1)\|\mathbf{x}_1^{(t-1)} - \bar{\mathbf{x}}_{s,1}^{(t-1)}\| \\ &\quad + \alpha L_1^2\sqrt{M}\|\bar{\mathbf{x}}_1^{(t-1)} - \mathbf{x}_1^*\|. \end{aligned} \quad (4.34)$$

Writing a system of equations from (4.28), (4.31) and (4.34), we have the following:

$$\begin{bmatrix} \|\mathbf{s}_1^{(t)} - \mathbf{g}_s(\mathbf{x}_1^{(t)})\| \\ \|\mathbf{x}_1^{(t)} - \bar{\mathbf{x}}_{s,1}^{(t)}\| \\ \sqrt{M}\|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\| \end{bmatrix} \leq \begin{bmatrix} (\frac{1+\beta}{2} + \alpha L_1) & L_1(2 + \alpha L_1) & \alpha L_1^2 \\ \alpha & \frac{1+\beta}{2} & 0 \\ 0 & \alpha L_1 & \delta_1 \end{bmatrix} \begin{bmatrix} \|\mathbf{s}_1^{(t-1)} - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})\| \\ \|\mathbf{x}_1^{(t-1)} - \bar{\mathbf{x}}_{s,1}^{(t-1)}\| \\ \sqrt{M}\|\bar{\mathbf{x}}_1^{(t-1)} - \mathbf{x}_1^*\| \end{bmatrix},$$

where \leq implies element-wise inequalities. Here $L_1 = \lambda_1(1 + 3\mu)$ and $\mathbf{x}_1^* = \pm \mathbf{q}_1$. Let us define

$$\mathbf{P}(\alpha) = \begin{bmatrix} (\frac{1+\beta}{2} + \alpha L_1) & L_1(2 + \alpha L_1) & \alpha L_1^2 \\ \alpha & \frac{1+\beta}{2} & 0 \\ 0 & \alpha L_1 & \delta_1 \end{bmatrix}.$$

Since $\mathbf{P}(\alpha)$ has non-negative entries and $\mathbf{P}^2(\alpha)$ has all positive entries, each entry of $\mathbf{P}^t(\alpha)$ will be $\mathcal{O}(\rho(\mathbf{P}(\alpha))^t)$, where $\rho(\mathbf{P}(\alpha))$ is the spectral radius of $\mathbf{P}(\alpha)$. If we choose α such that $\rho(\mathbf{P}(\alpha))$ is < 1 , then that implies $\|\mathbf{s}_1^{(t)} - \mathbf{g}_s(\mathbf{x}_1^{(t)})\|$, $\|\mathbf{x}_1^{(t)} - \bar{\mathbf{x}}_{s,1}^{(t)}\|$ and $\|\bar{\mathbf{x}}_1^{(t-1)} - \mathbf{x}_1^*\|$ converge at a linear rate. To find the required condition on α , we show in Lemma 15 that if $\alpha < \frac{\lambda_1 - \lambda_2}{42}(\frac{1-\beta}{9\lambda_1})^2$,

the spectral radius of $\mathbf{P}(\alpha)$ is strictly less than 1. This implies that if $\alpha < \frac{\lambda_1 - \lambda_2}{42} (\frac{1-\beta}{9\lambda_1})^2$, then $\|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\|$, $\|\mathbf{x}_{i,1}^{(t)} - \bar{\mathbf{x}}_1^{(t)}\|$ and $\|\mathbf{s}_{i,1}^{(t)} - \mathbf{g}(\mathbf{x}_1^{(t)})\|$ converge at a linear rate to 0. In other words, $\mathbf{x}_{i,1}^{(t)}$ converges linearly to $\mathbf{x}_1^* = \pm \mathbf{q}_1$. \square

Theorem 6. Suppose $\alpha \leq \frac{\lambda_1 - \lambda_2}{2\lambda_1^2} (\frac{1-\beta}{54})^2$, where λ_1, λ_2 are the largest and second-largest eigenvalues of \mathbf{C} , $\beta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_M(\mathbf{W})|\}$ is the second largest absolute eigenvalue of \mathbf{W} , $\mathbf{q}_1^T \mathbf{x}_{i,1}^{(0)} \neq 0$, and the graph underlying the network is connected. Then the estimate $\mathbf{x}_{i,1}^{(t)}$ from FAST-PCA-K converges to the eigenvector $\pm c_1 \mathbf{q}_1$ for some constant c_1 corresponding to the largest eigenvalue λ_1 of \mathbf{C} at each node $i = 1, \dots, M$ at a linear rate.

Proof. For proving the convergence of $\mathbf{x}_{i,1}^{(t)}$ to $\mathbf{x}_1^* = \pm c_1 \mathbf{q}_1$, $\forall i = 1, \dots, M$, we follow the same process as Theorem 5. We prove that the distance of average $\bar{\mathbf{x}}_1^{(t)}$ from \mathbf{x}_1^* , the consensus error as well as the distance of $\mathbf{s}_{i,1}^{(t)}$ from the average pseudo-gradient $\mathbf{g}(\mathbf{x}_1^{(t)}) = \frac{1}{M} \sum_{i=1}^M \mathbf{h}_i^K(\mathbf{x}_{i,1}^{(t)})$ decay to zero at a linear rate.

Now, We know from (4.18)

$$\begin{aligned} \bar{\mathbf{x}}_1^{(t)} &= \bar{\mathbf{x}}_1^{(t-1)} + \alpha \bar{\mathbf{s}}_1^{(t-1)} = \bar{\mathbf{x}}_1^{(t-1)} + \alpha \mathbf{g}(\mathbf{x}_1^{(t-1)}) \\ &= \bar{\mathbf{x}}_1^{(t-1)} + \alpha \mathbf{g}(\bar{\mathbf{x}}_{s,1}^{(t-1)}) + \alpha (\mathbf{g}(\mathbf{x}_1^{(t-1)}) - \mathbf{g}(\bar{\mathbf{x}}_{s,1}^{(t-1)})) \\ &= \bar{\mathbf{x}}_1^{(t-1)} + \frac{\alpha}{M} (\mathbf{C} \bar{\mathbf{x}}_1^{(t-1)} - \frac{(\bar{\mathbf{x}}_1^{(t-1)})^T \mathbf{C} \bar{\mathbf{x}}_1^{(t-1)}}{\|\bar{\mathbf{x}}_1^{(t-1)}\|^2} \bar{\mathbf{x}}_1^{(t-1)}) + \alpha (\mathbf{g}(\mathbf{x}_1^{(t-1)}) - \mathbf{g}(\bar{\mathbf{x}}_{s,1}^{(t-1)})). \end{aligned}$$

We know from Theorem 4 in the Subsection 4.3.2 that any general iterate of the form

$$\mathbf{x}_{g,1}^{(t)} = \mathbf{x}_{g,1}^{(t-1)} + \alpha (\mathbf{C} \mathbf{x}_{g,1}^{(t-1)} - \frac{(\mathbf{x}_{g,1}^{(t-1)})^T \mathbf{C} \mathbf{x}_{g,1}^{(t-1)}}{\|\mathbf{x}_{g,1}^{(t-1)}\|^2} \mathbf{x}_{g,1}^{(t-1)})$$

converges at a linear rate to a multiple of the eigenvector i.e., $\pm c_1 \mathbf{q}_1$ corresponding to the largest eigenvalue λ_1 of \mathbf{C} if the top two eigenvalues of \mathbf{C} are distinct, i.e., $\lambda_1 > \lambda_2$. Thus,

$$\begin{aligned} \|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\| &\leq \delta_1 \|\bar{\mathbf{x}}_1^{(t-1)} - \mathbf{x}_1^*\| + \alpha \|\mathbf{g}(\mathbf{x}_1^{(t-1)}) - \mathbf{g}(\bar{\mathbf{x}}_{s,1}^{(t-1)})\| \\ &\leq \delta_1 \|\bar{\mathbf{x}}_1^{(t-1)} - \mathbf{x}_1^*\| + \alpha \frac{L_1}{\sqrt{M}} \|\mathbf{x}_1^{(t-1)} - \bar{\mathbf{x}}_{s,1}^{(t-1)}\|, \end{aligned} \quad (4.35)$$

where $\mathbf{x}_1^* = \pm c_1 \mathbf{q}_1$, $\delta_1 = \frac{1+\alpha\lambda_2}{1+\alpha\lambda_1}$ and $L_1 = 6\lambda_1$. Proceeding exactly as in Theorem 5, we get

$$\begin{bmatrix} \|\mathbf{s}_1^{(t)} - \mathbf{g}_s(\mathbf{x}_1^{(t)})\| \\ \|\mathbf{x}_1^{(t)} - \bar{\mathbf{x}}_{s,1}^{(t)}\| \\ \sqrt{M} \|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\| \end{bmatrix} \leq \begin{bmatrix} (\frac{1+\beta}{2} + \alpha L_1) & L_1(2 + \alpha L_1) & \alpha L_1^2 \\ \alpha & \frac{1+\beta}{2} & 0 \\ 0 & \alpha L_1 & \delta_1 \end{bmatrix} \begin{bmatrix} \|\mathbf{s}_1^{(t-1)} - \mathbf{g}_s(\mathbf{x}_1^{(t-1)})\| \\ \|\mathbf{x}_1^{(t-1)} - \bar{\mathbf{x}}_{s,1}^{(t-1)}\| \\ \sqrt{M} \|\bar{\mathbf{x}}_1^{(t-1)} - \mathbf{x}_1^*\| \end{bmatrix}, \quad (4.36)$$

where \leq implies element-wise inequalities, $L_1 = 6\lambda_1$ and $\mathbf{x}_1^* = \pm c_1 \mathbf{q}_1$. Let us define

$$\mathbf{P}(\alpha) = \begin{bmatrix} (\frac{1+\beta}{2} + \alpha L_1) & L_1(2 + \alpha L_1) & \alpha L_1^2 \\ \alpha & \frac{1+\beta}{2} & 0 \\ 0 & \alpha L_1 & \delta_1 \end{bmatrix}.$$

Since $\mathbf{P}(\alpha)$ has non-negative entries and $\mathbf{P}^2(\alpha)$ has all positive entries, each entry of $\mathbf{P}^t(\alpha)$ will be $\mathcal{O}(\rho(\mathbf{P}(\alpha))^t)$, where $\rho(\mathbf{P}(\alpha))$ is the spectral radius of $\mathbf{P}(\alpha)$. If we choose α such that $\rho(\mathbf{P}(\alpha))$ is < 1 , then that implies $\|\mathbf{s}_1^{(t)} - \mathbf{g}_s(\mathbf{x}_1^{(t)})\|$, $\|\mathbf{x}_1^{(t)} - \bar{\mathbf{x}}_{s,1}^{(t)}\|$ and $\|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\|$ converge at a linear rate. To find the required condition on α , we show in Lemma 15 that if $\alpha < \frac{\lambda_1 - \lambda_2}{42} (\frac{1-\beta}{9\lambda_1})^2$, the spectral radius of $\mathbf{P}(\alpha)$ is strictly less than 1. This implies that if $\alpha < \frac{\lambda_1 - \lambda_2}{42} (\frac{1-\beta}{9\lambda_1})^2$, then $\|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\|$, $\|\mathbf{x}_{i,1}^{(t)} - \bar{\mathbf{x}}_1^{(t)}\|$ and $\|\mathbf{s}_{i,1}^{(t)} - \mathbf{g}(\mathbf{x}_1^{(t)})\|$ converge at a linear rate to 0. In other words, $\mathbf{x}_{i,1}^{(t)}$ converges linearly to $\mathbf{x}_1^* = \pm c_1 \mathbf{q}_1$, where c_1 is some constant. \square

Case II for Induction – $2 \leq k \leq K$:

We proceed with the proof of convergence for rest of the eigenvectors through induction. In case I, we proved the convergence of $\mathbf{x}_{i,1}^{(t)}$ for both FAST-PCA-O and FAST-PCA-K. By induction, we assume $\mathbf{x}_{i,p}^{(t)}$ converges for $p = 1, \dots, (k-1)$ at a linear rate,

Theorem 7. Suppose the estimate $\mathbf{x}_{i,p}^{(t)}$ from FAST-PCA-O remains bounded for $p = 1, \dots, k$ i.e., $\|\mathbf{x}_{i,p}^{(t)}\|^2 \leq \mu$, $\alpha \leq \frac{\min_{k=1, \dots, K} (\lambda_k - \lambda_{k+1})}{2\lambda_1^2(1+2\mu+k\mu)^2} (\frac{1-\beta}{9})^2$ where λ_k, λ_{k+1} are the k^{th} and $(k+1)^{\text{th}}$ largest eigenvalues of \mathbf{C} and $\beta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_M(\mathbf{W})|\}$, $\mathbf{q}_k^T \mathbf{x}_{i,k}^{(0)} \neq 0$, and the graph underlying the network is connected. Then the estimate $\mathbf{x}_{i,k}^{(t)}$ from FAST-PCA-O converges to the eigenvector $\pm \mathbf{q}_k$ corresponding to the largest eigenvalue λ_k of \mathbf{C} at each node $i = 1, \dots, M$ at a linear rate.

Proof. Assume that $\|\mathbf{x}_{i,p}^{(t)}\|^2 \leq \mu$ and $\mathbf{x}_{i,p}^{(t)}$ converges to $\pm \mathbf{q}_p$ for $p = 1, \dots, k-1$ linearly, i.e., there exist constants $b_i > 0$ and $\nu_i < 1$ such that

$$\left\| \sum_{p=1}^{k-1} \left(\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T - \mathbf{q}_p \mathbf{q}_p^T \right) \right\| \leq b_i \nu_i^t. \quad (4.37)$$

Using the definitions of $\mathbf{x}_k^{(t)}$, $\mathbf{s}_k^{(t)}$, $\mathbf{g}_k^{(t)}$, $\mathbf{h}(\mathbf{x}_k^{(t)})$ and same algebraic manipulations as in Theorem 6, we get

$$\|\mathbf{s}_k^{(t)} - \mathbf{g}_s(\mathbf{x}_k^{(t)})\| \leq \frac{1+\beta}{2} \|\mathbf{s}_k^{(t-1)} - \mathbf{g}_s(\mathbf{x}_k^{(t-1)})\| + L_k \|\mathbf{x}_k^{(t)} - \mathbf{x}_k^{(t-1)}\| \quad (4.38)$$

and,

$$\|\mathbf{x}_k^{(t)} - \bar{\mathbf{x}}_{s,k}^{(t)}\| \leq \frac{1+\beta}{2} \|\mathbf{x}_k^{(t-1)} - \bar{\mathbf{x}}_{s,k}^{(t-1)}\| + \alpha \|\mathbf{s}_k^{(t-1)} - \mathbf{g}_s(\mathbf{x}_k^{(t-1)})\|. \quad (4.39)$$

Now, we bound $\|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\|$. We know from (4.22)

$$\begin{aligned}
\mathbf{g}(\bar{\mathbf{x}}_{s,k}^{(t)}) &= \frac{1}{M} \left(\mathbf{C}\bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C}\bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)} - \sum_{i=1}^M \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} \right) \\
&= \frac{1}{M} \left(\mathbf{C}\bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C}\bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)} - \sum_{i=1}^M \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} \right) - \frac{1}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} \left(\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T - \mathbf{q}_p \mathbf{q}_p^T \right) \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} \\
&= \frac{1}{M} \left(\mathbf{C}\bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C}\bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)} - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)} \right) - \frac{1}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} \left(\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T - \mathbf{q}_p \mathbf{q}_p^T \right) \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} \\
&= \mathbf{g}'(\bar{\mathbf{x}}_{s,k}^{(t)}) - \mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t)})
\end{aligned}$$

where,

$$\mathbf{g}'(\bar{\mathbf{x}}_{s,k}^{(t)}) = \frac{1}{M} (\mathbf{C}\bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C}\bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)} - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)})$$

and

$$\mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t)}) = \frac{1}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} (\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T - \mathbf{q}_p \mathbf{q}_p^T) \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)}.$$

From (4.18), we have

$$\begin{aligned}
\bar{\mathbf{x}}_k^{(t)} &= \bar{\mathbf{x}}_k^{(t-1)} + \alpha \bar{\mathbf{s}}_k^{(t-1)} = \bar{\mathbf{x}}_k^{(t-1)} + \alpha \mathbf{g}(\mathbf{x}_k^{(t-1)}) \\
&= \bar{\mathbf{x}}_k^{(t-1)} + \alpha \mathbf{g}(\bar{\mathbf{x}}_{s,k}^{(t-1)}) + \alpha (\mathbf{g}(\mathbf{x}_k^{(t-1)}) - \mathbf{g}(\bar{\mathbf{x}}_{s,k}^{(t-1)})) \\
&= \bar{\mathbf{x}}_k^{(t-1)} + \frac{\alpha}{M} (\mathbf{C}\bar{\mathbf{x}}_k^{(t-1)} - (\bar{\mathbf{x}}_k^{(t-1)})^T \mathbf{C}\bar{\mathbf{x}}_k^{(t-1)} \bar{\mathbf{x}}_k^{(t-1)} - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t-1)}) - \alpha \mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t-1)}) \\
&\quad + \alpha (\mathbf{g}(\mathbf{x}_k^{(t-1)}) - \mathbf{g}(\bar{\mathbf{x}}_{s,k}^{(t-1)})).
\end{aligned}$$

We know from Theorem 1 in the previous chapter that any general iterate of the form

$$\mathbf{x}_{g,k}^{(t)} = \mathbf{x}_{g,k}^{(t-1)} + \alpha (\mathbf{C}\mathbf{x}_{g,k}^{(t-1)} - (\mathbf{x}_{g,k}^{(t-1)})^T \mathbf{C}\mathbf{x}_{g,k}^{(t-1)} \mathbf{x}_{g,k}^{(t-1)} - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \mathbf{x}_{g,k}^{(t-1)})$$

converges linearly as

$$\|\mathbf{x}_{g,k}^{(t)} - \mathbf{x}_k^*\| \leq \delta_k \|\mathbf{x}_{g,k}^{(t-1)} - \mathbf{x}_k^*\|$$

where $\mathbf{x}_k^* = \pm \mathbf{q}_k$ and $\delta_k = \frac{1+\alpha\lambda_{k+1}}{1+\alpha\lambda_k}$. Thus,

$$\begin{aligned}
\|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\| &\leq \delta_k \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| + \alpha \|\mathbf{g}(\mathbf{x}_k^{(t-1)}) - \mathbf{g}(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| + \alpha \|\mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| \\
&\leq \delta_k \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| + \alpha \frac{L_k}{\sqrt{M}} \|\mathbf{x}_k^{(t-1)} - \bar{\mathbf{x}}_{s,k}^{(t-1)}\| + \alpha \|\mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| \quad (4.40)
\end{aligned}$$

Now, we will bound $\|\mathbf{x}_k^{(t)} - \mathbf{x}_k^*\|$. Since $\mathbf{g}'(\bar{\mathbf{x}}_{s,k}^{(t)}) = \frac{1}{M} (\mathbf{C}\bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C}\bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)} - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)})$,

we have $\mathbf{g}'(\mathbf{x}_{s,k}^*) = \frac{1}{M} (\mathbf{C}\mathbf{q}_k - \mathbf{q}_k^T \mathbf{C}\mathbf{q}_k \mathbf{q}_k - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \mathbf{q}_k) = 0$. Hence,

$$\|\mathbf{g}'_s(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| = \sqrt{M} \|\mathbf{g}'(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| = \sqrt{M} \|\mathbf{g}'(\bar{\mathbf{x}}_{s,k}^{(t-1)}) - \mathbf{g}'(\mathbf{x}_{s,k}^*)\| \leq L_k \sqrt{M} \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\|.$$

Using the above inequality and Lemma 14, we get

$$\begin{aligned}
\|\mathbf{s}_k^{(t-1)}\| &= \|\mathbf{s}_k^{(t-1)} - \mathbf{g}_s(\mathbf{x}_k^{(t-1)}) + \mathbf{g}_s(\mathbf{x}_k^{(t-1)}) - \mathbf{g}_s(\bar{\mathbf{x}}_{s,k}^{(t-1)}) + \mathbf{g}_s(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| \\
&= \|\mathbf{s}_k^{(t-1)} - \mathbf{g}_s(\mathbf{x}_k^{(t-1)}) + \mathbf{g}_s(\mathbf{x}_k^{(t-1)}) - \mathbf{g}_s(\bar{\mathbf{x}}_{s,k}^{(t-1)}) + \mathbf{g}_s'(\bar{\mathbf{x}}_{s,k}^{(t-1)}) - \mathbf{f}_s(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| \\
&\leq \|\mathbf{s}_k^{(t-1)} - \mathbf{g}_s(\mathbf{x}_k^{(t-1)})\| + \|\mathbf{g}_s(\mathbf{x}_k^{(t-1)}) - \mathbf{g}_s(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| + \|\mathbf{g}_s'(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| + \|\mathbf{f}_s(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| \\
&\leq \|\mathbf{s}_k^{(t-1)} - \mathbf{g}_s(\mathbf{x}_k^{(t-1)})\| + L_k \|\mathbf{x}_k^{(t-1)} - \bar{\mathbf{x}}_{s,k}^{(t-1)}\| + L_k \sqrt{M} \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| + \sqrt{M} \|\mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t-1)})\|.
\end{aligned} \tag{4.41}$$

Thus,

$$\begin{aligned}
\|\mathbf{x}_k^{(t)} - \mathbf{x}_k^{(t-1)}\| &= \|(\mathbf{W} \otimes \mathbf{I})\mathbf{x}_k^{(t-1)} - \mathbf{x}_k^{(t-1)} + \alpha \mathbf{s}_k^{(t-1)}\| \\
&= \|(\mathbf{W} \otimes \mathbf{I} - \mathbf{I})(\mathbf{x}_k^{(t-1)} - \bar{\mathbf{x}}_{s,k}^{(t-1)}) + \alpha \mathbf{s}_k^{(t-1)}\| \\
&\leq 2\|\mathbf{x}_k^{(t-1)} - \bar{\mathbf{x}}_{s,k}^{(t-1)}\| + \alpha \|\mathbf{s}_k^{(t-1)}\| \\
&\leq \alpha \|\mathbf{s}_k^{(t-1)} - \mathbf{g}_s(\mathbf{x}_k^{(t-1)})\| + (2 + \alpha L_k) \|\mathbf{x}_k^{(t-1)} - \bar{\mathbf{x}}_{s,k}^{(t-1)}\| + \alpha L_k \sqrt{M} \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| + \\
&\quad \alpha \sqrt{M} \|\mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| \quad \text{using (4.41)}
\end{aligned} \tag{4.42}$$

Using the above inequality in (4.45), we get

$$\begin{aligned}
\|\mathbf{s}_k^{(t)} - \mathbf{g}_s(\mathbf{x}_k^{(t)})\| &\leq \left(\frac{1+\beta}{2} + \alpha L_k\right) \|\mathbf{s}_k^{(t-1)} - \mathbf{g}_s(\mathbf{x}_k^{(t-1)})\| + L_k(2 + \alpha L_k) \|\mathbf{x}_k^{(t-1)} - \bar{\mathbf{x}}_{s,k}^{(t-1)}\| + \\
&\quad \alpha L_k^2 \sqrt{M} \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| + \alpha L_k \sqrt{M} \|\mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t-1)})\|.
\end{aligned} \tag{4.43}$$

Writing a system of equations from (4.43), (4.46) and (4.47), we have the following:

$$\begin{bmatrix} \|\mathbf{s}_k^{(t)} - \mathbf{g}_s(\mathbf{x}_k^{(t)})\| \\ \|\mathbf{x}_k^{(t)} - \bar{\mathbf{x}}_{s,k}^{(t)}\| \\ \sqrt{M} \|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\| \end{bmatrix} \leq \begin{bmatrix} (\frac{1+\beta}{2} + \alpha L_k) & L_k(2 + \alpha L_k) & \alpha L_k^2 \\ \alpha & \frac{1+\beta}{2} & 0 \\ 0 & \alpha L_k & \delta_k \end{bmatrix} \begin{bmatrix} \|\mathbf{s}_k^{(t-1)} - \mathbf{g}_s(\mathbf{x}_k^{(t-1)})\| \\ \|\mathbf{x}_k^{(t-1)} - \bar{\mathbf{x}}_{s,k}^{(t-1)}\| \\ \sqrt{M} \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| \end{bmatrix} + \|\mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| \begin{bmatrix} \alpha L_k \sqrt{M} \\ 0 \\ \alpha \sqrt{M} \end{bmatrix}.$$

Let us denote

$$\mathbf{P}_k(\alpha) = \begin{bmatrix} (\frac{1+\beta}{2} + \alpha L_k) & L_k(2 + \alpha L_k) & \alpha L_k^2 \\ \alpha & \frac{1+\beta}{2} & 0 \\ 0 & \alpha L_k & \rho_k \end{bmatrix}.$$

Since $\mathbf{P}_k(\alpha)$ has non-negative entries and $\mathbf{P}_k^2(\alpha)$ has all positive entries, each entry of $\mathbf{P}_k^t(\alpha)$ will be $\mathcal{O}(\rho(\mathbf{P}_k(\alpha))^t)$, where $\rho(\mathbf{P}_k(\alpha))$ is the spectral radius of $\mathbf{P}_k(\alpha)$. From Lemma 15 we know if we choose $\alpha < \frac{\lambda_k - \lambda_{k+1}}{(k+5)(k+6)} (\frac{1-\beta}{9\lambda_1})^2$, then $\rho(\mathbf{P}_k(\alpha)) < 1$. Also, we know $\mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t)}) = \frac{1}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} (\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T - \mathbf{q}_p \mathbf{q}_p^T) \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)}$. Thus,

$$\begin{aligned}
\|\mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t)})\| &= \left\| \frac{1}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} (\mathbf{x}_{i,p}^{(t-1)} (\mathbf{x}_{i,p}^{(t-1)})^T - \mathbf{q}_p \mathbf{q}_p^T) \mathbf{C}_i \bar{\mathbf{x}}_k^{(t-1)} \right\| \\
&\leq \frac{1}{M} \sum_{i=1}^M \left\| \sum_{p=1}^{k-1} (\mathbf{x}_{i,p}^{(t-1)} (\mathbf{x}_{i,p}^{(t-1)})^T - \mathbf{q}_p \mathbf{q}_p^T) \right\| \|\mathbf{C}_i\| \|\bar{\mathbf{x}}_k^{(t-1)}\|
\end{aligned}$$

From (4.37), we know $\|\sum_{p=1}^{k-1}(\mathbf{x}_{i,p}^{(t-1)}(\mathbf{x}_{i,p}^{(t-1)})^T - \mathbf{q}_p \mathbf{q}_p^T)\| \leq b_i \nu_i^t$. Let $b = (\max_i b_i) \lambda_1 \|\bar{\mathbf{x}}_k^{(t-1)}\| > 0$ and $\nu = \max_i \nu_i < 1$. Thus the system of equations becomes

$$\begin{bmatrix} \|\mathbf{s}_k^{(t)} - \mathbf{g}_s(\mathbf{x}_k^{(t)})\| \\ \|\mathbf{x}_k^{(t)} - \bar{\mathbf{x}}_{s,k}^{(t)}\| \\ \sqrt{M} \|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\| \end{bmatrix} \leq \rho(\mathbf{P}_k(\alpha))^t \begin{bmatrix} \|\mathbf{s}_k^{(0)} - \mathbf{g}_s(\mathbf{x}_k^{(0)})\| \\ \|\mathbf{x}_k^{(0)} - \bar{\mathbf{x}}_{s,k}^{(0)}\| \\ \sqrt{M} \|\bar{\mathbf{x}}_k^{(0)} - \mathbf{x}_k^*\| \end{bmatrix} + b \nu^t \begin{bmatrix} \alpha L_k \sqrt{M} \\ 0 \\ \alpha \sqrt{M} \end{bmatrix}.$$

This implies that if $\alpha < \frac{\lambda_k - \lambda_{k+1}}{(k+5)(k+6)} (\frac{1-\beta}{9\lambda_1})^2$, then $\|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\|$, $\|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\|$ and $\|\mathbf{s}_{i,k}^{(t)} - \mathbf{g}(\mathbf{x}_k^{(t)})\|$ converge at a linear rate to 0. In other words, $\mathbf{x}_{i,k}^{(t)}$ converges linearly to $\mathbf{x}_k^* = \pm \mathbf{q}_k$ under the assumption that $\|\mathbf{x}_{i,p}^{(t)}\|$ remains bounded for $p = 1, \dots, k$. \square

Theorem 8. Suppose $\alpha \leq \frac{\min_{k=1,\dots,K}(\lambda_k - \lambda_{k+1})}{2\lambda_1^2(K+5)^2} (\frac{1-\beta}{9})^2$ where λ_k, λ_{k+1} are the k^{th} and $(k+1)^{\text{th}}$ largest eigenvalues of \mathbf{C} , $\beta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_M(\mathbf{W})|\}$ and $\mathbf{q}_k^T \mathbf{x}_{i,k}^{(0)} \neq 0$ and the graph underlying the network is connected, then the estimate $\mathbf{x}_{i,k}^{(t)}$ from FAST-PCA-K converges to the eigenvector $\pm \mathbf{q}_k$ corresponding to the largest eigenvalue λ_k of \mathbf{C} at each node $i = 1, \dots, M$ at a linear rate.

Proof. Assume that $\mathbf{x}_{i,p}^{(t)}$ converges to $\pm c_p \mathbf{q}_p$ for $p = 1, \dots, k-1$ linearly, i.e., there exist constants $b_i > 0$ and $\nu_i < 1$ such that

$$\left\| \sum_{p=1}^{k-1} \left(\frac{\mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T}{\|\mathbf{x}_{i,p}^{(t)}\|^2} - \mathbf{q}_p \mathbf{q}_p^T \right) \right\| \leq b_i \nu_i^t. \quad (4.44)$$

Using the definitions of $\mathbf{x}_k^{(t)}$, $\mathbf{s}_k^{(t)}$, $\mathbf{g}_k^{(t)}$, $\mathbf{h}(\mathbf{x}_k^{(t)})$ and same algebraic manipulations as in Theorem 6, we get

$$\|\mathbf{s}_k^{(t)} - \mathbf{g}_s(\mathbf{x}_k^{(t)})\| \leq \frac{1+\beta}{2} \|\mathbf{s}_k^{(t-1)} - \mathbf{g}_s(\mathbf{x}_k^{(t-1)})\| + L_k \|\mathbf{x}_k^{(t)} - \mathbf{x}_k^{(t-1)}\| \quad (4.45)$$

and,

$$\|\mathbf{x}_k^{(t)} - \bar{\mathbf{x}}_{s,k}^{(t)}\| \leq \frac{1+\beta}{2} \|\mathbf{x}_k^{(t-1)} - \bar{\mathbf{x}}_{s,k}^{(t-1)}\| + \alpha \|\mathbf{s}_k^{(t-1)} - \mathbf{g}_s(\mathbf{x}_k^{(t-1)})\|. \quad (4.46)$$

Now, we bound $\|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\|$. We know

$$\begin{aligned} \mathbf{g}(\bar{\mathbf{x}}_{s,k}^{(t)}) &= \frac{1}{M} \left(\mathbf{C} \bar{\mathbf{x}}_k^{(t)} - \frac{(\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)}}{\|\bar{\mathbf{x}}_k^{(t)}\|^2} \bar{\mathbf{x}}_k^{(t)} - \sum_{i=1}^M \sum_{p=1}^{k-1} \frac{\mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T}{\|\mathbf{x}_{i,p}^{(t)}\|^2} \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} \right) \\ &= \frac{1}{M} \left(\mathbf{C} \bar{\mathbf{x}}_k^{(t)} - \frac{(\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)}}{\|\bar{\mathbf{x}}_k^{(t)}\|^2} \bar{\mathbf{x}}_k^{(t)} - \sum_{i=1}^M \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} \right) - \frac{1}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} \left(\frac{\mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T}{\|\mathbf{x}_{i,p}^{(t)}\|^2} - \mathbf{q}_p \mathbf{q}_p^T \right) \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} \\ &= \frac{1}{M} \left(\mathbf{C} \bar{\mathbf{x}}_k^{(t)} - \frac{(\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)}}{\|\bar{\mathbf{x}}_k^{(t)}\|^2} \bar{\mathbf{x}}_k^{(t)} - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)} \right) - \frac{1}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} \left(\frac{\mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T}{\|\mathbf{x}_{i,p}^{(t)}\|^2} - \mathbf{q}_p \mathbf{q}_p^T \right) \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} \\ &= \mathbf{g}'(\bar{\mathbf{x}}_{s,k}^{(t)}) - \mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t)}) \end{aligned}$$

where,

$$\mathbf{g}'(\bar{\mathbf{x}}_{s,k}^{(t)}) = \frac{1}{M} \left(\mathbf{C} \bar{\mathbf{x}}_k^{(t)} - \frac{(\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)}}{\|\bar{\mathbf{x}}_k^{(t)}\|^2} \bar{\mathbf{x}}_k^{(t)} - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)} \right)$$

and

$$\mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t)}) = \frac{1}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} \left(\frac{\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T}{\|\mathbf{x}_{i,p}^{(t)}\|^2} - \mathbf{q}_p \mathbf{q}_p^T \right) \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)}.$$

From (4.18), we have

$$\begin{aligned} \bar{\mathbf{x}}_k^{(t)} &= \bar{\mathbf{x}}_k^{(t-1)} + \alpha \bar{\mathbf{s}}_k^{(t-1)} = \bar{\mathbf{x}}_k^{(t-1)} + \alpha \mathbf{g}(\mathbf{x}_k^{(t-1)}) \\ &= \bar{\mathbf{x}}_k^{(t-1)} + \alpha \mathbf{g}(\bar{\mathbf{x}}_{s,k}^{(t-1)}) + \alpha (\mathbf{g}(\mathbf{x}_k^{(t-1)}) - \mathbf{g}(\bar{\mathbf{x}}_{s,k}^{(t-1)})) \\ &= \bar{\mathbf{x}}_k^{(t-1)} + \frac{\alpha}{M} (\mathbf{C} \bar{\mathbf{x}}_k^{(t-1)} - \frac{(\bar{\mathbf{x}}_k^{(t-1)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t-1)}}{\|\bar{\mathbf{x}}_k^{(t-1)}\|^2} \bar{\mathbf{x}}_k^{(t-1)} - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t-1)}) - \alpha \mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t-1)}) \\ &\quad + \alpha (\mathbf{g}(\mathbf{x}_k^{(t-1)}) - \mathbf{g}(\bar{\mathbf{x}}_{s,k}^{(t-1)})). \end{aligned}$$

We know from Theorem 4 in the Subsection 4.3.2 that any general iterate of the form

$$\mathbf{x}_{g,k}^{(t)} = \mathbf{x}_{g,k}^{(t-1)} + \alpha (\mathbf{C} \mathbf{x}_{g,k}^{(t-1)} - \frac{(\mathbf{x}_{g,k}^{(t-1)})^T \mathbf{C} \mathbf{x}_{g,k}^{(t-1)}}{\|\mathbf{x}_{g,k}^{(t-1)}\|^2} \mathbf{x}_{g,k}^{(t-1)} - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \mathbf{x}_{g,k}^{(t-1)})$$

converges linearly as

$$\|\mathbf{x}_{g,k}^{(t)} - \mathbf{x}_k^*\| \leq \delta_k \|\mathbf{x}_{g,k}^{(t-1)} - \mathbf{x}_k^*\|$$

where $\mathbf{x}_k^* = \pm c_k \mathbf{q}_k$ and $\delta_k = \frac{1+\alpha\lambda_{k+1}}{1+\alpha\lambda_k}$. Thus,

$$\begin{aligned} \|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\| &\leq \delta_k \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| + \alpha \|\mathbf{g}(\mathbf{x}_k^{(t-1)}) - \mathbf{g}(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| + \alpha \|\mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| \\ &\leq \delta_k \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| + \alpha \frac{L_k}{\sqrt{M}} \|\mathbf{x}_k^{(t-1)} - \bar{\mathbf{x}}_{s,k}^{(t-1)}\| + \alpha \|\mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| \end{aligned} \quad (4.47)$$

Now, we will bound $\|\mathbf{x}_k^{(t)} - \mathbf{x}_k^{(t-1)}\|$. Since $\mathbf{g}'(\bar{\mathbf{x}}_{s,k}^{(t)}) = \frac{1}{M} (\mathbf{C} \bar{\mathbf{x}}_k^{(t)} - \frac{(\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)}}{\|\bar{\mathbf{x}}_k^{(t)}\|^2} \bar{\mathbf{x}}_k^{(t)} - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)})$, we have $\mathbf{g}'(\mathbf{x}_{s,k}^*) = \frac{1}{M} (c_k \mathbf{C} \mathbf{q}_k - \frac{c_k \mathbf{q}_k^T \mathbf{C} c_k \mathbf{q}_k}{c_k^2 \|\mathbf{q}_k\|^2} \mathbf{q}_k - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} c_k \mathbf{q}_k) = 0$. Hence,

$$\|\mathbf{g}'(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| = \sqrt{M} \|\mathbf{g}'(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| = \sqrt{M} \|\mathbf{g}'(\bar{\mathbf{x}}_{s,k}^{(t-1)}) - \mathbf{g}'(\mathbf{x}_{s,k}^*)\| \leq L_k \sqrt{M} \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\|.$$

Proceeding exactly as Theorem 7, we get

$$\begin{bmatrix} \|\mathbf{s}_k^{(t)} - \mathbf{g}_s(\mathbf{x}_k^{(t)})\| \\ \|\mathbf{x}_k^{(t)} - \bar{\mathbf{x}}_{s,k}^{(t)}\| \\ \sqrt{M} \|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\| \end{bmatrix} \leq \begin{bmatrix} (\frac{1+\beta}{2} + \alpha L_k) & L_k(2 + \alpha L_k) & \alpha L_k^2 \\ \alpha & \frac{1+\beta}{2} & 0 \\ 0 & \alpha L_k & \delta_k \end{bmatrix} \begin{bmatrix} \|\mathbf{s}_k^{(t-1)} - \mathbf{g}_s(\mathbf{x}_k^{(t-1)})\| \\ \|\mathbf{x}_k^{(t-1)} - \bar{\mathbf{x}}_{s,k}^{(t-1)}\| \\ \sqrt{M} \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| \end{bmatrix} + \|\mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t-1)})\| \begin{bmatrix} \alpha L_k \sqrt{M} \\ 0 \\ \alpha \sqrt{M} \end{bmatrix}.$$

Let us denote

$$\mathbf{P}_k(\alpha) = \begin{bmatrix} (\frac{1+\beta}{2} + \alpha L_k) & L_k(2 + \alpha L_k) & \alpha L_k^2 \\ \alpha & \frac{1+\beta}{2} & 0 \\ 0 & \alpha L_k & \rho_k \end{bmatrix}.$$

Since $\mathbf{P}_k(\alpha)$ has non-negative entries and $\mathbf{P}_k^2(\alpha)$ has all positive entries, each entry of $\mathbf{P}_k^t(\alpha)$ will be $\mathcal{O}(\rho(\mathbf{P}_k(\alpha))^t)$, where $\rho(\mathbf{P}_k(\alpha))$ is the spectral radius of $\mathbf{P}_k(\alpha)$. From Lemma 15 we

know if we choose $\alpha < \frac{\lambda_k - \lambda_{k+1}}{(k+5)(k+6)} (\frac{1-\beta}{9\lambda_1})^2$, then $\rho(\mathbf{P}_k(\alpha)) < 1$. Also, we know $\mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t)}) = \frac{1}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} (\frac{\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T}{\|\mathbf{x}_{i,p}^{(t)}\|^2} - \mathbf{q}_p \mathbf{q}_p^T) \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)}$. Thus,

$$\begin{aligned} \|\mathbf{f}(\bar{\mathbf{x}}_{s,k}^{(t)})\| &= \left\| \frac{1}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} \left(\frac{\mathbf{x}_{i,p}^{(t-1)} (\mathbf{x}_{i,p}^{(t-1)})^T}{\|\mathbf{x}_{i,p}^{(t-1)}\|^2} - \mathbf{q}_p \mathbf{q}_p^T \right) \mathbf{C}_i \bar{\mathbf{x}}_k^{(t-1)} \right\| \\ &\leq \frac{1}{M} \sum_{i=1}^M \left\| \sum_{p=1}^{k-1} \left(\frac{\mathbf{x}_{i,p}^{(t-1)} (\mathbf{x}_{i,p}^{(t-1)})^T}{\|\mathbf{x}_{i,p}^{(t-1)}\|^2} - \mathbf{q}_p \mathbf{q}_p^T \right) \right\| \|\mathbf{C}_i\| \|\bar{\mathbf{x}}_k^{(t-1)}\| \end{aligned}$$

From (4.44), we know $\left\| \sum_{p=1}^{k-1} \left(\frac{\mathbf{x}_{i,p}^{(t-1)} (\mathbf{x}_{i,p}^{(t-1)})^T}{\|\mathbf{x}_{i,p}^{(t-1)}\|^2} - \mathbf{q}_p \mathbf{q}_p^T \right) \right\| \leq b_i \nu_i^t$. Let $b = (\max_i b_i) \lambda_1 \|\bar{\mathbf{x}}_k^{(t-1)}\| > 0$ and $\nu = \max_i \nu_i < 1$. Thus the system of equations becomes

$$\begin{bmatrix} \|\mathbf{s}_k^{(t)} - \mathbf{g}_s(\mathbf{x}_k^{(t)})\| \\ \|\mathbf{x}_k^{(t)} - \bar{\mathbf{x}}_{s,k}^{(t)}\| \\ \sqrt{M} \|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\| \end{bmatrix} \leq \rho(\mathbf{P}_k(\alpha))^t \begin{bmatrix} \|\mathbf{s}_k^{(0)} - \mathbf{g}_s(\mathbf{x}_k^{(0)})\| \\ \|\mathbf{x}_k^{(0)} - \bar{\mathbf{x}}_{s,k}^{(0)}\| \\ \sqrt{M} \|\bar{\mathbf{x}}_k^{(0)} - \mathbf{x}_k^*\| \end{bmatrix} + b \nu^t \begin{bmatrix} \alpha L_k \sqrt{M} \\ 0 \\ \alpha \sqrt{M} \end{bmatrix}.$$

This implies that if $\alpha < \frac{\lambda_k - \lambda_{k+1}}{(k+5)(k+6)} (\frac{1-\beta}{9\lambda_1})^2$, then $\|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\|$, $\|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_{s,k}^{(t)}\|$ and $\|\mathbf{s}_{i,k}^{(t)} - \mathbf{g}(\mathbf{x}_k^{(t)})\|$ converge at a linear rate to 0. In other words, $\mathbf{x}_{i,k}^{(t)}$ converges linearly to $\mathbf{x}_k^* = \pm c_k \mathbf{q}_k$, where c_k is some constant. \square

From Theorem 6 and Theorem 8, we can see that if $\alpha < \frac{\min_k(\lambda_k - \lambda_{k+1})}{(K+5)(K+6)} (\frac{1-\beta}{9\lambda_1})^2$, where λ_1 is the largest eigenvalue of \mathbf{C} , K is the number of eigenvectors to be estimated and β is the absolute value of second largest eigenvalue of the weight matrix \mathbf{W} , then the estimates $\mathbf{x}_{i,k}^{(t)}$ of the k^{th} eigenvector for $k = 1, \dots, K$ at i^{th} node, $i = 1, \dots, M$ converge at a linear rate to a multiple of the eigenvector \mathbf{q}_k of \mathbf{C} i.e., $\pm c_k \mathbf{q}_k$. A one step normalization of each of the estimates at the end gives the set of orthonormal eigenvectors of \mathbf{C} .

In summary, both variants of FAST-PCA converge exactly to the true eigenvectors whilst completely doing away with the need of explicit consensus loop thereby making our solutions faster. Table 4.1 provides a comparison of the communication and iteration complexities of various distributed PCA (principal component analysis) and PSA (principal subspace analysis) algorithms in terms of error ϵ and eigengap gap . Here $gap_r = \frac{\lambda_K + 1}{\lambda_K}$ for PSA and $gap_r = \max_{k=1, \dots, K} \frac{\lambda_k + 1}{\lambda_k}$ for PCA algorithms. Also, $gap = \lambda_K - \lambda_{K+1}$ for PSA algorithms and $gap = \min_{k=1, \dots, K} \lambda_k - \lambda_{k+1}$ for PCA algorithms. Since we reduce the dependence of total iteration complexity on gap , our solutions are significantly faster than other algorithms as also shown through numerical experiments in the Section 4.6

Table 4.1: Comparison of Communication and Iteration Cost

	Comm./Iteration	No. of Iterations	Total Comm.	PCA/PSA
DistSeqPM	$\mathcal{O}(K \frac{1}{\log gap_r^{-1}} \log \frac{1}{\epsilon})$	$\mathcal{O}(K \frac{1}{\log gap_r^{-1}} \log \frac{1}{\epsilon})$	$\mathcal{O}(K^2 \frac{1}{\log^2 gap_r^{-1}} \log^2 \frac{1}{\epsilon})$	PCA
S-DOT	$\mathcal{O}(\frac{1}{\log gap_r^{-1}} \log \frac{1}{\epsilon})$	$\mathcal{O}(\frac{1}{\log gap_r^{-1}} \log \frac{1}{\epsilon})$	$\mathcal{O}(\frac{1}{\log^2 gap_r^{-1}} \log^2 \frac{1}{\epsilon})$	PSA
DeEPCA	$\mathcal{O}(\log \frac{1}{gap})$	$\mathcal{O}(\frac{1}{gap} \log \frac{1}{\epsilon})$	$\mathcal{O}(\frac{1}{gap} \log \frac{1}{gap} \log \frac{1}{\epsilon})$	PSA
DSA	1	$\mathcal{O}(\frac{1}{\log(1+gap)} \log \frac{1}{\epsilon})$ up to $\epsilon = \mathcal{O}(\alpha)$	$\mathcal{O}(\frac{1}{\log(1+gap)} \log \frac{1}{\epsilon})$	PCA
FAST-PCA	2	$\mathcal{O}(\frac{1}{\log(1+gap)} \log \frac{1}{\epsilon})$	$\mathcal{O}(\frac{1}{\log(1+gap)} \log \frac{1}{\epsilon})$	PCA

4.5 Statements and Proofs of Supporting Lemmas

4.5.1 Statement and Proof of Supporting Lemma for Modified Krusulina

The proof of Theorem 4 mainly stands on two lemmas, Lemma 10 and Lemma 11 which are given below.

Lemma 10. Suppose $z_{k,k}^{(0)} \neq 0$ and $\alpha < \frac{1}{\lambda_1}$. Then the following is true for $\gamma_k = \left(\frac{1}{1+\alpha\lambda_k}\right)^2 < 1$ and some constant $a_1 > 0$:

$$\sum_{l=1}^{k-1} (z_{k,l}^{(t+1)})^2 \leq a_1 \gamma_k^{t+1}. \quad (4.48)$$

Proof. For $l = 1, \dots, k-1$ we know from (4.11) that $z_{k,l}^{(t+1)} = a_k^{(t)} (1 - \alpha(\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)}) z_{k,l}^{(t)}$. Since $(\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)} \leq \lambda_1 < \frac{1}{\alpha}$, we have $1 + \alpha(\lambda_k - (\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)}) > \alpha\lambda_k \geq 0$.

Thus, we have for $l = 1, \dots, k-1$,

$$\begin{aligned}
\left(\frac{z_{k,l}^{(t+1)}}{z_{k,k}^{(t+1)}}\right)^2 &= \left(\frac{1 - \alpha(\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)}}{1 + \alpha(\lambda_k - (\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)})}\right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2 \\
&= \left(1 - \frac{\alpha\lambda_k}{1 + \alpha(\lambda_k - (\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)})}\right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2 \\
&\leq \left(1 - \frac{\alpha\lambda_k}{1 + \alpha\lambda_k}\right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2 \\
&= \left(\frac{1}{1 + \alpha\lambda_k}\right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2 \\
&= \gamma_k \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2, \quad \gamma_k = \left(\frac{1}{1 + \alpha\lambda_k}\right)^2 < 1.
\end{aligned}$$

Therefore, for $l = 1, \dots, k-1$, $(z_{k,l}^{(t+1)})^2 \leq \gamma_k^{t+1} \left(\frac{z_{k,l}^{(0)}}{z_{k,k}^{(0)}}\right)^2 (z_{k,k}^{(t+1)})^2$. Since $\|\tilde{\mathbf{x}}_k^{(t+1)}\|^2 = 1$ and $\|\tilde{\mathbf{x}}_k^{(0)}\|^2 = 1$, hence $(z_{k,k}^{(t+1)})^2 \leq 1$ and $z_{k,l}^{(0)} \leq 1$. Also, because of the assumption $z_{k,k}^{(0)} \neq 0$, let us

assume $(z_{k,k}^{(0)})^2 > \tilde{\eta}$. Thus, we can write

$$\sum_{l=1}^{k-1} (z_{k,l}^{t+1})^2 \leq \gamma_k^{t+1} \sum_{l=1}^{k-1} \frac{1}{\tilde{\eta}} = a_1 \gamma_k^{t+1}. \quad (4.49)$$

□

Lemma 11. Suppose $z_{k,k}^{(0)} \neq 0$ and $\alpha < \frac{1}{\lambda_1}$. Then the following is true for $\rho_k = \left(\frac{1+\alpha\lambda_{k+1}}{1+\alpha\lambda_k}\right)^2 < 1$ and some constant $a_2 > 0$:

$$\sum_{l=k+1}^d (z_{k,l}^{t+1})^2 \leq a_2 \rho_k^{t+1}. \quad (4.50)$$

Proof. For $l = k, \dots, d$ we know from (4.11) that $z_{k,l}^{(t+1)} = a_k^{(t)} (1 + \alpha(\lambda_l - (\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)})) z_{k,l}^{(t)}$. Since $(\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)} \leq \lambda_1 < \frac{1}{\alpha}$, we have $1 + \alpha(\lambda_l - (\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)}) > \alpha\lambda_l \geq 0, \forall l = k, \dots, d$.

Thus, we have for $l = k+1, \dots, d$,

$$\begin{aligned} \left(\frac{z_{k,l}^{(t+1)}}{z_{k,k}^{(t+1)}}\right)^2 &= \left(\frac{1 + \alpha(\lambda_l - (\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)})}{1 + \alpha(\lambda_k - (\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)})}\right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2 \\ &= \left(1 - \frac{\alpha(\lambda_k - \lambda_l)}{1 + \alpha(\lambda_k - (\tilde{\mathbf{x}}_{g,k}^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_{g,k}^{(t)})}\right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2 \\ &\leq \left(1 - \frac{\alpha(\lambda_k - \lambda_l)}{1 + \alpha\lambda_k}\right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2 \\ &= \left(\frac{1 + \alpha\lambda_l}{1 + \alpha\lambda_k}\right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2 \leq \left(\frac{1 + \alpha\lambda_{k+1}}{1 + \alpha\lambda_k}\right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2 \\ &= \rho_k \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}}\right)^2, \quad \rho_k = \left(\frac{1 + \alpha\lambda_{k+1}}{1 + \alpha\lambda_k}\right)^2 < 1. \end{aligned}$$

Therefore, for $l = k+1, \dots, d$, $(z_{k,l}^{t+1})^2 \leq \rho_k^{t+1} \left(\frac{z_{k,l}^{(0)}}{z_{k,k}^{(0)}}\right)^2 (z_{k,k}^{t+1})^2$. Since $\|\tilde{\mathbf{x}}_k^{(t+1)}\|^2 = 1$ and $\|\tilde{\mathbf{x}}_k^{(0)}\|^2 = 1$, hence $(z_{k,k}^{t+1})^2 \leq 1$ and $z_{k,l}^{(0)} \leq 1$. Also, since $z_{k,k}^{(0)} \neq 0$, let us assume $(z_{k,k}^{(0)})^2 > \tilde{\eta}$.

Thus, we can write

$$\sum_{l=k+1}^d (z_{k,l}^{t+1})^2 \leq \rho_k^{t+1} \sum_{l=k+1}^d \frac{1}{\tilde{\eta}} = a_2 \rho_k^{t+1}. \quad (4.51)$$

□

4.5.2 Statement and Proof of Supporting Lemma for FAST-PCA

First two lemmas prove the Lipschitz continuity of the pseudo-gradients \mathbf{h}_i^O and \mathbf{h}_i^K

Lemma 12. The function $\mathbf{h}_{i,t}^O : \mathbb{R}^d \rightarrow \mathbb{R}^d$ with $\mathbf{h}_{i,t}^O(\mathbf{v}) = \mathbf{C}_i \mathbf{v} - (\mathbf{v})^T \mathbf{C}_i \mathbf{v} \mathbf{v} - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{v}$ is Lipschitz continuous with Lipschitz constant $L_k = \lambda_1(1 + (k+2)\mu)$ when $\|\mathbf{v}\|^2 \leq \mu$ and $\|\mathbf{x}_{i,p}\|^2 \leq \mu$ for $p = 1, \dots, k-1$.

Proof. For $\mathbf{v} \in \mathbb{R}^d$ such that $\|\mathbf{v}\|^2 \leq \mu$, the function $\mathbf{h}_i^O(\mathbf{v})$ is defined as

$$\mathbf{h}_i^O(\mathbf{v}) = \mathbf{C}_i \mathbf{v} - (\mathbf{v})^T \mathbf{C}_i \mathbf{v} \mathbf{v} - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{v},$$

Thus,

$$\begin{aligned} \mathbf{h}_i^O(\mathbf{v}_1) - \mathbf{h}_i^O(\mathbf{v}_2) &= \mathbf{C}_i(\mathbf{v}_1 - \mathbf{v}_2) - (\mathbf{v}_1)^T \mathbf{C}_i \mathbf{v}_1 \mathbf{v}_1 + (\mathbf{v}_2)^T \mathbf{C}_i \mathbf{v}_2 \mathbf{v}_2 - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i (\mathbf{v}_1 - \mathbf{v}_2) \\ &= (\mathbf{C}_i - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i) (\mathbf{v}_1 - \mathbf{v}_2) - (\mathbf{v}_1)^T \mathbf{C}_i \mathbf{v}_1 (\mathbf{v}_1 - \mathbf{v}_2) - ((\mathbf{v}_1)^T \mathbf{C}_i \mathbf{v}_1 - (\mathbf{v}_2)^T \mathbf{C}_i \mathbf{v}_2) \mathbf{v}_2 \\ &= (\mathbf{C}_i - (\mathbf{v}_1)^T \mathbf{C}_i \mathbf{v}_1 \mathbf{I} - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i) (\mathbf{v}_1 - \mathbf{v}_2) - ((\mathbf{v}_1 + \mathbf{v}_2)^T \mathbf{C}_i (\mathbf{v}_1 - \mathbf{v}_2)) \mathbf{v}_2 \end{aligned}$$

$$\begin{aligned} \|\mathbf{h}_i^O(\mathbf{v}_1) - \mathbf{h}_i^O(\mathbf{v}_2)\| &\leq \|\mathbf{C}_i - (\mathbf{v}_1)^T \mathbf{C}_i \mathbf{v}_1 \mathbf{I} - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i\| \|\mathbf{v}_1 - \mathbf{v}_2\| + \|(\mathbf{v}_1 + \mathbf{v}_2)^T \mathbf{C}_i (\mathbf{v}_1 - \mathbf{v}_2)\| \|\mathbf{v}_2\| \\ &\leq \|\mathbf{C}_i - (\mathbf{v}_1)^T \mathbf{C}_i \mathbf{v}_1 \mathbf{I} - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i\| \|\mathbf{v}_1 - \mathbf{v}_2\| \\ &\quad + \|\mathbf{v}_1 + \mathbf{v}_2\| \|\mathbf{C}_i\| \|\mathbf{v}_1 - \mathbf{v}_2\| \|\mathbf{v}_2\| \\ &\leq \|\mathbf{C}_i - (\mathbf{v}_1)^T \mathbf{C}_i \mathbf{v}_1 \mathbf{I} - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i\| \|\mathbf{v}_1 - \mathbf{v}_2\| \\ &\quad + \|\mathbf{v}_2\| (\|\mathbf{v}_1\| + \|\mathbf{v}_2\|) \|\mathbf{C}_i\| \|\mathbf{v}_1 - \mathbf{v}_2\| \\ &\leq (\|\mathbf{C}_i\| + |(\mathbf{v}_1)^T \mathbf{C}_i \mathbf{v}_1| + \sum_{p=1}^{k-1} \|\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i\|) \|\mathbf{v}_1 - \mathbf{v}_2\| \\ &\quad + \|\mathbf{v}_2\| (\|\mathbf{v}_1\| + \|\mathbf{v}_2\|) \|\mathbf{C}_i\| \|\mathbf{v}_1 - \mathbf{v}_2\| \\ &\leq (\lambda_1 + \lambda_1 \|\mathbf{v}_1\|^2 + (k-1) \lambda_1 \|\mathbf{x}_{i,p}^{(t)}\|^2) \|\mathbf{v}_1 - \mathbf{v}_2\| + \lambda_1 \sqrt{\mu} (\sqrt{\mu} + \sqrt{\mu}) \|\mathbf{v}_1 - \mathbf{v}_2\| \\ &\leq (\lambda_1 + \lambda_1 \mu + (k-1) \lambda_1 \mu) \|\mathbf{v}_1 - \mathbf{v}_2\| + 2 \lambda_1 \mu \|\mathbf{v}_1 - \mathbf{v}_2\| \\ &\leq (\lambda_1 + k \lambda_1 \mu + 2 \lambda_1 \mu) \|\mathbf{v}_1 - \mathbf{v}_2\| \\ &\leq \lambda_1 (1 + (k+2) \mu) \|\mathbf{v}_1 - \mathbf{v}_2\| \end{aligned}$$

□

Lemma 13. The function $\mathbf{h}_{i,t}^K : \mathbb{R}^d \rightarrow \mathbb{R}^d$ with $\mathbf{h}_{i,t}^K(\mathbf{v}) = \mathbf{C}_i \mathbf{v} - \frac{(\mathbf{v})^T \mathbf{C}_i \mathbf{v}}{\|\mathbf{v}\|^2} \mathbf{v} - \sum_{p=1}^{k-1} \frac{\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T}{\|\mathbf{x}_{i,p}^{(t)}\|^2} \mathbf{C}_i \mathbf{v}$ is Lipschitz continuous with Lipschitz constant $L_k = \lambda_1(k+5)$.

Proof. For a continuous function $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$, we know $\|f(x) - f(y)\| \leq \|\nabla f(x)\| \|x - y\|$.

Thus, the Lipschitz constant can be given by the upper bound of $\|\nabla f(x)\|$. Applying derivative

on both sides to the function

$$\mathbf{h}_{i,t}^K(\mathbf{v}) = \mathbf{C}_i \mathbf{v} - \frac{(\mathbf{v})^T \mathbf{C}_i \mathbf{v}}{\|\mathbf{v}\|^2} \mathbf{v} - \sum_{p=1}^{k-1} \frac{\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T}{\|\mathbf{x}_{i,p}^{(t)}\|^2} \mathbf{C}_i \mathbf{v}$$

we get

$$\begin{aligned} \frac{\partial}{\partial \mathbf{v}} \mathbf{h}_{i,t}^K(\mathbf{v}) &= \mathbf{C}_i - \frac{\partial}{\partial \mathbf{v}} \left(\frac{(\mathbf{v})^T \mathbf{C}_i \mathbf{v}}{\|\mathbf{v}\|^2} \right) \mathbf{v}^T - \frac{(\mathbf{v})^T \mathbf{C}_i \mathbf{v}}{\|\mathbf{v}\|^2} \mathbf{I} - \left(\sum_{p=1}^{k-1} \frac{\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T}{\|\mathbf{x}_{i,p}^{(t)}\|^2} \mathbf{C}_i \right)^T \\ &= \mathbf{C}_i - \frac{2\|\mathbf{v}\|^2 \mathbf{C}_i \mathbf{v} - 2(\mathbf{v})^T \mathbf{C}_i \mathbf{v} \mathbf{v}}{\|\mathbf{v}\|^4} \mathbf{v}^T - \frac{(\mathbf{v})^T \mathbf{C}_i \mathbf{v}}{\|\mathbf{v}\|^2} \mathbf{I} - \left(\sum_{p=1}^{k-1} \frac{\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T}{\|\mathbf{x}_{i,p}^{(t)}\|^2} \mathbf{C}_i \right)^T \\ &= \mathbf{C}_i - \frac{2\mathbf{C}_i \mathbf{v} \mathbf{v}^T}{\|\mathbf{v}\|^2} + \frac{2(\mathbf{v})^T \mathbf{C}_i \mathbf{v} \mathbf{v} \mathbf{v}^T}{\|\mathbf{v}\|^4} - \frac{(\mathbf{v})^T \mathbf{C}_i \mathbf{v}}{\|\mathbf{v}\|^2} \mathbf{I} - \sum_{p=1}^{k-1} \mathbf{C}_i \frac{\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T}{\|\mathbf{x}_{i,p}^{(t)}\|^2} \\ \left\| \frac{\partial}{\partial \mathbf{v}} \mathbf{h}_{i,t}(\mathbf{v}) \right\| &\leq \|\mathbf{C}_i\| + \left\| \frac{2\mathbf{C}_i \mathbf{v} \mathbf{v}^T}{\|\mathbf{v}\|^2} \right\| + \left\| \frac{2(\mathbf{v})^T \mathbf{C}_i \mathbf{v} \mathbf{v} \mathbf{v}^T}{\|\mathbf{v}\|^4} \right\| + \left\| \frac{(\mathbf{v})^T \mathbf{C}_i \mathbf{v}}{\|\mathbf{v}\|^2} \mathbf{I} \right\| + \sum_{p=1}^{k-1} \left\| \mathbf{C}_i \frac{\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T}{\|\mathbf{x}_{i,p}^{(t)}\|^2} \right\| \\ &\leq \lambda_{i,1} + 2 \frac{\lambda_{i,1} \|\mathbf{v}\|^2}{\|\mathbf{v}\|^2} + 2 |(\mathbf{v})^T \mathbf{C}_i \mathbf{v}| \frac{\|\mathbf{v}\|^2}{\|\mathbf{v}\|^4} + \frac{|(\mathbf{v})^T \mathbf{C}_i \mathbf{v}|}{\|\mathbf{v}\|^2} + \sum_{p=1}^{k-1} \left\| \mathbf{C}_i \right\| \frac{\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T}{\|\mathbf{x}_{i,p}^{(t)}\|^2} \\ &\leq \lambda_{i,1} + 2\lambda_{i,1} + 2\lambda_{i,1} + \lambda_{i,1} + \lambda_{i,1}(k-1) \\ &= (k+5)\lambda_{i,1} \end{aligned}$$

Thus,

$$\|\mathbf{h}_{i,t}^K(\mathbf{v}_1) - \mathbf{h}_{i,t}^K(\mathbf{v}_2)\| \leq \lambda_{i,1}(k+5)\|\mathbf{v}_1 - \mathbf{v}_2\| \leq \lambda_1(k+5)\|\mathbf{v}_1 - \mathbf{v}_2\|, \quad (4.52)$$

where the last inequality uses the fact that $\mathbf{C}_i \preceq \mathbf{C}$, hence $\lambda_{i,1} \leq \lambda_1$. \square

Lemma 14. *The following inequalities hold true:*

1. $\|\mathbf{g}(\mathbf{x}_k^{(t)}) - \mathbf{g}(\mathbf{x}_k^{(t-1)})\|_2 \leq \frac{L_k}{\sqrt{M}} \|\mathbf{x}_k^{(t)} - \mathbf{x}_k^{(t-1)}\|_2$
2. $\|\mathbf{g}(\mathbf{x}_k^{(t)}) - \mathbf{g}(\bar{\mathbf{x}}_{s,k}^{(t)})\|_2 \leq \frac{L_k}{\sqrt{M}} \|\mathbf{x}_k^{(t)} - \bar{\mathbf{x}}_{s,k}^{(t)}\|_2,$

where $L_k = \lambda_1(1 + (k+2)\mu)$ when $\mathbf{g}(\mathbf{x}_k^{(t)}) = \frac{1}{M} \sum_{i=1}^M \mathbf{h}_i^O(\mathbf{x}_{i,k}^{(t)})$ and $L_k = \lambda_1(k+5)$ when $\mathbf{g}(\mathbf{x}_k^{(t)}) = \frac{1}{M} \sum_{i=1}^M \mathbf{h}_i^K(\mathbf{x}_{i,k}^{(t)})$.

Proof. 1.

$$\begin{aligned}
\|\mathbf{g}(\mathbf{x}_k^{(t)}) - \mathbf{g}(\mathbf{x}_k^{(t-1)})\|_2^2 &= \frac{1}{M^2} \left\| \sum_{i=1}^M (\mathbf{h}_i^{O/K}(\mathbf{x}_{i,k}^{(t)}) - \mathbf{h}_i^{O/K}(\mathbf{x}_{i,k}^{(t-1)})) \right\|^2 \\
&\leq \frac{1}{M^2} M \sum_{i=1}^M \|\mathbf{h}_i^{O/K}(\mathbf{x}_{i,k}^{(t)}) - \mathbf{h}_i^{O/K}(\mathbf{x}_{i,k}^{(t-1)})\|^2, \quad \text{using the fact } \left\| \sum_{i=1}^M \mathbf{a}_i \right\|^2 \leq M \sum_{i=1}^M \|\mathbf{a}_i\|^2 \\
&\leq \frac{L_k^2}{M} \sum_{i=1}^M \|\mathbf{x}_{i,k}^{(t)} - \mathbf{x}_{i,k}^{(t-1)}\|^2 \\
&= \frac{L_k^2}{M} \|\mathbf{x}_k^{(t)} - \mathbf{x}_k^{(t-1)}\|^2 \\
\|\mathbf{g}(\mathbf{x}_k^{(t)}) - \mathbf{g}(\mathbf{x}_k^{(t-1)})\|_2 &\leq \frac{L_k}{\sqrt{M}} \|\mathbf{x}_k^{(t)} - \mathbf{x}_k^{(t-1)}\|_2
\end{aligned}$$

2.

$$\begin{aligned}
\|\mathbf{g}(\mathbf{x}_k^{(t)}) - \mathbf{g}(\bar{\mathbf{x}}_{s,k}^{(t)})\|_2^2 &= \frac{1}{M^2} \left\| \sum_{i=1}^M (\mathbf{h}_i^{O/K}(\mathbf{x}_{i,k}^{(t)}) - \mathbf{h}_i^{O/K}(\bar{\mathbf{x}}_k^{(t)})) \right\|^2 \\
&\leq \frac{1}{M^2} M \sum_{i=1}^M \|\mathbf{h}_i^{O/K}(\mathbf{x}_{i,k}^{(t)}) - \mathbf{h}_i^{O/K}(\bar{\mathbf{x}}_k^{(t)})\|^2, \quad \text{using the fact } \left\| \sum_{i=1}^M \mathbf{a}_i \right\|^2 \leq M \sum_{i=1}^M \|\mathbf{a}_i\|^2 \\
&\leq \frac{L_k^2}{M} \sum_{i=1}^M \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\|^2 \\
&= \frac{L_k^2}{M} \|\mathbf{x}_k^{(t)} - \bar{\mathbf{x}}_{s,k}^{(t)}\|^2 \\
\|\mathbf{g}(\mathbf{x}_k^{(t)}) - \mathbf{g}(\bar{\mathbf{x}}_{s,k}^{(t)})\|_2 &\leq \frac{L_k}{\sqrt{M}} \|\mathbf{x}_k^{(t)} - \bar{\mathbf{x}}_{s,k}^{(t)}\|_2
\end{aligned}$$

□

Finally, the following lemma gives the condition of step size required for linear convergence of FAST-PCA.

Lemma 15. *For a matrix $\mathbf{P}_k(\alpha)$ such that*

$$\mathbf{P}(\alpha) = \begin{bmatrix} (\frac{1+\beta}{2} + \alpha L_k) & L_k(2 + \alpha L_k) & \alpha L_k^2 \\ \alpha & \frac{1+\beta}{2} & 0 \\ 0 & \alpha L_k & \delta_k \end{bmatrix}$$

where $L_k = (k+5)\lambda_1$ and $\delta_k = \frac{1+\alpha\lambda_{k+1}}{1+\alpha\lambda_k}$, the spectral radius $\rho(\mathbf{P}_k(\alpha))$ is strictly less than 1 if $\alpha < \frac{\lambda_1 - \lambda_2}{42} \left(\frac{1-\beta}{9\lambda_1} \right)^2$.

Proof. Since $\mathbf{P}_k(\alpha)$ is a non-negative matrix, by Perron-Frobenius theorem it's characteristic polynomial has a simple positive real root r such that $\rho(\mathbf{P}_k(\alpha)) = r$. We know, $\delta_k = \frac{1+\alpha\lambda_{k+1}}{1+\alpha\lambda_k} <$

1, $L_k = (k+5)\lambda_1$. Now, the characteristic polynomial $p(\gamma)$ of $\mathbf{P}(\alpha)$ is given as

$$\begin{aligned}
p(\gamma) &= |\gamma \mathbf{I} - \mathbf{P}(\alpha)| \\
&= \begin{vmatrix} \gamma - (\frac{1+\beta}{2} + \alpha L_k) & -L_k(\alpha L_k + 2) & -\alpha L_k^2 \\ -\alpha & \gamma - \frac{1+\beta}{2} & 0 \\ 0 & -\alpha L_k & \gamma - \delta_1 \end{vmatrix} \\
&= (\gamma - \frac{1+\beta}{2} - \alpha L_k)(\gamma - \frac{1+\beta}{2})(\gamma - \delta_k) + \alpha(-L_k(\alpha L_k + 2)(\gamma - \delta_k) - \alpha^2 L_k^3) \\
&= ((\gamma - \frac{1+\beta}{2} - \alpha L_k)(\gamma - \frac{1+\beta}{2}) - \alpha L_k(\alpha L_k + 2))(\gamma - \delta_k) - \alpha^3 L_k^3 \\
&= p_0(\gamma)(\gamma - \delta_k) - \alpha^3 L_k^3
\end{aligned}$$

where,

$$\begin{aligned}
p_0(\gamma) &= (\gamma - \frac{1+\beta}{2} - \alpha L_k)(\gamma - \frac{1+\beta}{2}) - \alpha L_k(\alpha L_k + 2) \\
&= \gamma^2 - (1 + \beta + \alpha L_k)\gamma + \frac{1+\beta}{2}(\frac{1+\beta}{2} + \alpha L_k) - \alpha L_k(\alpha L_k + 2) \\
&= (\gamma - \gamma_1)(\gamma - \gamma_2),
\end{aligned}$$

where γ_1, γ_2 are the roots of $p_0(\gamma)$, given as

$$\gamma_{1,2} = \frac{1 + \beta + \alpha L_k \pm \sqrt{5\alpha^2 L_k^2 + 8\alpha L_k}}{2}. \quad (4.53)$$

If $0 < \alpha < \frac{1}{L_k}$, $\alpha L_k < 1$ and it implies $\alpha^2 L_k^2 < \alpha L_k < \sqrt{\alpha L_k}$. Thus,

$$\begin{aligned}
\gamma_{1,2} &= \frac{1 + \beta + \alpha L_k \pm \sqrt{5\alpha^2 L_k^2 + 8\alpha L_k}}{2} \\
&< \frac{1 + \beta + \alpha L_k + \sqrt{5\alpha^2 L_k^2 + 8\alpha L_k}}{2} \\
&< \frac{1 + \beta + \sqrt{\alpha L_k} + \sqrt{5\alpha L_k + 8\alpha L_k}}{2} \\
&< \frac{1 + \beta + \sqrt{\alpha L_k} + \sqrt{16\alpha L_k}}{2} \\
&= \frac{1 + \beta + 5\sqrt{\alpha L_k}}{2} = \gamma_0
\end{aligned}$$

For $\gamma \geq \gamma_0$, $p_0(\gamma) \geq (\gamma - \gamma_0)^2$. Now, let $\gamma^* = \max\{1 - \frac{\alpha(\lambda_k - \lambda_{k+1})}{1 + \alpha\lambda_k}, \frac{1+\beta}{2} + 4.5\sqrt{\alpha L_k} \sqrt{\frac{(1 + \alpha\lambda_k)L_k}{\lambda_k - \lambda_{k+1}}}\} > \gamma_0$. Then

$$\begin{aligned}
p(\gamma^*) &\geq \frac{1}{2} \frac{\alpha(\lambda_k - \lambda_{k+1})}{1 + \alpha\lambda_k} (4.5\sqrt{\alpha L_k} \sqrt{\frac{(1 + \alpha\lambda_k)L_k}{\lambda_k - \lambda_{k+1}}} - 2.5\sqrt{\alpha L_k})^2 - \alpha^3 L_k^3 \\
&\geq \frac{1}{2} \frac{\alpha(\lambda_k - \lambda_{k+1})}{1 + \alpha\lambda_{k+1}} (4.5\sqrt{\alpha L_k} \sqrt{\frac{(1 + \alpha\lambda_k)L_k}{\lambda_k - \lambda_{k+1}}} - 2.5\sqrt{\alpha L_k} \sqrt{\frac{(1 + \alpha\lambda_k)L_k}{\lambda_k - \lambda_{k+1}}})^2 - \alpha^3 L_k^3 \\
&\geq \frac{1}{2} \frac{\alpha(\lambda_k - \lambda_{k+1})}{1 + \alpha\lambda_k} (2\sqrt{\alpha L_k} \sqrt{\frac{(1 + \alpha\lambda_k)L_k}{\lambda_k - \lambda_{k+1}}})^2 - \alpha^3 L_k^3 \\
&= 2\alpha^2 L_k^2 - \alpha^3 L_k^3 \geq 0.
\end{aligned}$$

Evidently, $p(\gamma)$ is a strictly increasing function on $[\max\{\delta_k, \gamma_0\}, +\infty)$ and since this interval includes γ^* , $p(\gamma)$ has no real roots on $(\gamma^*, +\infty)$. Thus, the real root of the characteristic polynomial is $\leq \gamma^*$. Hence $\rho(\mathbf{P}_k(\alpha)) \leq \gamma^*$. If we choose α such that $\gamma^* < 1$, then $\rho(\mathbf{P}_k(\alpha)) < 1$ and the convergence would be linear. For $\gamma^* < 1$, we need

$$\begin{aligned} \frac{1+\beta}{2} + 4.5\sqrt{\alpha L_k} \sqrt{\frac{(1+\alpha\lambda_k)L_k}{\lambda_k - \lambda_{k+1}}} &< 1 \\ \sqrt{\alpha L_k} \sqrt{\frac{(1+\alpha\lambda_k)L_k}{\lambda_k - \lambda_{k+1}}} &< \frac{1-\beta}{9} \\ (\alpha L_k) \frac{(1+\alpha\lambda_k)L_k}{\lambda_k - \lambda_{k+1}} &< \left(\frac{1-\beta}{9}\right)^2 \\ \alpha(1+\alpha\lambda_k) &< \frac{\lambda_k - \lambda_{k+1}}{L_k^2} \left(\frac{1-\beta}{9}\right)^2 \end{aligned}$$

Since $L_k > \lambda_k$, for $\alpha < \frac{1}{L_k}$, $\alpha\lambda_k \leq 1$. Thus, $1 + \alpha\lambda_k \leq 2$. If $\alpha \leq \frac{\lambda_k - \lambda_{k+1}}{2L_k^2} \left(\frac{1-\beta}{9}\right)^2 < \frac{1}{L_k}$, then $\alpha(1 + \alpha\lambda_k) \leq 2\alpha \leq \frac{\lambda_k - \lambda_{k+1}}{L_k^2} \left(\frac{1-\beta}{9}\right)^2$. Thus, for the following condition on step size α

$$\begin{aligned} \text{FAST-PCA-O : } \quad \alpha &\leq \frac{\lambda_k - \lambda_{k+1}}{2\lambda_1^2(1 + 2\mu + k\mu)^2} \left(\frac{1-\beta}{9}\right)^2 \\ \text{FAST-PCA-K : } \quad \alpha &\leq \frac{\lambda_k - \lambda_{k+1}}{2\lambda_1^2(k+5)^2} \left(\frac{1-\beta}{9}\right)^2, \end{aligned}$$

both versions of FAST-PCA converge at a linear rate. \square

4.6 Experimental Results

In this section, we demonstrate the efficacy of our proposed algorithm FAST-PCA through experiments on synthetic as well as real-world data. We compare the performance of our algorithm with existing algorithms of (centralized) orthogonal iteration (OI), (centralized) sequential power method (SeqPM), distributed sequential power method (SeqDistPM), distributed orthogonal iteration algorithms (S-DOT, SA-DOT) [47], a orthogonal iteration+gradient tracking based method DeEPCA [53] and our previously proposed distributed Sanger's algorithm (DSA) [16]. In the case of OI and SeqPM, we assume that all the samples are available at a single location and, for the estimation of K dominant eigenvectors of \mathbf{C} , SeqPM performs power method K times sequentially starting from the most dominant eigenvector. SeqDistPM is the distributed version of SeqPM, which uses an explicit consensus loop with a fixed number T_c of consensus iterations per iteration of the power iteration [49, 50], whereas S-DOT and SA-DOT are distributed versions of OI using fixed and increasing number of consensus iterations per orthogonal iteration. The DSA is a distributed generalized Hebbian algorithm that converges linearly to a neighborhood of the true eigenvectors of the global covariance matrix. Assuming

that the cost of communicating $\mathbb{R}^{d \times K}$ matrices across the network in one (outer loop) iteration is one unit, the x-axis of all the plots indicate the total communication cost, i.e., total inner and outer loop communications. In the algorithms with one time scale, this is the same as the number of total outer loop iterations (since inner iterations = 0). The y-axis of the plots denotes the average angle between the estimated eigenvectors $\mathbf{x}_{i,k}^{(t)}$ and the true eigenvectors $\pm \mathbf{q}_k$ across all the M nodes in the network given by

$$\mathcal{E} = \frac{1}{MK} \sum_{i=1}^M \sum_{k=1}^K \left(1 - \left(\frac{\mathbf{x}_{i,k}^T \mathbf{q}_k}{\|\mathbf{x}_{i,k}\|} \right)^2 \right). \quad (4.54)$$

4.6.1 Synthetic Data

We study the effect of factors like eigengap and distinct/repeated eigenvalues on the performance of our algorithm in comparison to various other existing PCA and distributed PCA algorithms. To that end, we generate Erdos-Renyi graphs ($p = 0.5$) to simulate the distributed setup with $M = 20$ nodes. We also generate synthetic data with different eigengaps of $\Delta_K = \frac{\lambda_{K+1}}{\lambda_K} \in \{0.8, 0.97\}$. The data is generated such that each node has 5000 i.i.d samples, i.e. $N_i = 5000$ with $d = 20$ drawn from a multivariate Gaussian distribution with zero mean and fixed covariance matrix Σ . The number of eigenvectors to be estimated is set to $K = 5$. For SeqPM, SeqDistPM and S-DOT, the number of consensus iterations per outer loop iteration is $T_c = 50$ and the number of maximum consensus iterations in case of SA-DOT is set to 50 as well. For the Erdos-Renyi topology, we use a step size of $\alpha = 0.7$ for our algorithm and for cyclic graph, we use $\alpha = 0.1$. The results reported are an average of 10 Monte-Carlo simulations. Figure 4.1 compares the performance of our proposed algorithm FAST-PCA with centralized OI, SeqPM, SeqDistPM, S-DOT, SA-DOT, DeEPCA and DSA when the subspace eigenvalues $\lambda_1, \dots, \lambda_K$ are distinct, i.e. $\lambda_1 > \lambda_2 > \dots, > \lambda_K$. It is clear that our algorithm significantly outperforms SeqPM and SeqDistPM since estimating one eigenvector at a time slows down the convergence of these methods. Also, the requirement of an explicit consensus loop implies the communication cost of these methods is high as indicated by the plots. Even though S-DOT and SA-DOT estimate the whole subspace (but not necessarily the eigenvectors) simultaneously, explicit consensus loop makes those relatively slow as well. As expected, since DSA converges only to a neighborhood of the true solutions, our new proposed algorithm outperforms it. The performance of FAST-PCA is similar to DeEPCA in this case, although DeEPCA requires explicit QR normalization after every iteration whereas FAST-PCA requires no explicit normalization. This normalization step in DeEPCA requires an additional $\mathcal{O}(K^2 d)$ computations per iterations. It is desired from any distributed algorithm to perform similar to their centralized counterparts and it is clear from

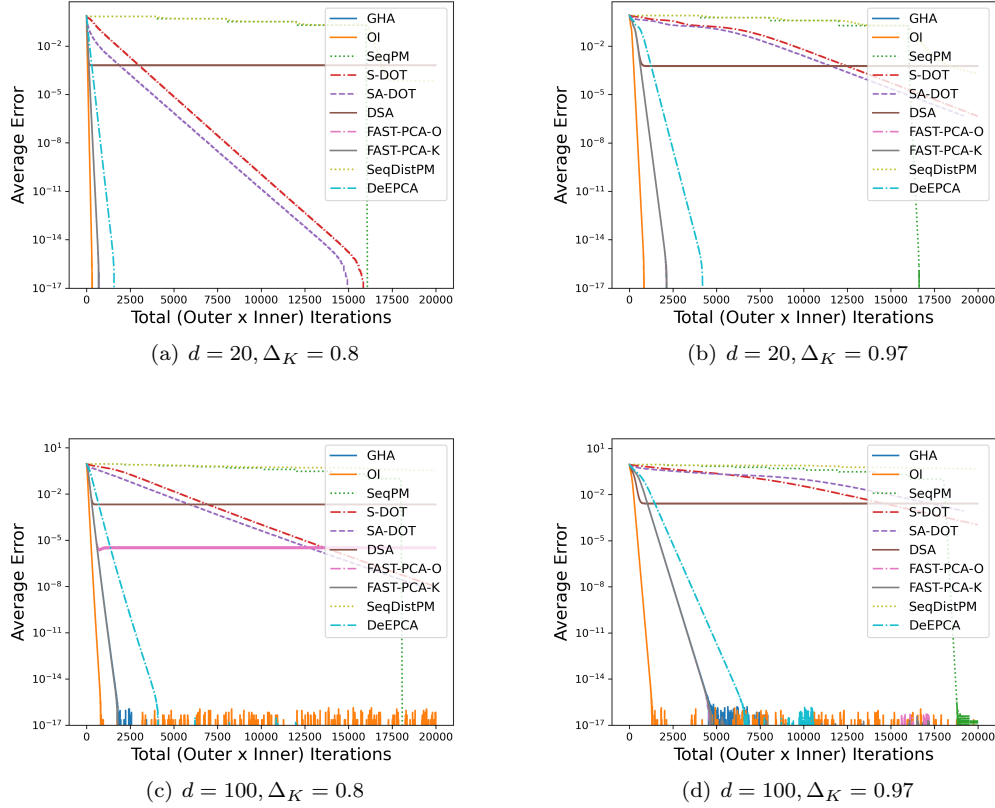


Figure 4.1: Performance comparison of FAST-PCA with various algorithms for two different eigengaps.

the figures that our algorithm FAST-PCA performs very similar to centralized OI.

Figure 4.2 shows a similar performance comparison for $K = 3$ when the subspace eigenvalues are very close to each other, i.e. $\lambda_1 \approx \lambda_2 \approx \dots \approx \lambda_K$. The Gaussian distribution generated in this case has covariance matrix Σ with equal subspace eigenvalues but due to the finite number of samples, the eigenvalues of \mathbf{C} are not exactly equal albeit almost equal. It is evident that the performance of every algorithm significantly deprecates in this scenario. Nonetheless, in this case FAST-PCA outperforms all other algorithms including DeEPCA, while still being close to centralized OI in terms of performance.

4.6.2 Real-World Data

We also provide some results for the real-world datasets of MNIST [71] and CIFAR10 [72]. In the first subsection, we assume that data is randomly but equally (in terms of number of samples) among the nodes. In the next subsection, we show the effect of data being distributed by category on the performance of the algorithms.

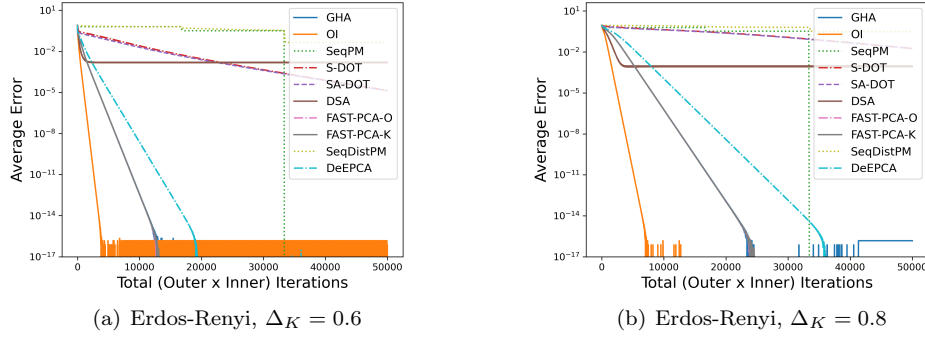


Figure 4.2: Performance comparison of FAST-PCA with various algorithms for two different eigengaps and two graph topologies in the case of (almost) equal eigenvalues.

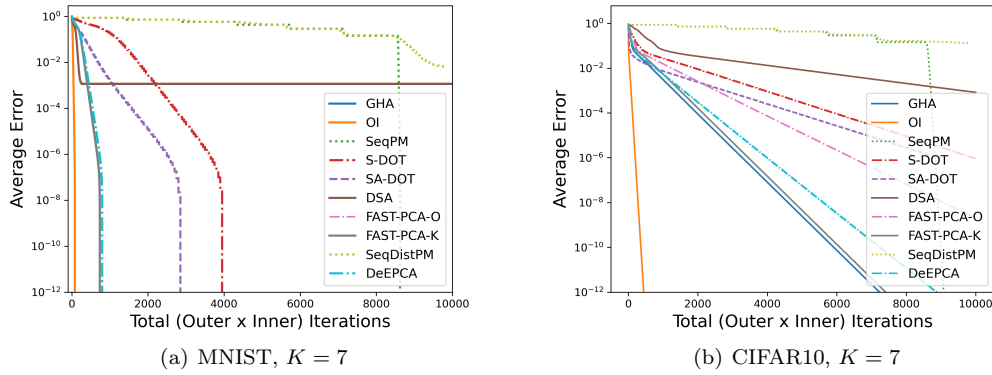


Figure 4.3: Performance comparison of FAST-PCA with various algorithms for MNIST and CIFAR10

Even Distribution of Data

We simulate the distributed setup with an Erdos-Renyi graph with $p = 0.5$ and $M = 20$ nodes. Both these datasets have $N = 60,000$ samples distributed equally among the nodes, making $N_i = 3000$. The data dimensions are $d = 784$ for MNIST and $d = 1024$ for CIFAR10. Figure 4.3(a) shows the comparison of the various PCA algorithms for MNIST dataset when $K = 10$ dominant eigenvectors are estimated. The step-size used for FAST-PCA and DSA in this case is $\alpha = 0.1$. Similar results for CIFAR10 are shown in figure 4.3(b) when $K = 5$ and $\alpha = 0.8$ is used.

Uneven Distribution of Data

Next, we investigate the case when data is not distributed randomly among the nodes. Specifically, we look into the case when data samples corresponding to only one class are available at

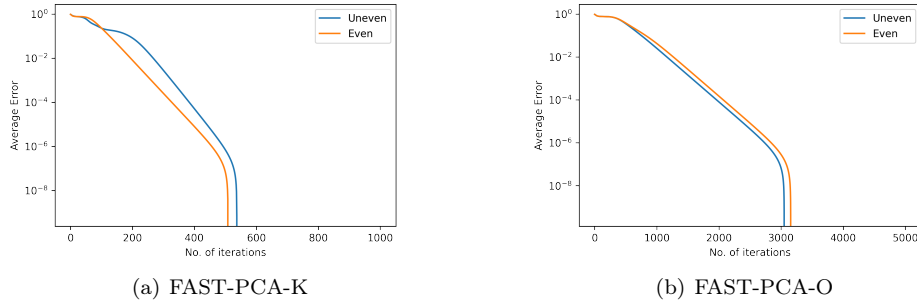


Figure 4.4: Performance comparison of FAST-PCA-O/K for even and uneven distribution of MNIST data

a node. MNIST is a dataset of hand written digits from 0-9. We simulate a network of $M = 10$ nodes such that the i^{th} node has only data samples corresponding to label $(i - 1)$.

4.6.3 Autoencoder Implementation

A single layer autoencoder that has linear activation units when trained using Oja’s rule is known to learn dimension reduced and uncorrelated representations. Our proposed algorithm FAST-PCA is a feedforward neural network-based one time-scale method for representation learning in distributed networks. To that end, we implemented FAST-PCA-O (Oja’s version) in Tensorflow v2 [75, 76] to learn representations for Fashion-MNIST [77] dataset and use those representations for classification.

Fashion-MNIST [77] is a dataset of 28×28 gray-scales images, with each image being associated with one of the 10 labels. There are 60,000 training images and 10,000 testing images. We used tensorflow models to build a classifier that has an input layer (specified later for each experiment) followed by a dense layer with certain units with a ReLU activation function and an output layer of 10 units. First, we train the classifier in the centralized setting i.e., when the entire data (60,000 samples) is available at a single location, using the entire (raw) feature vectors of the samples i.e., when the input layer is of size 784. Tensorflow, by default, does a mini-batch training of the classifier. Using an ‘Adam’ optimizer and ‘Sparse Categorical Cross entropy’ loss function, the average test accuracy that we get for 20 trials is 88.25%. This would be our baseline. Next, we use generalized Hebbian rule in the centralized setting to learn feature vectors of different dimensions i.e., we train autoencoders with different number of hidden units and use these learned feature vectors in the classifiers. Now, in the distributed batch setting that we are considering in this chapter, let us assume that the data is distributed between $M = 10$ nodes i.e., each node has only 6000 samples. Training autoencoders at each node whose weights

Table 4.2: Comparison of Classification accuracy

Features used	Raw	GHA	FAST-PCA
i/p dim. (hidden layer)	784 (256)	256 (128)	256 (128)
accuracy	88.25	88.8	88.8

are updated using our FAST-PCA algorithm, we learn feature representations of length 256. Table 4.2 shows the comparison of classification accuracy when using the raw data vs the learned representation. Evidently, learning the feature vector using FAST-PCA gives the same classification accuracy as in the centralized case. Further, not only is the classification accuracy is slightly better when uncorrelated features are learned, the classifier network required is also smaller implying lesser trainable parameters are required. If one classifier is trained per node using the smaller batch of samples available at the node, then definitely classification accuracy drops. In that case, average accuracy using the raw data batch of 6000 samples is 83.82% and using the learned features is 84.62%.

4.7 Conclusion

In this chapter , we proposed and analyzed two versions of a novel algorithm for distributed Principal Component Analysis (PCA) that truly serves the complete purpose of dimension reduction and uncorrelated feature learning in the scenario where data samples are distributed across a network. We provided detailed theoretical analysis to prove that our proposed algorithm converges linearly, exactly and globally, i.e., starting from any random unit vectors, to the eigenvectors of the global covariance matrix. We also provided experimental results that further validate our claims and demonstrate the communication efficiency and overall effectiveness of our solution.

Chapter 5

DIEGO: Distributed PCA in Streaming Data Settings

The focus so far in this thesis has been on distributed PCA when data is fixed at every node of the network. This chapter considers a more real-world setting: the case of streaming data. Distributed PCA in the streaming setting is a relatively under-studied problem. A distributed algorithm based on the generalized Hebbian algorithm is proposed in this chapter that converges to the eigenvectors of the population covariance matrix in the streaming data case.

5.1 Introduction

One of the defining traits of modern world information is the “velocity” at which data is being generated. A recent study showed that about 2.5 quintillion bytes of new data is created everyday. Thus, to truly encapsulate all aspects of this modern data the continuous streaming nature of data has to be taken into account, along with high dimension and large volume. Machine learning algorithms are typically built on the concept of *continuous learning* from the streaming data to perpetually improve their generalization (to unseen data) power. Making this data usable for machine learning based applications require representation learning solutions that continuously adapt based on new data. Examples of applications where one can find such streaming data are financial trading, video surveillance, autonomous vehicles etc. At the same time, the inherent distribution of data across geographical locations demand the study of distributed solutions even in streaming settings. In the light of such scenarios, we study distributed PCA solutions in streaming settings in this chapter .

Several stochastic methods exist in literature that were specifically developed for PCA in streaming settings like Oja’s rule [12], Sanger’s method [13], Krasulina [21], Warmuth [78]. Oja’s rule and its generalization, Sanger’s method, are in particular popular for estimation of eigenvectors of population covariance matrix due to their ease and stability of implementation as well as convergence guarantees. Recently, Oja’s method [31] have also been studied and analyzed for finite sample complexities of these algorithms. In particular, it was shown that Oja’s method has a optimal rate of $\mathcal{O}(1/t)$ for centralized PCA in streaming data case for the estimation of top eigenvector. This was generalized for the case of subspace estimation in [35] which also proved an optimal rate $\mathcal{O}(1/t)$. In this chapter we propose a distributed version of Oja’s algorithm for the estimation of multiple eigenvectors, not only subspace, in streaming data case.

5.1.1 Our Contributions

The main contributions of this chapter are 1) an algorithm called *DIstributEd Generalized Oja’s Method* (DIEGO) for distributed PCA in the streaming data case, 2) theoretical analysis that proves convergence for the estimation of the dominant eigenvector ($K = 1$) 3) experimental results that demonstrate the efficiency of our solution.

Our primary focus in this chapter is to develop a solution for distributed PCA when the data is streaming at the nodes. The goal remains in line with the previous chapters; we aim for dimension reduction and feature decorrelation at the same time. That is, we aim to find the eigenvectors of the population covariance matrix that can result in a representation of data which retains maximum information while having uncorrelated features (in expectation). To that end, we propose a distributed algorithm that is based on the generalization of Oja’s rule for multiple eigenvector estimation. The algorithm is a one-time scale combine-and-update approach. We provide detailed analysis for the estimation of top eigenvector from independent and identical (i.i.d) samples in the distributed case. Specifically, we show that for the case of $K = 1$, the algorithm converges asymptotically as $t \rightarrow \infty$. We also provide extensive numerical experiments to show the effect of different parameters like eigengap, graph connectivity etc. on the performance of the algorithm.

5.2 Problem Description

The goal of this chapter is to develop an algorithm for finding the leading eigenvectors (principal components) of the population covariance matrix when data samples are streaming in real time.

In a centralized case, when all the independent and identically distributed (i.i.d) data points $\mathbf{y} \in \mathbb{R}^d$ from a zero-mean distribution with covariance matrix $\mathbf{\Sigma}$ arrive at a single location, PCA can be mathematically formulated as

$$\mathbf{X} = \arg \min_{\mathbf{X} \in \mathbb{R}^{d \times K}, \mathbf{X}^T \mathbf{X} = \mathbf{I}} \mathbb{E} \left[\|\mathbf{y} - \mathbf{X} \mathbf{X}^T \mathbf{y}\|_2^2 \right] \quad \text{such that} \quad \forall l \neq q, \left(\mathbb{E} \left[\mathbf{X}^T \mathbf{y} \mathbf{y}^T \mathbf{X} \right] \right)_{lq} = 0. \quad (5.1)$$

The constraint $\left(\mathbb{E} \left[\mathbf{X}^T \mathbf{y} \mathbf{y}^T \mathbf{X} \right] \right)_{lq} = 0, \forall l \neq q$, ensures that \mathbf{X} decorrelates the features of \mathbf{y} . It is evident that $\mathbb{E} \left[\mathbf{X}^T \mathbf{y} \mathbf{y}^T \mathbf{X} \right] = \mathbf{X}^T \mathbb{E} \left[\mathbf{y} \mathbf{y}^T \right] \mathbf{X}$ will be a diagonal matrix if and only if \mathbf{X} contains the eigenvectors of $\mathbb{E} \left[\mathbf{y} \mathbf{y}^T \right] = \mathbf{\Sigma}$.

In a distributed setup, data is available at geographically scattered locations. Let us consider an undirected and connected network of M nodes described by a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{1, \dots, M\}$ is the set of nodes and \mathcal{E} is the set of edges between the nodes. For each node i , the set of its directly connected neighbors is given by \mathcal{N}_i . The inflow of data streams can be one of the two ways i) nodes receive parts of a data sample for e.g., in sensor networks or, ii) nodes receive full data samples. In this chapter, we assume the setting consistent with the rest of the thesis; data being distributed by samples. If we assume one sample arrives per time instant at one node, then at each time instant M samples will be arriving in total in the network. Assuming that there is no time lag between arrival of samples and the processing times, we ask a question: Can collaboration help the nodes reach the eigenvectors at a faster rate? In other words, is it possible to utilize the information of M samples at every time instant at every node even though a node gets direct access to only one sample. In this chapter we try to answer this question. We propose an distributed algorithm based on a generalization of Oja's rule for estimating the eigenvectors of $\mathbf{\Sigma}$ in the streaming settings.

5.3 Proposed Algorithm

One of the classical methods for solving the PCA problem in streaming data settings is the Oja's algorithm. It is an iterative algorithm for the estimation of dominant eigenvector of $\mathbf{\Sigma}$ and is given as

$$\begin{aligned} \mathbf{x}_t &= \mathbf{x}_{t-1} + \alpha_t \mathbf{y}_t \mathbf{y}_t^T \mathbf{x}_{t-1} \\ \mathbf{x}_t &= \frac{\mathbf{x}_t}{\|\mathbf{x}_t\|}, \end{aligned}$$

where α_t is the step size at time t , \mathbf{y}_t is the sample at time t and $\mathbf{\Sigma} = \mathbb{E} \left[\mathbf{y}_t \mathbf{y}_t^T \right]$. It has been shown in literature [31] that Oja's algorithm converges as $\mathcal{O}(1/t)$ when step size decreasing as

$\alpha_t = \frac{c}{t}$ is used. In the distributed setup, each node i receives a sample $\mathbf{y}_{i,t}$ at time instant t such that $\left[\mathbf{y}_{i,t}\mathbf{y}_{i,t}^T\right] = \mathbf{\Sigma}$ and in order to learn the dominant eigenvector of the population covariance matrix $\mathbf{\Sigma}$, each node can simply run Oja's algorithm as:

$$\begin{aligned}\mathbf{x}_{i,t} &= \mathbf{x}_{i,t-1} + \alpha_t \mathbf{y}_{i,t} \mathbf{y}_{i,t}^T \mathbf{x}_{i,t-1} \\ \mathbf{x}_{i,t} &= \frac{\mathbf{x}_{i,t}}{\|\mathbf{x}_{i,t}\|},\end{aligned}$$

But in order to utilize the information of the other samples that have arrived at the same time instant in the network, we propose to harness the soft power of collaboration. To that end, we propose to apply a combine-and-adapt approach to Oja's rule for dominant eigenvector estimation at each node. This distributed Oja's algorithm essentially combines the eigenvector estimates from all the neighboring nodes and updates the combined estimate based on local data. Algorithm 3 describes the method in detail. The weight matrix $\mathbf{W} = [w_{ij}]$ is a doubly stochastic matrix that conforms to the underlying graph topology [69], i.e., $w_{ij} \neq 0$ if $(i, j) \in \mathcal{E}$ or $i = j$ and 0 otherwise. A necessary assumption for convergence of the algorithm here is the graph connectivity, which ensures that the magnitude of the second largest eigenvalue of \mathbf{W} is strictly less than 1. Algorithm 3 only estimates the dominant eigenvector of the population

Algorithm 3: Distributed Oja's Algorithm ($K = 1$)

Input: $\mathbf{y}_{i,1}, \mathbf{y}_{i,2}, \dots, [w_{ij}], \alpha$
Initialize: $\forall i, \mathbf{x}_{i,0} \leftarrow \mathbf{x}_{\text{init}} : \|\mathbf{x}_{\text{init}}\|^2 = 1$
1: **for** $t = 1, 2, \dots$ **do**
2: $\alpha_t = \frac{\alpha}{t}$
3: $\mathbf{x}_{i,t} = \mathbf{x}_{i,t-1} + \alpha_t \sum_j w_{ij} \mathbf{y}_{j,t} \mathbf{y}_{j,t}^T \mathbf{x}_{j,t-1}$
4: $\mathbf{x}_{i,t} = \frac{\mathbf{x}_{i,t}}{\|\mathbf{x}_{i,t}\|}$
5: **end for**
Return: $\mathbf{x}_{i,t}, i = 1, 2, \dots, M$

covariance matrix $\mathbf{\Sigma}$. Oja's rule was generalized for the estimation of a K -dimensional subspace and it was also shown in [35] that the algorithm

$$\mathbf{X}_t = \mathbf{X}_{t-1} + \alpha_t \mathbf{Y}_t \mathbf{Y}_t^T \mathbf{X}_{t-1} \quad (5.2)$$

$$\mathbf{X}_t = QR(\mathbf{X}_t), \quad (5.3)$$

converges $\mathcal{O}(1/t)$ to the dominant K dimensional subspace. For the estimation of top K eigenvectors, and not just subspace, in the streaming and distributed setting, we propose a different generalization of the Oja's rule called DIStributed Generalized Oja's Method (DIEGO). The idea behind the approach is to subtract the effect of the more dominant eigenvectors while estimating the less dominant ones. Instead of a sequential approach where one eigenvector is

fully estimated and then its effect is subtracted, we estimate all the eigenvectors simultaneously. The detailed approach is given in Algorithm 4.

Algorithm 4: Distributed Generalized Oja's Algorithm (DIEGO)

Input: $\mathbf{y}_{i,1}, \mathbf{y}_{i,2}, \dots, [w_{ij}], \alpha$
Initialize: $\forall i, \mathbf{x}_{i,0} \leftarrow \mathbf{x}_{\text{init}} : \|\mathbf{x}_{\text{init}}\|^2 = 1$
1: **for** $t = 1, 2, \dots$ **do**
2: $\alpha_t = \frac{\alpha}{t}$
3: $\mathbf{x}_{i,k,t} = \mathbf{x}_{i,k,t-1} + \alpha_t \sum_j w_{ij} (\mathbf{I} - \sum_{p=1}^{k-1} \mathbf{x}_{j,p,t-1} \mathbf{x}_{j,p,t-1}^T) \mathbf{y}_{j,t} \mathbf{y}_{j,t}^T \mathbf{x}_{j,k,t-1}$
4: $\mathbf{x}_{i,k,t} = \frac{\mathbf{x}_{i,k,t}}{\|\mathbf{x}_{i,k,t}\|}$
5: **end for**
Return: $\mathbf{x}_{i,k,t}, i = 1, 2, \dots, M, k = 1, \dots, K$

In the next section, we provide detailed analysis of our proposed algorithm DIEGO for the estimation of dominant eigenvector i.e., $K = 1$ (essentially for Algorithm 3). We show that the estimates $\mathbf{x}_{i,1}^{(t)}$ at each node i converge for any random unit-norm initialization and a certain condition on step size, to the eigenvectors $\pm \mathbf{q}_1$ of the covariance matrix Σ .

5.4 Convergence Analysis

This section describes the convergence analysis of the Distributed Oja's Algorithm for the estimation of the dominant eigenvector of the population covariance matrix Σ in a distributed setting. Let us the sample covariance matrices $\mathbf{A}_{i,t} = \mathbf{y}_{i,t} \mathbf{y}_{i,t}^T$ have the following properties:

1. $\|\mathbf{A}_{i,t}\| \leq r$
2. $\left\| \mathbb{E} \left[(\mathbf{A}_{i,t} - \Sigma)(\mathbf{A}_{i,t} - \Sigma)^T \right] \right\| \leq v$

Lemma 16. *The following are true:*

- $1 + x \leq e^x$ for all x
- $1 + x \geq e^{(x-x^2)}$ for all $x \geq 0$
- $\frac{1}{1+x} \leq \sum_{i=1}^{\infty} \frac{1}{(x+i)^2} \leq \frac{1}{x}$
- $\langle \mathbf{A}, \mathbf{B} \rangle \leq \langle \mathbf{A}, \mathbf{C} \rangle$ for PSD matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ with $\mathbf{B} \preceq \mathbf{C}$
- $\text{Tr}(\mathbf{A}^T \mathbf{B}) \leq \frac{1}{2} \text{Tr}(\mathbf{A}^T \mathbf{A} + \mathbf{B}^T \mathbf{B})$ for all matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$

Let $\mathbf{x}_t \in \mathbb{R}^{Md}$ be defined as

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{x}_{1,t} \\ \mathbf{x}_{2,t} \\ \vdots \\ \mathbf{x}_{M,t} \end{bmatrix}$$

Since we are looking at convergence in terms of angle between estimated and true eigenvector, we can push the normalization to the end. In other words, we can analyze the following update equation:

$$\mathbf{x}_{i,t} = \mathbf{x}_{i,t-1} + \alpha_t \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{y}_{j,t} \mathbf{y}_{j,t}^T \mathbf{x}_{j,t-1}, \quad (5.4)$$

where $\mathbf{x}_{i,t}$ is the estimate of the eigenvector at node i after t iterations and α_t is the decreasing step-size. Writing (5.4) using the definition of \mathbf{x}_t , we get

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t \mathbf{x}_{t-1}, \quad (5.5)$$

where $\mathbf{A}_t \in \mathbb{R}^{Md \times Md}$ is a block diagonal matrices with $\mathbf{A}_{i,t} = \mathbf{y}_{i,t} \mathbf{y}_{i,t}^T$ on its diagonal such that $\mathbb{E} [\mathbf{A}_t] = \mathbf{I} \otimes \Sigma = \tilde{\Sigma}$ and $\tilde{\mathbf{W}} = [w_{ij}] \otimes \mathbf{I} = \mathbf{W} \otimes \mathbf{I}$. We can view our algorithm as applying the matrix

$$\mathbf{B}_t = (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t)(\mathbf{I} + \alpha_{t-1} \tilde{\mathbf{W}} \mathbf{A}_{t-1}) \dots (\mathbf{I} + \alpha_1 \tilde{\mathbf{W}} \mathbf{A}_1) \quad (5.6)$$

on \mathbf{x}_0 and giving an output as

$$\mathbf{x}_t = \frac{\mathbf{B}_t \mathbf{x}_0}{\|\mathbf{B}_t \mathbf{x}_0\|} \sqrt{M}, \quad (5.7)$$

where the scaling of \sqrt{M} is to match the fact that unit norm $\mathbf{x}_{i,t}$'s from the original algorithm are concatenated in \mathbf{x}_t . Thus, (5.6) and (5.7) can be viewed as one step power method for \mathbf{B}_t . Further, let's assume that all nodes are initialized with the same unit vector i.e., $\mathbf{x}_{1,0} = \mathbf{x}_{2,0} = \dots = \mathbf{x}_{M,0} = \mathbf{x}$. Thus, $\mathbf{x}_0 = \mathbf{1} \otimes \mathbf{x}$. Further let's assume

Lemma 17. *Let $\mathbf{B} \in \mathbb{R}^{Md \times Md}$ and $\mathbf{D} = \mathbf{B}(\mathbf{1} \otimes \mathbf{I})$, let $\tilde{\mathbf{v}} \in \mathbb{R}^{Md}$ be a vector such that $\tilde{\mathbf{v}} = \mathbf{1} \otimes \mathbf{v}$ where $\mathbf{v} \in \mathbb{R}^d$ is a unit vector. Let \mathbf{V}_\perp be the matrix whose columns form the space orthogonal to $\tilde{\mathbf{v}}$. If $\tilde{\mathbf{x}} = \mathbf{1} \otimes \mathbf{x} \in \mathbb{R}^{Md}$ such that $\mathbf{x} \in \mathbb{R}^d$ is chosen uniformly at random from the surface of the unit sphere, then with probability at least $1 - \delta$*

$$\sin^2 \left(\tilde{\mathbf{v}}, \frac{\mathbf{B} \tilde{\mathbf{x}}}{\|\mathbf{B} \tilde{\mathbf{x}}\|} \sqrt{M} \right) = 1 - \left(\frac{\tilde{\mathbf{v}}^T \mathbf{B} \tilde{\mathbf{x}}}{\|\tilde{\mathbf{v}}\| \|\mathbf{B} \tilde{\mathbf{x}}\|} \right)^2 \leq \frac{C \log(1/\delta)}{\delta^2} \frac{\text{Tr}(\mathbf{V}_\perp^T \mathbf{D} \mathbf{D}^T \mathbf{V}_\perp)}{\frac{1}{M} \tilde{\mathbf{v}}^T \mathbf{D} \mathbf{D}^T \tilde{\mathbf{v}}}$$

Proof. We know $\tilde{\mathbf{x}} = \mathbf{1} \otimes \mathbf{x} = (\mathbf{1} \otimes \mathbf{I})\mathbf{x}$. As \mathbf{x} is uniformly distributed over the unit sphere, we can say $\mathbf{x} = \frac{\mathbf{g}}{\|\mathbf{g}\|}$ where $\mathbf{g} \sim \mathcal{N}(0, \mathbf{I})$.

$$\begin{aligned}
1 - \left(\frac{\tilde{\mathbf{v}}^T \mathbf{B} \tilde{\mathbf{x}}}{\|\tilde{\mathbf{v}}\| \|\mathbf{B} \tilde{\mathbf{x}}\|} \right)^2 &= 1 - \frac{(\tilde{\mathbf{v}}^T \mathbf{B} (\mathbf{1} \otimes \mathbf{I}) \mathbf{x})^2}{M \|\mathbf{B} (\mathbf{1} \otimes \mathbf{I}) \mathbf{x}\|^2} = 1 - \frac{(\tilde{\mathbf{v}}^T \mathbf{D} \mathbf{x})^2}{M \|\mathbf{D} \mathbf{x}\|^2} \quad \text{where } \mathbf{D} = \mathbf{B} (\mathbf{1} \otimes \mathbf{I}) \in \mathbb{R}^{Md \times d} \\
&= 1 - \frac{(\tilde{\mathbf{v}}^T \mathbf{D} \mathbf{x})^2}{M \|\mathbf{D} \mathbf{x}\|^2} = \frac{M \mathbf{x}^T \mathbf{D}^T \mathbf{D} \mathbf{x} - \mathbf{x}^T \mathbf{D}^T \tilde{\mathbf{v}} \tilde{\mathbf{v}}^T \mathbf{D} \mathbf{x}}{M \mathbf{x}^T \mathbf{D}^T \mathbf{D} \mathbf{x}} \\
&= \frac{\mathbf{x}^T \mathbf{D}^T (M \mathbf{I} - \tilde{\mathbf{v}} \tilde{\mathbf{v}}^T) \mathbf{D} \mathbf{x}}{M \mathbf{x}^T \mathbf{D}^T \mathbf{D} \mathbf{x}} = \frac{\mathbf{g}^T \mathbf{D}^T (\mathbf{I} - \frac{1}{M} \tilde{\mathbf{v}} \tilde{\mathbf{v}}^T) \mathbf{D} \mathbf{g}}{\mathbf{g}^T \mathbf{D}^T \mathbf{D} \mathbf{g}} \\
&\stackrel{\zeta_1}{\leq} \frac{C}{\delta^2} \frac{\mathbf{g}^T \mathbf{D}^T (\mathbf{I} - \frac{1}{M} \tilde{\mathbf{v}} \tilde{\mathbf{v}}^T) \mathbf{D} \mathbf{g}}{\frac{1}{M} \tilde{\mathbf{v}}^T \mathbf{D} \mathbf{D}^T \tilde{\mathbf{v}}} \\
&\stackrel{\zeta_2}{\leq} \frac{C \log(1/\delta)}{\delta^2} \frac{\text{Tr}(\mathbf{D}^T (\mathbf{I} - \frac{1}{M} \tilde{\mathbf{v}} \tilde{\mathbf{v}}^T) \mathbf{D})}{\frac{1}{M} \tilde{\mathbf{v}}^T \mathbf{D} \mathbf{D}^T \tilde{\mathbf{v}}},
\end{aligned}$$

where C is an absolute constant. ζ_1 follows from the fact that $M \mathbf{g}^T \mathbf{D}^T \mathbf{D} \mathbf{g} \geq (\tilde{\mathbf{v}}^T \mathbf{D} \mathbf{g})^2 \geq \frac{\delta^2}{C} \tilde{\mathbf{v}}^T \mathbf{D} \mathbf{D}^T \tilde{\mathbf{v}}$, where the first inequality is Cauchy Schwarz and the second inequality follows from the fact that $\tilde{\mathbf{v}}^T \mathbf{D} \mathbf{g}$ is a Gaussian random variable with variance $\|\mathbf{D}^T \tilde{\mathbf{v}}\|^2$ and $\Pr(|g| \leq \delta) \leq C\delta$ for a normal random variable $g \sim \mathcal{N}(0, 1)$. Similarly, ζ_2 follows from the fact that $\mathbf{g}^T \mathbf{D}^T (\mathbf{I} - \frac{1}{M} \tilde{\mathbf{v}} \tilde{\mathbf{v}}^T) \mathbf{D} \mathbf{g}$ is a χ^2 random variable with $\text{Tr}(\mathbf{D}^T (\mathbf{I} - \frac{1}{M} \tilde{\mathbf{v}} \tilde{\mathbf{v}}^T) \mathbf{D})$ degrees of freedom. \square

Let $\mathbf{q}_1, \dots, \mathbf{q}_d$ denote the eigenvectors of Σ and $\lambda_1 > \lambda_2 \geq \dots \geq \lambda_d$ be the corresponding eigenvalues. Let $\tilde{\mathbf{q}}_1 = \mathbf{1} \otimes \mathbf{q}_1$. Now, $\frac{1}{\sqrt{M}} \mathbf{1}$ is the dominant unit norm eigenvector of \mathbf{W} with corresponding eigenvalue of 1, which implies that $\frac{1}{\sqrt{M}} \tilde{\mathbf{q}}_1 = \frac{1}{\sqrt{M}} \mathbf{1} \otimes \mathbf{q}_1$ is the dominant eigenvector of $\mathbf{W} \otimes \Sigma$. Evidently, if \mathbf{Q}_\perp is the matrix of orthogonal columns that span the subspace orthogonal to $\frac{1}{\sqrt{M}} \tilde{\mathbf{q}}_1$, then $\mathbf{Q}_\perp \mathbf{Q}_\perp^T = \mathbf{I} - \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T$. Lemma 17 shows that to prove the convergence of distributed Oja's algorithm, we need two important pieces. First, we need to show that with constant probability $\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1$ is relatively large and second, $\text{Tr}(\mathbf{D}_t^T \mathbf{Q}_\perp \mathbf{Q}_\perp^T \mathbf{D}_t) = \text{Tr}(\mathbf{Q}_\perp^T \mathbf{D}_t \mathbf{D}_t^T \mathbf{Q}_\perp)$ is relatively small.

Lemma 18. *For all $t \geq 0$ and $\alpha_t \geq 0$, we have*

$$\|\mathbb{E} [\mathbf{D}_t \mathbf{D}_t^T]\| \leq M \exp\left(\sum_{m \in [t]} 2\alpha_m \lambda_1 + \alpha_m^2 \bar{v}\right),$$

where $\bar{v} = v + \lambda_1^2$.

Proof. Let $\eta_t = \|\mathbb{E} [\mathbf{D}_t \mathbf{D}_t^T]\|$, i.e., $\mathbb{E} [\mathbf{D}_t \mathbf{D}_t^T] \preceq \eta_t \mathbf{I}$. Now,

$$\mathbf{D}_t = \mathbf{B}_t (\mathbf{1} \otimes \mathbf{I}) = (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t) \mathbf{B}_{t-1} (\mathbf{1} \otimes \mathbf{I}) = (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t) \mathbf{D}_{t-1}$$

For all $t > 0$,

$$\begin{aligned}
\mathbb{E} [\mathbf{D}_t \mathbf{D}_t^T] &= \mathbb{E} [(\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t) \mathbf{D}_{t-1} \mathbf{D}_{t-1}^T (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t)^T] \\
&\preceq \eta_{t-1} \mathbb{E} [(\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t) (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t)^T] \\
&= \eta_{t-1} \mathbb{E} [\mathbf{I} + \alpha_t \mathbf{A}_t \tilde{\mathbf{W}} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t + \alpha_t^2 \tilde{\mathbf{W}} \mathbf{A}_t^2 \tilde{\mathbf{W}}] \\
&= \eta_{t-1} \left[\mathbf{I} + \alpha_t \tilde{\Sigma} \tilde{\mathbf{W}} + \alpha_t \tilde{\mathbf{W}} \tilde{\Sigma} + \alpha_t^2 \mathbb{E} [\tilde{\mathbf{W}} \mathbf{A}_t^2 \tilde{\mathbf{W}}] \right] \tag{5.8}
\end{aligned}$$

Since $\left\| \mathbb{E} [(\mathbf{A}_{i,t} - \Sigma)(\mathbf{A}_{i,t} - \Sigma)^T] \right\| \leq v$, we have $\mathbb{E} [(\mathbf{A}_{i,t} - \Sigma)(\mathbf{A}_{i,t} - \Sigma)^T] \preceq v \mathbf{I}$. This implies

$$\mathbb{E} [(\tilde{\mathbf{W}} \mathbf{A}_t - \tilde{\mathbf{W}} \tilde{\Sigma})(\tilde{\mathbf{W}} \mathbf{A}_t - \tilde{\mathbf{W}} \tilde{\Sigma})^T] \preceq v \mathbf{I}$$

Thus,

$$\mathbb{E} [\tilde{\mathbf{W}} \mathbf{A}_t^2 \tilde{\mathbf{W}}] = \tilde{\mathbf{W}} \tilde{\Sigma}^2 \tilde{\mathbf{W}} + \mathbb{E} [(\tilde{\mathbf{W}} \mathbf{A}_t - \tilde{\mathbf{W}} \tilde{\Sigma})(\tilde{\mathbf{W}} \mathbf{A}_t - \tilde{\mathbf{W}} \tilde{\Sigma})^T] \preceq \tilde{\Sigma}^2 + v \mathbf{I}$$

Using the above inequality in (5.8), we get

$$\mathbb{E} [\mathbf{D}_t \mathbf{D}_t^T] \preceq \eta_{t-1} \left[\mathbf{I} + \alpha_t \tilde{\Sigma} \tilde{\mathbf{W}} + \alpha_t \tilde{\mathbf{W}} \tilde{\Sigma} + \alpha_t^2 (\tilde{\Sigma}^2 + v \mathbf{I}) \right]$$

We know, $\|\tilde{\mathbf{W}}^2\| = \|\tilde{\mathbf{W}}\| = 1$, $\|\tilde{\Sigma}\| = \lambda_1$ and $\|\tilde{\Sigma}^2\| = \lambda_1^2$. Therefore,

$$\eta_t \leq \eta_{t-1} (1 + 2\alpha_t \lambda_1 + \alpha_t^2 (\lambda_1^2 + v)) = \eta_{t-1} (1 + 2\alpha_t \lambda_1 + \alpha_t^2 \bar{v})$$

Using the fact that $\mathbf{B}_0 = \mathbf{I}$, i.e., $\eta_0 = M$ and $1 + x \leq \exp(x)$, the result follows. \square

Using Lemma 18 we next bound $\mathbb{E} [\text{Tr}(\mathbf{Q}_\perp^T \mathbf{D}_t \mathbf{D}_t^T \mathbf{Q}_\perp)]$. This will help us in bounding $\text{Tr}(\mathbf{Q}_\perp^T \mathbf{D}_t \mathbf{D}_t^T \mathbf{Q}_\perp)$ using Markov's inequality.

Lemma 19. For all $t \geq 0$ and $\alpha_t \leq \frac{1}{\lambda_1}$,

$$\mathbb{E} [\text{Tr}(\mathbf{Q}_\perp^T \mathbf{D}_t \mathbf{D}_t^T \mathbf{Q}_\perp)] \leq M \exp \left(\sum_{m \in [t]} 2\alpha_m \tilde{\lambda}_2 + \alpha_m^2 \bar{v} \right) \left(d + \sum_{p \in [t]} \alpha_p^2 v \exp \left(\sum_{m \in [p]} 2\alpha_m (\lambda_1 - \tilde{\lambda}_2) \right) \right).$$

Proof. Let $\eta_t = \mathbb{E} [\text{Tr}(\mathbf{Q}_\perp^T \mathbf{D}_t \mathbf{D}_t^T \mathbf{Q}_\perp)] = \left\langle \mathbb{E} [\mathbf{D}_t \mathbf{D}_t^T], \mathbf{Q}_\perp \mathbf{Q}_\perp^T \right\rangle$. Now,

$$\begin{aligned}
\mathbb{E} [\text{Tr}(\mathbf{Q}_\perp^T \mathbf{D}_t \mathbf{D}_t^T \mathbf{Q}_\perp)] &= \mathbb{E} [\text{Tr}(\mathbf{Q}_\perp^T (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t) \mathbf{D}_{t-1} \mathbf{D}_{t-1}^T (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t)^T \mathbf{Q}_\perp)] \\
&= \mathbb{E} \left\langle \mathbf{D}_{t-1} \mathbf{D}_{t-1}^T, (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t) \mathbf{Q}_\perp \mathbf{Q}_\perp^T (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t)^T \right\rangle \\
&= \left\langle \mathbb{E} [\mathbf{D}_{t-1} \mathbf{D}_{t-1}^T], \mathbb{E} [(\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t) \mathbf{Q}_\perp \mathbf{Q}_\perp^T (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t)^T] \right\rangle \tag{5.9}
\end{aligned}$$

Now,

$$\begin{aligned}
& \mathbb{E} \left[(\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t) \mathbf{Q}_\perp \mathbf{Q}_\perp^\top (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t)^\top \right] \\
&= \mathbb{E} \left[(\mathbf{Q}_\perp \mathbf{Q}_\perp^\top + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\top) (\mathbf{I} + \alpha_t \mathbf{A}_t \tilde{\mathbf{W}}) \right] \\
&= \mathbb{E} \left[\mathbf{Q}_\perp \mathbf{Q}_\perp^\top + \alpha_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\top \mathbf{A}_t \tilde{\mathbf{W}} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\top + \alpha_t^2 \tilde{\mathbf{W}} \mathbf{A}_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\top \mathbf{A}_t \tilde{\mathbf{W}} \right] \\
&= \mathbf{Q}_\perp \mathbf{Q}_\perp^\top + \alpha_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\top \tilde{\Sigma} \tilde{\mathbf{W}} + \alpha_t \tilde{\mathbf{W}} \tilde{\Sigma} \mathbf{Q}_\perp \mathbf{Q}_\perp^\top + \alpha_t^2 \mathbb{E} \left[\tilde{\mathbf{W}} \mathbf{A}_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\top \mathbf{A}_t \tilde{\mathbf{W}} \right] \\
&= \mathbf{Q}_\perp \mathbf{Q}_\perp^\top + \alpha_t \tilde{\mathbf{W}} \tilde{\Sigma} \mathbf{Q}_\perp \mathbf{Q}_\perp^\top + \alpha_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\top \tilde{\Sigma} \tilde{\mathbf{W}} + \alpha_t^2 \tilde{\mathbf{W}} \tilde{\Sigma} \mathbf{Q}_\perp \mathbf{Q}_\perp^\top \tilde{\Sigma} \tilde{\mathbf{W}} + \alpha_t^2 \mathbb{E} \left[\tilde{\mathbf{W}} (\mathbf{A}_t - \tilde{\Sigma}) \mathbf{Q}_\perp \mathbf{Q}_\perp^\top (\mathbf{A}_t - \tilde{\Sigma}) \tilde{\mathbf{W}} \right] \\
&\preceq \mathbf{Q}_\perp \mathbf{Q}_\perp^\top + \alpha_t \tilde{\mathbf{W}} \tilde{\Sigma} \mathbf{Q}_\perp \mathbf{Q}_\perp^\top + \alpha_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\top \tilde{\Sigma} \tilde{\mathbf{W}} + \alpha_t^2 \tilde{\mathbf{W}} \tilde{\Sigma} \mathbf{Q}_\perp \mathbf{Q}_\perp^\top \tilde{\Sigma} \tilde{\mathbf{W}} + \alpha_t^2 \mathbb{E} \left[(\tilde{\mathbf{W}} \mathbf{A}_t - \tilde{\mathbf{W}} \tilde{\Sigma}) (\tilde{\mathbf{W}} \mathbf{A}_t - \tilde{\mathbf{W}} \tilde{\Sigma})^\top \right] \\
&\preceq \mathbf{Q}_\perp \mathbf{Q}_\perp^\top + \alpha_t \tilde{\mathbf{W}} \tilde{\Sigma} \mathbf{Q}_\perp \mathbf{Q}_\perp^\top + \alpha_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\top \tilde{\Sigma} \tilde{\mathbf{W}} + \alpha_t^2 \tilde{\mathbf{W}} \tilde{\Sigma} \mathbf{Q}_\perp \mathbf{Q}_\perp^\top \tilde{\Sigma} \tilde{\mathbf{W}} + \alpha_t^2 v \mathbf{I} \tag{5.10}
\end{aligned}$$

Now, $\tilde{\mathbf{W}} \tilde{\Sigma} = \tilde{\Sigma} \tilde{\mathbf{W}} = \mathbf{W} \otimes \Sigma$. Since $\frac{1}{\sqrt{M}} \mathbf{1} \otimes \mathbf{q}_1$ is the dominant eigenvector of $\mathbf{W} \otimes \Sigma$ corresponding to the eigenvalue λ_1 and \mathbf{Q}_\perp is the matrix of eigenvectors orthogonal to $\frac{1}{\sqrt{M}} \mathbf{1} \otimes \mathbf{q}_1$. Now the second largest eigenvalue of $\mathbf{W} \otimes \Sigma$ is $\tilde{\lambda}_2 = \max\{\lambda_2, \sigma \lambda_1\}$, where σ is the second largest eigenvalue of \mathbf{W} and $\sigma < 1$. Thus from (5.10), we get

$$\begin{aligned}
\mathbb{E} \left[(\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t)^\top \mathbf{Q}_\perp \mathbf{Q}_\perp^\top (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t) \right] &\preceq (1 + 2\alpha_t \tilde{\lambda}_2 + \alpha_t^2 \tilde{\lambda}_2^2) \mathbf{Q}_\perp \mathbf{Q}_\perp^\top + \alpha_t^2 v \mathbf{I} \\
&\preceq (1 + 2\alpha_t \tilde{\lambda}_2 + \alpha_t^2 \lambda_1^2 + \alpha_t^2 v) \mathbf{Q}_\perp \mathbf{Q}_\perp^\top + \alpha_t^2 v \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^\top
\end{aligned}$$

Plugging the above in (5.9), we get

$$\begin{aligned}
\eta_t &\leq (1 + 2\alpha_t \tilde{\lambda}_2 + \alpha_t^2 \lambda_1^2 + \alpha_t^2 v) \left\langle \mathbb{E} \left[\mathbf{D}_{t-1} \mathbf{D}_{t-1}^\top \right], \mathbf{Q}_\perp \mathbf{Q}_\perp^\top \right\rangle + \alpha_t^2 v \left\langle \mathbb{E} \left[\mathbf{D}_{t-1} \mathbf{D}_{t-1}^\top \right], \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^\top \right\rangle \\
&\leq (1 + 2\alpha_t \tilde{\lambda}_2 + \alpha_t^2 \bar{v}) \eta_{t-1} + \alpha_t^2 v \left\| \mathbb{E} \left[\mathbf{D}_{t-1} \mathbf{D}_{t-1}^\top \right] \right\| \\
&\leq \exp(2\alpha_t \tilde{\lambda}_2 + \alpha_t^2 \bar{v}) \eta_{t-1} + v M \alpha_t^2 \exp\left(\sum_{m \in [t-1]} 2\alpha_m \lambda_1 + \alpha_m^2 \bar{v} \right), \tag{5.11}
\end{aligned}$$

where last inequality follows from $1 + x \leq \exp(x)$ and Lemma 18. Recuring (5.11), we get

$$\begin{aligned}
\eta_t &\leq \exp\left(\sum_{m \in [t]} 2\alpha_m \tilde{\lambda}_2 + \alpha_m^2 \bar{v} \right) \eta_0 + v M \sum_{p \in [t]} \alpha_p^2 \exp\left(\sum_{m \in [p]} 2\alpha_m \lambda_1 + \alpha_m^2 \bar{v} \right) \exp\left(\sum_{m=p+1}^t 2\alpha_m \tilde{\lambda}_2 + \alpha_m^2 \bar{v} \right) \\
&= \exp\left(\sum_{m \in [t]} 2\alpha_m \tilde{\lambda}_2 + \alpha_m^2 \bar{v} \right) \left(\eta_0 + v M \sum_{p \in [t]} \alpha_p^2 \exp\left(\sum_{m \in [p]} 2\alpha_m \lambda_1 + \alpha_m^2 \bar{v} \right) \exp\left(- \sum_{m \in [p]} 2\alpha_m \tilde{\lambda}_2 - \alpha_m^2 \bar{v} \right) \right) \\
&= \exp\left(\sum_{m \in [t]} 2\alpha_m \tilde{\lambda}_2 + \alpha_m^2 \bar{v} \right) \left(\eta_0 + v M \sum_{p \in [t]} \alpha_p^2 \exp\left(\sum_{m \in [p]} 2\alpha_m (\lambda_1 - \tilde{\lambda}_2) \right) \right).
\end{aligned}$$

Since $\mathbf{D}_0 = \mathbf{1} \otimes \mathbf{I}$, $\eta_0 = M(d-1) \leq Md$. Thus,

$$\eta_t \leq M \exp\left(\sum_{m \in [t]} 2\alpha_m \tilde{\lambda}_2 + \alpha_m^2 \bar{v} \right) \left(d + v \sum_{p \in [t]} \alpha_p^2 \exp\left(\sum_{m \in [p]} 2\alpha_m (\lambda_1 - \tilde{\lambda}_2) \right) \right)$$

□

Next, we lower bound $\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1$. In Lemma 20 we lower bound $\mathbb{E} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right]$ and in Lemma 21 we upper bound $\text{Var} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right]$.

Lemma 20. *For all $t \geq 0$ and $\alpha_t \geq 0$, we have*

$$\mathbb{E} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right] \geq \exp \left(\sum_{m \in [t]} 2\alpha_m \lambda_1 - 4\alpha_m^2 \lambda_1^2 \right)$$

Proof. Let $\eta_t = \mathbb{E} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right] = \mathbb{E} \left[\text{Tr} \left(\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right) \right]$. Since $\mathbf{D}_t = (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t) \mathbf{D}_{t-1}$, we have

$$\begin{aligned} \eta_t &= \mathbb{E} \left[\text{Tr} \left(\frac{1}{M} \tilde{\mathbf{q}}_1^T (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t) \mathbf{D}_{t-1} \mathbf{D}_{t-1}^T (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t)^T \tilde{\mathbf{q}}_1 \right) \right] \\ &= \left\langle \mathbb{E} \left[\mathbf{D}_{t-1} \mathbf{D}_{t-1}^T \right], \mathbb{E} \left[(\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t) \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t)^T \right] \right\rangle \\ &= \left\langle \mathbb{E} \left[\mathbf{D}_{t-1} \mathbf{D}_{t-1}^T \right], \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T + \alpha_t \tilde{\mathbf{W}} \tilde{\Sigma} \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T + \alpha_t \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \tilde{\Sigma} \tilde{\mathbf{W}} + \alpha_t^2 \mathbb{E} \left[\tilde{\mathbf{W}} \mathbf{A}_t \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \mathbf{A}_t \tilde{\mathbf{W}} \right] \right\rangle \\ &\geq \left\langle \mathbb{E} \left[\mathbf{D}_{t-1} \mathbf{D}_{t-1}^T \right], \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T + \alpha_t \tilde{\mathbf{W}} \tilde{\Sigma} \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T + \alpha_t \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \tilde{\Sigma} \tilde{\mathbf{W}} \right\rangle \\ &= \left\langle \mathbb{E} \left[\mathbf{D}_{t-1} \mathbf{D}_{t-1}^T \right], \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T + 2\alpha_t \lambda_1 \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \right\rangle \\ &= (1 + 2\alpha_t \lambda_1) \left\langle \mathbb{E} \left[\mathbf{D}_{t-1} \mathbf{D}_{t-1}^T \right], \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \right\rangle = (1 + 2\alpha_t \lambda_1) \eta_{t-1} \end{aligned} \quad (5.12)$$

Since $\mathbf{D}_0 = \mathbf{1} \otimes \mathbf{I}$, $\eta_0 = M$. Proceeding recursively and using the fact that $1 + x \geq \exp(x - x^2)$ for all $x \geq 0$, we get

$$\mathbb{E} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right] \geq M \exp \left(\sum_{m \in [t]} 2\alpha_m \lambda_1 - 4\alpha_m^2 \lambda_1^2 \right). \quad (5.13)$$

□

Lemma 21. *For $t \geq 0$ and $\alpha_t \leq \frac{1}{4r}$, $\forall t$,*

$$\mathbb{E} \left[\left(\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right)^2 \right] \leq M^2 \exp \left(\sum_{m \in [t]} 4\alpha_m \lambda_1 + 10\alpha_m^2 \bar{v} \right)$$

Proof. Let $\mathbf{H}_{t,s} = (\mathbf{I} + \alpha_t \tilde{\mathbf{W}} \mathbf{A}_t) \dots (\mathbf{I} + \alpha_{t-s+1} \tilde{\mathbf{W}} \mathbf{A}_{t-s+1})$ and $\eta_s = \mathbb{E} \left[\left(\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,s} \mathbf{H}_{t,s}^T \tilde{\mathbf{q}}_1 \right)^2 \right]$.

Note that $\eta_t = \mathbb{E} \left[\left(\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} \mathbf{H}_{t,t}^T \tilde{\mathbf{q}}_1 \right)^2 \right]$. Now,

$$\begin{aligned} \eta_t &= \text{Tr} \left(\mathbb{E} \left[\left(\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} \mathbf{H}_{t,t}^T \tilde{\mathbf{q}}_1 \right)^2 \right] \right) = \text{Tr} \left(\mathbb{E} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} \mathbf{H}_{t,t}^T \tilde{\mathbf{q}}_1 \frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} \mathbf{H}_{t,t}^T \tilde{\mathbf{q}}_1 \right] \right) \\ &= \text{Tr} \left(\mathbb{E} \left[\mathbf{H}_{t,t}^T \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} \mathbf{H}_{t,t}^T \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} \right] \right) \\ &= \text{Tr} \left(\mathbb{E} \left[\begin{aligned} &(\mathbf{I} + \alpha_1 \tilde{\mathbf{W}} \mathbf{A}_1)^T \mathbf{H}_{t,t-1}^T \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t-1} (\mathbf{I} + \alpha_1 \tilde{\mathbf{W}} \mathbf{A}_1) (\mathbf{I} + \alpha_1 \tilde{\mathbf{W}} \mathbf{A}_1)^T \\ &\times \mathbf{H}_{t,t-1}^T \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t-1} (\mathbf{I} + \alpha_1 \tilde{\mathbf{W}} \mathbf{A}_1) \end{aligned} \right] \right) \\ &= \text{Tr} \left(\mathbb{E} \left[(\mathbf{I} + \alpha_1 \mathbf{A}_1 \tilde{\mathbf{W}}) \mathbf{G}_{t-1} (\mathbf{I} + \alpha_1 \tilde{\mathbf{W}} \mathbf{A}_1) (\mathbf{I} + \alpha_1 \mathbf{A}_1 \tilde{\mathbf{W}}) \mathbf{G}_{t-1} (\mathbf{I} + \alpha_1 \tilde{\mathbf{W}} \mathbf{A}_1) \right] \right), \end{aligned} \quad (5.14)$$

where $\mathbf{G}_{t-1} = \mathbf{H}_{t,t-1}^T \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t-1} \in \mathbb{R}^{Md \times Md}$. To bound η_t , we first bound the above expression (5.14) for an arbitrary $\mathbf{G}_{t-1} = \mathbf{G}$. We take expectation over only \mathbf{A}_1 and then finally take an expectation over \mathbf{G}_{t-1} . Let $\mathbf{1} \otimes \mathbf{I} = \mathbf{C}$ (for ease of notation), then for a fixed arbitrary \mathbf{G} , we have

$$\begin{aligned}
& \text{Tr} \left(\mathbb{E} \left[(\mathbf{I} + \alpha_1 \mathbf{A}_1 \tilde{\mathbf{W}}) \mathbf{G} (\mathbf{I} + \alpha_1 \tilde{\mathbf{W}} \mathbf{A}_1) (\mathbf{I} + \alpha_1 \mathbf{A}_1 \tilde{\mathbf{W}}) \mathbf{G} (\mathbf{I} + \alpha_1 \tilde{\mathbf{W}} \mathbf{A}_1) \right] \right) \\
&= \text{Tr} \left(\mathbb{E} \left[(\mathbf{G} + \alpha_1 \mathbf{A}_1 \tilde{\mathbf{W}} \mathbf{G}) (\mathbf{I} + \alpha_1 \tilde{\mathbf{W}} \mathbf{A}_1) (\mathbf{G} + \alpha_1 \mathbf{A}_1 \tilde{\mathbf{W}} \mathbf{G}) (\mathbf{I} + \alpha_1 \tilde{\mathbf{W}} \mathbf{A}_1) \right] \right) \\
&= \text{Tr} \left(\mathbb{E} \left[(\mathbf{G} + \alpha_1 \mathbf{A}_1 \tilde{\mathbf{W}} \mathbf{G} + \alpha_1 \mathbf{G} \tilde{\mathbf{W}} \mathbf{A}_1 + \alpha_1^2 \mathbf{A}_1 \tilde{\mathbf{W}} \mathbf{G} \tilde{\mathbf{W}} \mathbf{A}_1)^2 \right] \right) \\
&= \text{Tr} \left(\mathbf{G}^2 + \alpha_1 \mathbb{E} \left[\mathbf{G} \mathbf{A}_1 \tilde{\mathbf{W}} \mathbf{G} \right] + \alpha_1 \left[\mathbf{G}^2 \tilde{\mathbf{W}} \mathbf{A}_1 \right] + \alpha_1^2 \left[\mathbf{G} \mathbf{A}_1 \tilde{\mathbf{W}} \mathbf{G} \tilde{\mathbf{W}} \mathbf{A}_1 \right] + \right. \\
&\quad \alpha_1 \left[\mathbf{A}_1 \tilde{\mathbf{W}} \mathbf{G}^2 \right] + \alpha_1^2 \left[\mathbf{A}_1 \tilde{\mathbf{W}} \mathbf{G} \mathbf{A}_1 \tilde{\mathbf{W}} \mathbf{G} \right] + \alpha_1^2 \left[\mathbf{A}_1 \tilde{\mathbf{W}} \mathbf{G}^2 \tilde{\mathbf{W}} \mathbf{A}_1 \right] + \alpha_1^3 \left[\mathbf{A}_1 \tilde{\mathbf{W}} \mathbf{G} \mathbf{A}_1 \tilde{\mathbf{W}} \mathbf{G} \tilde{\mathbf{W}} \mathbf{A}_1 \right] \\
&\quad \alpha_1 \left[\mathbf{G} \tilde{\mathbf{W}} \mathbf{A}_1 \mathbf{G} \right] + \alpha_1^2 \left[\mathbf{G} \tilde{\mathbf{W}} \mathbf{A}_1^2 \tilde{\mathbf{W}} \mathbf{G} \right] + \alpha_1^2 \left[\mathbf{G} \tilde{\mathbf{W}} \mathbf{A}_1 \mathbf{G} \tilde{\mathbf{W}} \mathbf{A}_1 \right] + \alpha_1^3 \left[\mathbf{G} \tilde{\mathbf{W}} \mathbf{A}_1^2 \tilde{\mathbf{W}} \mathbf{G} \tilde{\mathbf{W}} \mathbf{A}_1 \right] \\
&\quad \alpha_1^2 \left[\mathbf{A}_1 \tilde{\mathbf{W}} \mathbf{G} \tilde{\mathbf{W}} \mathbf{A}_1 \mathbf{G} \right] + \alpha_1^3 \left[\mathbf{A}_1 \tilde{\mathbf{W}} \mathbf{G} \tilde{\mathbf{W}} \mathbf{A}_1^2 \tilde{\mathbf{W}} \mathbf{G} \right] + \alpha_1^3 \left[\mathbf{A}_1 \tilde{\mathbf{W}} \mathbf{G} \tilde{\mathbf{W}} \mathbf{A}_1 \mathbf{G} \tilde{\mathbf{W}} \mathbf{A}_1 \right] + \\
&\quad \left. \alpha_1^4 \left[\mathbf{A}_1 \tilde{\mathbf{W}} \mathbf{G} \tilde{\mathbf{W}} \mathbf{A}_1^2 \tilde{\mathbf{W}} \mathbf{G} \tilde{\mathbf{W}} \mathbf{A}_1 \right] \right) \tag{5.15}
\end{aligned}$$

We simplify and bound the terms (5.15) as follows.

$$\begin{aligned}
1^{st} \text{ order: } & \text{Tr}\left(\mathbb{E}\left[\mathbf{G}\mathbf{A}_1\tilde{\mathbf{W}}\mathbf{G}\right] + \mathbb{E}\left[\mathbf{G}^2\tilde{\mathbf{W}}\mathbf{A}_1\right] + \mathbb{E}\left[\mathbf{A}_1\tilde{\mathbf{W}}\mathbf{G}^2\right] + \mathbb{E}\left[\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1\mathbf{G}\right]\right) \\
& = 2\text{Tr}\left(\mathbb{E}\left[\tilde{\mathbf{W}}\mathbf{A}_1\mathbf{G}^2\right] + \mathbb{E}\left[\mathbf{A}_1\tilde{\mathbf{W}}\mathbf{G}^2\right]\right) = 2\text{Tr}\left(\tilde{\mathbf{W}}\tilde{\Sigma}\mathbf{G}^2 + \tilde{\Sigma}\tilde{\mathbf{W}}\mathbf{G}^2\right) \\
& \leq 2(\|\tilde{\mathbf{W}}\tilde{\Sigma}\|\text{Tr}(\mathbf{G}^2) + \|\tilde{\Sigma}\tilde{\mathbf{W}}\|\text{Tr}(\mathbf{G}^2)) = 4\lambda_1\text{Tr}(\mathbf{G}^2) \\
2^{nd} \text{ order: } & \text{Tr}\left(\mathbb{E}\left[\mathbf{G}\mathbf{A}_1\tilde{\mathbf{W}}\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1\right] + \mathbb{E}\left[\mathbf{A}_1\tilde{\mathbf{W}}\mathbf{G}\mathbf{A}_1\tilde{\mathbf{W}}\mathbf{G}\right] + \mathbb{E}\left[\mathbf{A}_1\tilde{\mathbf{W}}\mathbf{G}^2\tilde{\mathbf{W}}\mathbf{A}_1\right] + \right. \\
& \quad \mathbb{E}\left[\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1^2\tilde{\mathbf{W}}\mathbf{G}\right] + \mathbb{E}\left[\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1\right] + \mathbb{E}\left[\mathbf{A}_1\tilde{\mathbf{W}}\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1\mathbf{G}\right]\left.\right) \\
& = 2\text{Tr}\left(\mathbb{E}\left[\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1^2\tilde{\mathbf{W}}\mathbf{G}\right]\right) + 2\text{Tr}\left(\mathbb{E}\left[\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1\mathbf{G}\mathbf{A}_1\tilde{\mathbf{W}}\right]\right) + \text{Tr}\left(\mathbb{E}\left[\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1\right]\right) \\
& \quad + \text{Tr}\left(\mathbb{E}\left[\tilde{\mathbf{W}}\mathbf{G}\mathbf{A}_1\tilde{\mathbf{W}}\mathbf{G}\mathbf{A}_1\right]\right) \\
& \leq 2\text{Tr}\left(\mathbf{G}\tilde{\mathbf{W}}\mathbb{E}\left[\mathbf{A}_1^2\right]\tilde{\mathbf{W}}\mathbf{G}\right) + \mathbb{E}\left[\text{Tr}\left(\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1^2\tilde{\mathbf{W}}\mathbf{G}\right)\right] + \mathbb{E}\left[\text{Tr}\left(\tilde{\mathbf{W}}\mathbf{A}_1\mathbf{G}^2\mathbf{A}_1\tilde{\mathbf{W}}\right)\right] + \\
& \quad 0.5\mathbb{E}\left[\text{Tr}\left(\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1^2\tilde{\mathbf{W}}\mathbf{G}\right)\right] + 0.5\mathbb{E}\left[\text{Tr}\left(\mathbf{A}_1\tilde{\mathbf{W}}\mathbf{G}^2\tilde{\mathbf{W}}\mathbf{A}_1\right)\right] + 0.5\mathbb{E}\left[\text{Tr}\left(\tilde{\mathbf{W}}\mathbf{G}\mathbf{A}_1^2\tilde{\mathbf{W}}\right)\right] \\
& \quad + 0.5\mathbb{E}\left[\text{Tr}\left(\mathbf{A}_1\mathbf{G}\tilde{\mathbf{W}}^2\mathbf{G}\mathbf{A}_1\right)\right] \\
& = 0.5\mathbb{E}\left[\text{Tr}\left(\mathbf{A}_1^2\tilde{\mathbf{W}}\mathbf{G}^2\tilde{\mathbf{W}}\right)\right] + \mathbb{E}\left[\text{Tr}\left(\mathbf{A}_1^2\mathbf{G}\tilde{\mathbf{W}}^2\mathbf{G}\right)\right] + \mathbb{E}\left[\text{Tr}\left(\tilde{\mathbf{W}}^2\mathbf{A}_1\mathbf{G}^2\mathbf{A}_1\right)\right] \\
& \quad + 3.5\text{Tr}\left(\mathbf{G}\tilde{\mathbf{W}}\mathbb{E}\left[\mathbf{A}_1^2\right]\tilde{\mathbf{W}}\mathbf{G}\right) \\
& \leq 0.5\text{Tr}\left(\mathbb{E}\left[\mathbf{A}_1^2\right]\tilde{\mathbf{W}}\mathbf{G}^2\tilde{\mathbf{W}}\right) + \text{Tr}\left(\mathbb{E}\left[\mathbf{A}_1^2\right]\mathbf{G}\tilde{\mathbf{W}}^2\mathbf{G}\right) + \mathbb{E}\left[\text{Tr}\left(\mathbf{A}_1\mathbf{G}^2\mathbf{A}_1\right)\right] \\
& \quad + 3.5\text{Tr}\left(\mathbf{G}\tilde{\mathbf{W}}\mathbb{E}\left[\mathbf{A}_1^2\right]\tilde{\mathbf{W}}\mathbf{G}\right) \\
& \leq 6(\lambda_1^2 + v)\text{Tr}\left(\tilde{\mathbf{W}}^2\mathbf{G}^2\right) \leq 6(\lambda_1^2 + v)\text{Tr}\left(\mathbf{G}^2\right) \\
3^{rd} \text{ order: } & \text{Tr}\left(\mathbb{E}\left[\mathbf{A}_1\tilde{\mathbf{W}}\mathbf{G}\mathbf{A}_1\tilde{\mathbf{W}}\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1\right] + \mathbb{E}\left[\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1^2\tilde{\mathbf{W}}\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1\right] + \mathbb{E}\left[\mathbf{A}_1\tilde{\mathbf{W}}\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1^2\tilde{\mathbf{W}}\mathbf{G}\right] + \right. \\
& \quad \left.\mathbb{E}\left[\mathbf{A}_1\tilde{\mathbf{W}}\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1\right]\right) \\
& = 2\text{Tr}\left(\mathbb{E}\left[\mathbf{A}_1\tilde{\mathbf{W}}\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1^2\tilde{\mathbf{W}}\mathbf{G}\right] + \mathbb{E}\left[\tilde{\mathbf{W}}\mathbf{A}_1\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1^2\tilde{\mathbf{W}}\mathbf{G}\right]\right) \\
& \leq 2\|\mathbf{A}_1\tilde{\mathbf{W}}\|\text{Tr}\left(\mathbf{G}\tilde{\mathbf{W}}\mathbb{E}\left[\mathbf{A}_1^2\right]\tilde{\mathbf{W}}\mathbf{G}\right) + 2\|\tilde{\mathbf{W}}\mathbf{A}_1\|\text{Tr}\left(\mathbf{G}\tilde{\mathbf{W}}\mathbb{E}\left[\mathbf{A}_1^2\right]\tilde{\mathbf{W}}\mathbf{G}\right) \\
& \leq 2\|\mathbf{A}_1\|(\lambda_1^2 + v)\text{Tr}\left(\tilde{\mathbf{W}}^2\mathbf{G}^2\right) + 2\|\mathbf{A}_1\|(\lambda_1^2 + v)\text{Tr}\left(\tilde{\mathbf{W}}^2\mathbf{G}^2\right) \\
& \leq 4r(\lambda_1^2 + v)\text{Tr}\left(\mathbf{G}^2\right) \quad \text{since } \|\mathbf{A}_{i,t}\| \leq r \\
4^{th} \text{ order: } & \text{Tr}\left(\mathbb{E}\left[\mathbf{A}_1\tilde{\mathbf{W}}\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1^2\tilde{\mathbf{W}}\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1\right]\right) = \text{Tr}\left(\mathbb{E}\left[\mathbf{A}_1^2\tilde{\mathbf{W}}\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1^2\tilde{\mathbf{W}}\mathbf{G}\tilde{\mathbf{W}}\right]\right) \\
& \leq \mathbb{E}\left[\|\mathbf{A}_1^2\|\text{Tr}\left(\tilde{\mathbf{W}}\mathbf{G}\tilde{\mathbf{W}}\mathbf{A}_1^2\tilde{\mathbf{W}}\mathbf{G}\tilde{\mathbf{W}}\right)\right] \\
& \leq r^2\text{Tr}\left(\tilde{\mathbf{W}}\mathbf{G}\tilde{\mathbf{W}}\mathbb{E}\left[\mathbf{A}_1^2\right]\tilde{\mathbf{W}}\mathbf{G}\tilde{\mathbf{W}}\right) \leq r^2(\lambda_1^2 + v)\text{Tr}\left(\mathbf{G}^2\right)
\end{aligned}$$

Thus,

$$\begin{aligned}
& \text{Tr} \left(\mathbb{E} \left[(\mathbf{I} + \alpha_1 \mathbf{A}_1 \tilde{\mathbf{W}}) \mathbf{G} (\mathbf{I} + \alpha_1 \tilde{\mathbf{W}} \mathbf{A}_1) (\mathbf{I} + \alpha_1 \mathbf{A}_1 \tilde{\mathbf{W}}) \mathbf{G} (\mathbf{I} + \alpha_1 \tilde{\mathbf{W}} \mathbf{A}_1) \right] \right) \\
& \leq (1 + 4\alpha_1 \lambda_1 + 6\alpha_1^2 (\lambda_1^2 + v) + 4\alpha_1^3 r (\lambda_1^2 + v) + \alpha_1^4 r^2 (\lambda_1^2 + v)) \text{Tr}(\mathbf{G}^2) \\
& \leq (1 + 4\alpha_1 \lambda_1 + 10\alpha_1^2 \bar{v}) \text{Tr}(\mathbf{G}^2) \leq \exp(4\alpha_1 \lambda_1 + 10\alpha_1^2 \bar{v}) \text{Tr}(\mathbf{G}^2),
\end{aligned}$$

where in the last line we used that $\alpha_t \leq \frac{1}{4r}$ and $1 + x \leq \exp(x)$. Now plugging the above in (5.14) and using $\mathbf{G} = \mathbf{G}_{t-1} = \mathbf{H}_{t,t-1}^T \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t-1}$, we have

$$\eta_t = \text{Tr} \left(\mathbb{E} \left[(\mathbf{I} + \alpha_1 \mathbf{A}_1 \tilde{\mathbf{W}}) \mathbf{G}_{t-1} (\mathbf{I} + \alpha_1 \tilde{\mathbf{W}} \mathbf{A}_1) (\mathbf{I} + \alpha_1 \mathbf{A}_1 \tilde{\mathbf{W}}) \mathbf{G}_{t-1} (\mathbf{I} + \alpha_1 \tilde{\mathbf{W}} \mathbf{A}_1) \right] \right) \quad (5.16)$$

$$\leq \exp(4\alpha_1 \lambda_1 + 10\alpha_1^2 \bar{v}) \text{Tr}(\mathbf{G}_{t-1}^2) = \exp(4\alpha_1 \lambda_1 + 10\alpha_1^2 \bar{v}) \eta_{t-1} \quad (5.17)$$

Since $\eta_0 = 1$, we have

$$\mathbb{E} \left[\left(\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} \mathbf{H}_{t,t}^T \tilde{\mathbf{q}}_1 \right)^2 \right] \leq \exp \left(\sum_{m \in [t]} 4\alpha_m \lambda_1 + 10\alpha_m^2 \bar{v} \right) \quad (5.18)$$

Note that $\mathbf{D}_t = \mathbf{H}_{t,t}(\mathbf{1} \otimes \mathbf{I})$. Thus

$$\begin{aligned}
\mathbb{E} \left[\left(\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right)^2 \right] &= \text{Tr} \left(\mathbb{E} \left[\left(\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} (\mathbf{1} \mathbf{1}^T \otimes \mathbf{I}) \mathbf{H}_{t,t}^T \tilde{\mathbf{q}}_1 \right)^2 \right] \right) \\
&= \text{Tr} \left(\mathbb{E} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} (\mathbf{1} \mathbf{1}^T \otimes \mathbf{I}) \mathbf{H}_{t,t}^T \tilde{\mathbf{q}}_1 \frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} (\mathbf{1} \mathbf{1}^T \otimes \mathbf{I}) \mathbf{H}_{t,t}^T \tilde{\mathbf{q}}_1 \right] \right) \\
&= \text{Tr} \left(\mathbb{E} \left[(\mathbf{1} \mathbf{1}^T \otimes \mathbf{I}) \mathbf{H}_{t,t}^T \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} (\mathbf{1} \mathbf{1}^T \otimes \mathbf{I}) \mathbf{H}_{t,t}^T \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} \right] \right) \\
&\leq \mathbb{E} \left[\|\mathbf{1} \mathbf{1}^T \otimes \mathbf{I}\| \text{Tr} \left(\mathbf{H}_{t,t}^T \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} (\mathbf{1} \mathbf{1}^T \otimes \mathbf{I}) \mathbf{H}_{t,t}^T \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} \right) \right] \\
&\leq M \mathbb{E} \left[\text{Tr} \left((\mathbf{1} \mathbf{1}^T \otimes \mathbf{I}) \mathbf{H}_{t,t}^T \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} \mathbf{H}_{t,t}^T \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} \right) \right] \\
&\leq M^2 \mathbb{E} \left[\text{Tr} \left(\mathbf{H}_{t,t}^T \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} \mathbf{H}_{t,t}^T \frac{1}{M} \tilde{\mathbf{q}}_1 \tilde{\mathbf{q}}_1^T \mathbf{H}_{t,t} \right) \right] \\
&= M^2 \eta_t \leq M^2 \exp \left(\sum_{m \in [t]} 4\alpha_m \lambda_1 + 10\alpha_m^2 \bar{v} \right)
\end{aligned}$$

□

Theorem 9. For any fixed $\delta > 0$ and $\alpha_t = \frac{\eta}{(\lambda_1 - \tilde{\lambda}_2)(\beta + t)}$ for $\eta > 0.5$ and

$$\beta = 20 \max \left(\frac{r\eta}{(\lambda_1 - \tilde{\lambda}_2)}, \frac{(v + \lambda_1^2)\eta^2}{(\lambda_1 - \tilde{\lambda}_2)^2 \log(1 + \frac{\delta}{100})} \right).$$

Then the (concatenated) output of the algorithm \mathbf{x}_t converges to $\tilde{\mathbf{q}}_1 = \mathbf{1} \otimes \mathbf{q}_1$ as follows:

$$\sin^2(\tilde{\mathbf{q}}_1, \mathbf{x}_t) \leq \frac{C' \log(1/\delta)}{\delta^3} \left(d\left(\frac{\beta}{t}\right)^{2\eta} + \frac{v(\beta + 1)^2 \eta^2}{t\beta^2(2\eta - 1)(\lambda_1 - \tilde{\lambda}_2)^2} \right),$$

with probability at least $1 - \delta$, where C' is an absolute constant.

Proof. As discussed earlier, to prove convergence of the distributed Oja's method described in Algorithm 3, we bound the terms $\frac{1}{M}\tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1$ and $\text{Tr}(\mathbf{Q}_\perp^T \mathbf{D}_t \mathbf{D}_t^T \mathbf{Q}_\perp)$ that help bound the error. First, using Chebyshev's inequality, we get

$$P \left[\left| \frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 - \mathbb{E} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right] \right| \geq \frac{1}{\sqrt{\delta}} \sqrt{\text{Var} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right]} \right] < \delta.$$

So, with probability at least $1 - \delta$, the following holds:

$$\begin{aligned} & \left| \frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 - \mathbb{E} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right] \right| \leq \frac{1}{\sqrt{\delta}} \sqrt{\text{Var} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right]} \\ \text{i.e., } & \frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 - \mathbb{E} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right] \geq -\frac{1}{\sqrt{\delta}} \sqrt{\text{Var} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right]} \\ \text{i.e., } & \frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \geq \mathbb{E} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right] - \frac{1}{\sqrt{\delta}} \sqrt{\text{Var} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right]} \\ \text{i.e., } & \frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \geq \mathbb{E} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right] - \frac{1}{\sqrt{\delta}} \sqrt{\mathbb{E} \left[\left(\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right)^2 \right] - \mathbb{E} \left[\frac{1}{M} \tilde{\mathbf{q}}_1^T \mathbf{D}_t \mathbf{D}_t^T \tilde{\mathbf{q}}_1 \right]^2} \\ & \geq \text{Mexp} \left(2\lambda_1 \sum_{m \in [t]} \alpha_m - 4\lambda_1^2 \sum_{m \in [t]} \alpha_m^2 \right) - \\ & \quad \frac{M}{\sqrt{\delta}} \sqrt{\exp \left(4\lambda_1 \sum_{m \in [t]} \alpha_m + 10\bar{v} \sum_{m \in [t]} \alpha_m^2 \right) - \exp^2 \left(2\lambda_1 \sum_{m \in [t]} \alpha_m - 4\lambda_1^2 \sum_{m \in [t]} \alpha_m^2 \right)} \\ & = \text{Mexp} \left(2\lambda_1 \sum_{m \in [t]} \alpha_m - 4\lambda_1^2 \sum_{m \in [t]} \alpha_m^2 \right) - \\ & \quad \frac{M}{\sqrt{\delta}} \sqrt{\exp \left(4\lambda_1 \sum_{m \in [t]} \alpha_m + 10\bar{v} \sum_{m \in [t]} \alpha_m^2 \right) - \exp \left(4\lambda_1 \sum_{m \in [t]} \alpha_m - 8\lambda_1^2 \sum_{m \in [t]} \alpha_m^2 \right)} \\ & = \text{Mexp} \left(2\lambda_1 \sum_{m \in [t]} \alpha_m - 4\lambda_1^2 \sum_{m \in [t]} \alpha_m^2 \right) \left(1 - \frac{1}{\sqrt{\delta}} \sqrt{\exp \left((10\bar{v} + 8\lambda_1^2) \sum_{m \in [t]} \alpha_m^2 \right) - 1} \right) \\ & \geq \text{Mexp} \left(2\lambda_1 \sum_{m \in [t]} \alpha_m - 4\lambda_1^2 \sum_{m \in [t]} \alpha_m^2 \right) \left(1 - \frac{1}{\sqrt{\delta}} \sqrt{\exp \left(18\bar{v} \sum_{m \in [t]} \alpha_m^2 \right) - 1} \right) \end{aligned}$$

Also, using Markov's inequality, we have

$$P \left[\text{Tr}(\mathbf{Q}_\perp^T \mathbf{D}_t \mathbf{D}_t^T \mathbf{Q}_\perp) \geq \frac{1}{\delta} \mathbb{E} [\text{Tr}(\mathbf{Q}_\perp^T \mathbf{D}_t \mathbf{D}_t^T \mathbf{Q}_\perp)] \right] \leq \delta$$

Thus, with probability at least $1 - \delta$, we have

$$\begin{aligned} \text{Tr}(\mathbf{Q}_\perp^T \mathbf{D}_t \mathbf{D}_t^T \mathbf{Q}_\perp) & \leq \frac{1}{\delta} \mathbb{E} [\text{Tr}(\mathbf{Q}_\perp^T \mathbf{D}_t \mathbf{D}_t^T \mathbf{Q}_\perp)] \\ & \leq \frac{M}{\delta} \exp \left(\sum_{m \in [t]} 2\alpha_m \tilde{\lambda}_2 + \alpha_m^2 \bar{v} \right) \left(d + \sum_{p \in [t]} \alpha_p^2 v \exp \left(\sum_{m \in [p]} 2\alpha_m (\lambda_1 - \tilde{\lambda}_2) \right) \right). \end{aligned}$$

Using Lemma 17, we have

$$\begin{aligned}
\sin^2(\tilde{\mathbf{q}}_1, \mathbf{x}_t) &\leq \frac{C \log(1/\delta)}{\delta^3} \frac{\exp\left(2\tilde{\lambda}_2 \sum_{m \in [t]} \alpha_m + \bar{v} \sum_{m \in [t]} \alpha_m^2\right) \left(d + v \sum_{m \in [t]} \alpha_m^2 \exp\left(2(\lambda_1 - \tilde{\lambda}_2) \sum_{p \in [m]} \alpha_p\right)\right)}{\exp\left(2\lambda_1 \sum_{m \in [t]} \alpha_m - 4\lambda_1^2 \sum_{m \in [t]} \alpha_m^2\right) \left(1 - \frac{1}{\sqrt{\delta}} \sqrt{\exp\left(18\bar{v} \sum_{m \in [t]} \alpha_m^2\right) - 1}\right)} \\
&= \frac{C \log(1/\delta)}{\delta^3} \frac{\exp\left(2(\tilde{\lambda}_2 - \lambda_1) \sum_{m \in [t]} \alpha_m + (\bar{v} + 4\lambda_1^2) \sum_{m \in [t]} \alpha_m^2\right) \left(d + v \sum_{m \in [t]} \alpha_m^2 \exp\left(2(\lambda_1 - \tilde{\lambda}_2) \sum_{p \in [m]} \alpha_p\right)\right)}{\left(1 - \frac{1}{\sqrt{\delta}} \sqrt{\exp\left(18\bar{v} \sum_{m \in [t]} \alpha_m^2\right) - 1}\right)} \\
&\leq \frac{C \log(1/\delta)}{\delta^3} \frac{\exp\left(5\bar{v} \sum_{m \in [t]} \alpha_m^2\right) \exp\left(2(\tilde{\lambda}_2 - \lambda_1) \sum_{m \in [t]} \alpha_m\right) \left(d + v \sum_{m \in [t]} \alpha_m^2 \exp\left(2(\lambda_1 - \tilde{\lambda}_2) \sum_{p \in [m]} \alpha_p\right)\right)}{\left(1 - \frac{1}{\sqrt{\delta}} \sqrt{\exp\left(18\bar{v} \sum_{m \in [t]} \alpha_m^2\right) - 1}\right)} \\
&\leq \frac{C \log(1/\delta)}{\delta^3} \frac{\exp\left(5\bar{v} \sum_{m \in [t]} \alpha_m^2\right) \left(d \exp\left(2(\tilde{\lambda}_2 - \lambda_1) \sum_{m \in [t]} \alpha_m\right) + v \sum_{m \in [t]} \alpha_m^2 \exp\left(2(\tilde{\lambda}_2 - \lambda_1) \sum_{p=m+1}^t \alpha_p\right)\right)}{\left(1 - \frac{1}{\sqrt{\delta}} \sqrt{\exp\left(18\bar{v} \sum_{m \in [t]} \alpha_m^2\right) - 1}\right)}
\end{aligned}$$

Since $\alpha_t = \frac{\eta}{(\lambda_1 - \tilde{\lambda}_2)(\beta + t)}$, we have

$$18\bar{v} \sum_{m \in [t]} \alpha_m^2 = \frac{18\bar{v}\eta^2}{(\lambda_1 - \tilde{\lambda}_2)^2} \left(\frac{1}{(\beta + 1)^2} + \frac{1}{(\beta + 2)^2} + \dots + \frac{1}{(\beta + t)^2} \right) \leq \frac{18\bar{v}\eta^2}{\beta(\lambda_1 - \tilde{\lambda}_2)^2}$$

Also, from our assumption

$$\begin{aligned}
\beta &\geq \frac{20(v + \lambda_1^2)\eta^2}{(\lambda_1 - \tilde{\lambda}_2)^2 \log(1 + \frac{\delta}{100})} = \frac{20\eta^2\bar{v}}{(\lambda_1 - \tilde{\lambda}_2)^2 \log(1 + \frac{\delta}{100})} > \frac{18\eta^2\bar{v}}{(\lambda_1 - \tilde{\lambda}_2)^2 \log(1 + \frac{\delta}{100})} \\
\text{i.e., } \frac{\beta(\lambda_1 - \tilde{\lambda}_2)^2}{18\bar{v}\eta^2} &\geq \frac{1}{\log(1 + \frac{\delta}{100})} \implies \frac{18\bar{v}\eta^2}{\beta(\lambda_1 - \tilde{\lambda}_2)^2} \leq \log(1 + \frac{\delta}{100})
\end{aligned}$$

Thus,

$$\begin{aligned}
18\bar{v} \sum_{m \in [t]} \alpha_m^2 &\leq \log(1 + \frac{\delta}{100}) \\
\text{i.e., } \exp\left(18\bar{v} \sum_{m \in [t]} \alpha_m^2\right) &\leq 1 + \frac{\delta}{100} \\
\text{i.e., } \exp\left(18\bar{v} \sum_{m \in [t]} \alpha_m^2\right) - 1 &\leq \frac{\delta}{100} \\
\text{i.e., } \sqrt{\exp\left(18\bar{v} \sum_{m \in [t]} \alpha_m^2\right) - 1} &\leq \frac{\sqrt{\delta}}{10} \\
\text{i.e., } \frac{1}{\sqrt{\delta}} \sqrt{\exp\left(18\bar{v} \sum_{m \in [t]} \alpha_m^2\right) - 1} &\leq 0.1 \\
\text{i.e., } 1 - \frac{1}{\sqrt{\delta}} \sqrt{\exp\left(18\bar{v} \sum_{m \in [t]} \alpha_m^2\right) - 1} &\geq 0.9 \\
\text{i.e., } \frac{1}{1 - \frac{1}{\sqrt{\delta}} \sqrt{\exp\left(18\bar{v} \sum_{m \in [t]} \alpha_m^2\right) - 1}} &\leq 1.11
\end{aligned}$$

Similarly,

$$\exp\left(5\bar{v} \sum_{m \in [t]} \alpha_m^2\right) \leq 1 + \frac{\delta}{100} \leq 1.01$$

Thus,

$$\sin^2(\tilde{\mathbf{q}}_1, \mathbf{x}_t) \leq \frac{C' \log(1/\delta)}{\delta^3} \left(d \exp\left(2(\tilde{\lambda}_2 - \lambda_1) \sum_{m \in [t]} \alpha_m\right) + v \sum_{m \in [t]} \alpha_m^2 \exp\left(2(\tilde{\lambda}_2 - \lambda_1) \sum_{p=m+1}^t \alpha_p\right) \right) \quad (5.19)$$

Now, since $\sum_{m \in [t]} \alpha_m$ is partial harmonic series, we have

$$\begin{aligned} \sum_{m \in [t]} \alpha_m &\geq \frac{\eta}{(\lambda_1 - \tilde{\lambda}_2)} \log\left(1 + \frac{t}{\beta}\right) \\ \text{i.e., } 2(\lambda_1 - \tilde{\lambda}_2) \sum_{m \in [t]} \alpha_m &\geq 2\eta \log\left(\frac{\beta + t}{\beta}\right) \\ \text{i.e., } 2(\tilde{\lambda}_2 - \lambda_1) \sum_{m \in [t]} \alpha_m &\leq 2\eta \log\left(\frac{\beta}{\beta + t}\right) \\ \text{i.e., } \exp\left(2(\tilde{\lambda}_2 - \lambda_1) \sum_{m \in [t]} \alpha_m\right) &\leq \left(\frac{\beta}{\beta + t}\right)^{2\eta} \end{aligned} \quad (5.20)$$

Also,

$$\begin{aligned} \sum_{p=m+1}^t \alpha_p &\geq \frac{\eta}{(\lambda_1 - \tilde{\lambda}_2)} \log\left(\frac{\beta + t + 1}{\beta + m + 1}\right) \\ \text{i.e., } 2(\tilde{\lambda}_2 - \lambda_1) \sum_{p=m+1}^t \alpha_p &\leq 2\eta \log\left(\frac{\beta + m + 1}{\beta + t + 1}\right) \\ \text{i.e., } \exp\left(2(\tilde{\lambda}_2 - \lambda_1) \sum_{p=m+1}^t \alpha_p\right) &\leq \left(\frac{\beta + m + 1}{\beta + t + 1}\right)^{2\eta} \end{aligned}$$

Thus,

$$\begin{aligned} \sum_{m \in [t]} \alpha_m^2 \exp\left(2(\tilde{\lambda}_2 - \lambda_1) \sum_{p=m+1}^t \alpha_p\right) &\leq \frac{\eta^2}{(\lambda_1 - \tilde{\lambda}_2)^2} \sum_{m \in [t]} \frac{1}{(\beta + m)^2} \left(\frac{\beta + m + 1}{\beta + t + 1}\right)^{2\eta} \\ &= \frac{\eta^2}{(\lambda_1 - \tilde{\lambda}_2)^2 (\beta + t + 1)^{2\eta}} \sum_{m \in [t]} \frac{1}{(\beta + m)^2} (\beta + m + 1)^{2\eta} \\ &= \frac{\eta^2}{(\lambda_1 - \tilde{\lambda}_2)^2 (\beta + t + 1)^{2\eta}} \sum_{m \in [t]} \left(1 + \frac{1}{\beta + m}\right)^2 (\beta + m + 1)^{2\eta-2} \\ &\leq \frac{\eta^2}{(\lambda_1 - \tilde{\lambda}_2)^2 (\beta + t + 1)^{2\eta}} \sum_{m \in [t]} \left(1 + \frac{1}{\beta}\right)^2 (\beta + m + 1)^{2\eta-2} \\ &\leq \frac{(\beta + 1)^2 \eta^2}{\beta^2 (\lambda_1 - \tilde{\lambda}_2)^2 (\beta + t + 1)^{2\eta}} \sum_{m \in [t]} (\beta + m + 1)^{2\eta-2} \\ &\stackrel{\zeta_1}{\leq} \frac{(\beta + 1)^2 \eta^2}{\beta^2 (\lambda_1 - \tilde{\lambda}_2)^2 (\beta + t + 1)^{2\eta}} \frac{(\beta + t + 1)^{2\eta-1}}{2\eta - 1} \\ &= \frac{(\beta + 1)^2 \eta^2}{\beta^2 (2\eta - 1) (\lambda_1 - \tilde{\lambda}_2)^2 (\beta + t + 1)}, \end{aligned} \quad (5.21)$$

where ζ_1 is by bounding the sum using the corresponding integral. Plugging (5.20) and (5.21) in (5.19), we get

$$\begin{aligned} \sin^2(\tilde{\mathbf{q}}_1, \mathbf{x}_t) &\leq \frac{C' \log(1/\delta)}{\delta^3} \left(d \left(\frac{\beta}{\beta+t} \right)^{2\eta} + \frac{v(\beta+1)^2 \eta^2}{\beta^2 (2\eta-1) (\lambda_1 - \tilde{\lambda}_2)^2 (\beta+t+1)} \right) \\ &\leq \frac{C' \log(1/\delta)}{\delta^3} \left(d \left(\frac{\beta}{t} \right)^{2\eta} + \frac{v(\beta+1)^2 \eta^2}{t \beta^2 (2\eta-1) (\lambda_1 - \tilde{\lambda}_2)^2} \right) \end{aligned}$$

Thus the error decreases at each node of the network as $t \rightarrow \infty$. This analysis proves convergence of the proposed algorithm for the case of $K = 1$. Although the speed up in the convergence due to collaboration in the network does not show up in the result, there is indeed an improvement in the convergence rate which we will demonstrate through experiments in the Section 5.6. In the next section, we provide the convergence analysis of our proposed algorithm for a special case, when exact averaging is possible at the nodes. This setting typically arises in a federated learning setup. \square

5.5 Distributed PCA in a Federated Learning Setup

5.5.1 Problem Setup

This section describes the convergence analysis of the Distributed Oja's Algorithm for the estimation of the dominant eigenvector of the population covariance matrix Σ in a federated learning setting. This setting assumes a master-slave kind of architecture, where nodes send their local estimates to a central master node which then takes the average of those local estimates and sends it back to all the nodes. The update equation at every node in this setting is:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \alpha_t \frac{1}{M} \sum_{i=1}^M \mathbf{y}_{i,t} \mathbf{y}_{i,t}^T \mathbf{x}_{t-1} \quad (5.22)$$

$$\mathbf{x}_t = \frac{\mathbf{x}_t}{\|\mathbf{x}_t\|}, \quad (5.23)$$

Note that this is significantly different from DIEGO in one major aspect. The estimate at each node \mathbf{x}_t will be same after each iteration due to exact averaging.

5.5.2 Convergence Analysis

The convergence analysis for the algorithm (5.22) and (5.23) is similar to the proof of DIEGO given in Section 5.4. Since we are looking at convergence in terms of angle between estimated

and true eigenvector, we can push the normalization to the end. In other words, we can analyze the following update equation:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \alpha_t \frac{1}{M} \sum_{i=1}^M \mathbf{y}_{i,t} \mathbf{y}_{i,t}^T \mathbf{x}_{t-1} \quad (5.24)$$

$$= \mathbf{x}_{t-1} + \alpha_t \frac{1}{M} \sum_{i=1}^M \mathbf{A}_{i,t} \mathbf{x}_{t-1} \quad (5.25)$$

where $\mathbf{A}_{i,t} = \mathbf{y}_{i,t} \mathbf{y}_{i,t}^T$ and α_t is the decreasing step-size. Here, let $\mathbf{A}_t = \frac{1}{M} \sum_{i=1}^M \mathbf{A}_{i,t}$. Under our assumptions

$$1. \quad \|\mathbf{A}_{i,t}\| \leq r$$

$$2. \quad \left\| \mathbb{E} \left[(\mathbf{A}_{i,t} - \Sigma)(\mathbf{A}_{i,t} - \Sigma)^T \right] \right\| \leq v,$$

we have $\mathbb{E}[\mathbf{A}_t] = \Sigma$ and $\left\| \mathbb{E} \left[(\mathbf{A}_t - \Sigma)(\mathbf{A}_t - \Sigma)^T \right] \right\| \leq \frac{v}{M}$. Re-writing (5.24), we have

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \alpha_t \mathbf{A}_t \mathbf{x}_{t-1}, \quad (5.26)$$

We can view our algorithm as applying the matrix

$$\mathbf{B}_t = (\mathbf{I} + \alpha_t \mathbf{A}_t)(\mathbf{I} + \alpha_{t-1} \mathbf{A}_{t-1}) \dots (\mathbf{I} + \alpha_1 \mathbf{A}_1) \quad (5.27)$$

on \mathbf{x}_0 and giving an output as

$$\mathbf{x}_t = \frac{\mathbf{B}_t \mathbf{x}_0}{\|\mathbf{B}_t \mathbf{x}_0\|}. \quad (5.28)$$

Thus, (5.27) and (5.28) can be viewed as one step power method for \mathbf{B}_t .

Lemma 22. *Let $\mathbf{B} \in \mathbb{R}^{d \times d}$ and let $\mathbf{v} \in \mathbb{R}^d$ be a unit vector. Let \mathbf{V}_\perp be the matrix whose columns form the space orthogonal to \mathbf{v} . If $\mathbf{x} \in \mathbb{R}^d$ is chosen uniformly at random from the surface of the unit sphere, then with probability at least $1 - \delta$*

$$\sin^2 \left(\mathbf{v}, \frac{\mathbf{B}\mathbf{x}}{\|\mathbf{B}\mathbf{x}\|} \right) = 1 - \left(\frac{\mathbf{v}^T \mathbf{B}\mathbf{x}}{\|\mathbf{B}\mathbf{x}\|} \right)^2 \leq \frac{C \log(1/\delta)}{\delta^2} \frac{\text{Tr}(\mathbf{V}_\perp^T \mathbf{B} \mathbf{B}^T \mathbf{V}_\perp)}{\mathbf{v}^T \mathbf{B} \mathbf{B}^T \mathbf{v}}$$

Proof. As \mathbf{x} is uniformly distributed over the unit sphere, we can say $\mathbf{x} = \frac{\mathbf{g}}{\|\mathbf{g}\|}$ where $\mathbf{g} \sim \mathcal{N}(0, \mathbf{I})$.

$$\begin{aligned} 1 - \left(\frac{\mathbf{v}^T \mathbf{B}\mathbf{x}}{\|\mathbf{B}\mathbf{x}\|} \right)^2 &= \frac{\mathbf{x}^T \mathbf{B}^T \mathbf{B} \mathbf{x} - \mathbf{x}^T \mathbf{B}^T \mathbf{v} \mathbf{v}^T \mathbf{B} \mathbf{x}}{\mathbf{x}^T \mathbf{B}^T \mathbf{B} \mathbf{x}} \\ &= \frac{\mathbf{x}^T \mathbf{B}^T (\mathbf{I} - \mathbf{v} \mathbf{v}^T) \mathbf{B} \mathbf{x}}{\mathbf{x}^T \mathbf{B}^T \mathbf{B} \mathbf{x}} = \frac{\mathbf{g}^T \mathbf{B}^T (\mathbf{I} - \mathbf{v} \mathbf{v}^T) \mathbf{B} \mathbf{g}}{\mathbf{g}^T \mathbf{B}^T \mathbf{B} \mathbf{g}} \\ &\leq \frac{C}{\delta^2} \frac{\mathbf{g}^T \mathbf{B}^T (\mathbf{I} - \mathbf{v} \mathbf{v}^T) \mathbf{B} \mathbf{g}}{\mathbf{v}^T \mathbf{B} \mathbf{B}^T \mathbf{v}} \\ &\leq \frac{C \log(1/\delta)}{\delta^2} \frac{\text{Tr}(\mathbf{B}^T (\mathbf{I} - \mathbf{v} \mathbf{v}^T) \mathbf{B})}{\mathbf{v}^T \mathbf{D} \mathbf{D}^T \mathbf{v}}, \end{aligned}$$

where C is an absolute constant. ζ_1 follows from the fact that $\mathbf{g}^T \mathbf{B}^T \mathbf{B} \mathbf{g} \geq (\mathbf{v}^T \mathbf{B} \mathbf{g})^2 \geq \frac{\delta^2}{C} \mathbf{v}^T \mathbf{B} \mathbf{B}^T \mathbf{v}$, where the first inequality is Cauchy Schwarz and the second inequality follows from the fact that $\mathbf{v}^T \mathbf{B} \mathbf{g}$ is a Gaussian random variable with variance $\|\mathbf{B}^T \mathbf{v}\|^2$ and $\Pr(|g| \leq \delta) \leq C\delta$ for a normal random variable $g \sim \mathcal{N}(0, 1)$. Similarly, ζ_2 follows from the fact that $\mathbf{g}^T \mathbf{B}^T (\mathbf{I} - \mathbf{v} \mathbf{v}^T) \mathbf{B} \mathbf{g}$ is a χ^2 random variable with $\text{Tr}(\mathbf{B}^T (\mathbf{I} - \tilde{\mathbf{v}} \tilde{\mathbf{v}}^T) \mathbf{B})$ degrees of freedom. \square

Let $\mathbf{q}_1, \dots, \mathbf{q}_d$ denote the eigenvectors of Σ and $\lambda_1 > \lambda_2 \geq \dots \geq \lambda_d$ be the corresponding eigenvalues. Evidently, if \mathbf{Q}_\perp is the matrix of orthogonal columns that span the subspace orthogonal to \mathbf{q}_1 , then $\mathbf{Q}_\perp \mathbf{Q}_\perp^T = \mathbf{I} - \mathbf{q}_1 \mathbf{q}_1^T$. Lemma 22 shows that to prove the convergence of distributed Oja's algorithm, we need two important pieces. First, we need to show that with constant probability $\mathbf{q}_1^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{q}_1$ is relatively large and second, $\text{Tr}(\mathbf{B}_t^T \mathbf{Q}_\perp \mathbf{Q}_\perp^T \mathbf{B}_t) = \text{Tr}(\mathbf{Q}_\perp^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{Q}_\perp)$ is relatively small.

Lemma 23. *For all $t \geq 0$ and $\alpha_t \geq 0$, we have*

$$\|\mathbb{E} [\mathbf{B}_t \mathbf{B}_t^T]\| \leq \exp\left(\sum_{m \in [t]} 2\alpha_m \lambda_1 + \alpha_m^2 \bar{v}\right),$$

where $\bar{v} = \frac{v}{M} + \lambda_1^2$.

Proof. Let $\eta_t = \|\mathbb{E} [\mathbf{B}_t \mathbf{B}_t^T]\|$, i.e., $\mathbb{E} [\mathbf{B}_t \mathbf{B}_t^T] \preceq \eta_t \mathbf{I}$. Now, for all $t > 0$,

$$\begin{aligned} \mathbb{E} [\mathbf{B}_t \mathbf{B}_t^T] &= \mathbb{E} [(\mathbf{I} + \alpha_t \mathbf{A}_t) \mathbf{B}_{t-1} \mathbf{B}_{t-1}^T (\mathbf{I} + \alpha_t \mathbf{A}_t)^T] \\ &\preceq \eta_{t-1} \mathbb{E} [(\mathbf{I} + \alpha_t \mathbf{A}_t)(\mathbf{I} + \alpha_t \mathbf{A}_t)^T] \\ &= \eta_{t-1} \mathbb{E} [\mathbf{I} + 2\alpha_t \mathbf{A}_t + \alpha_t^2 \mathbf{A}_t^2] \\ &= \eta_{t-1} \left[\mathbf{I} + 2\alpha_t \Sigma + \alpha_t^2 \mathbb{E} [\mathbf{A}_t^2] \right] \end{aligned} \tag{5.29}$$

Since $\left\| \mathbb{E} [(\mathbf{A}_t - \Sigma)(\mathbf{A}_t - \Sigma)^T] \right\| \leq \frac{v}{M}$, we have

$$\mathbb{E} [\mathbf{A}_t^2] = \Sigma^2 + \mathbb{E} [(\mathbf{A}_t - \Sigma)(\mathbf{A}_t - \Sigma)^T] \preceq \Sigma^2 + \frac{v}{M} \mathbf{I}$$

Using the above inequality in (5.29), we get

$$\mathbb{E} [\mathbf{B}_t \mathbf{B}_t^T] \preceq \eta_{t-1} \left[\mathbf{I} + 2\alpha_t \Sigma + \alpha_t^2 (\Sigma^2 + \frac{v}{M} \mathbf{I}) \right]$$

We know $\|\Sigma\| = \lambda_1$ and $\|\Sigma^2\| = \lambda_1^2$. Therefore,

$$\eta_t \leq \eta_{t-1} \left(1 + 2\alpha_t \lambda_1 + \alpha_t^2 (\lambda_1^2 + \frac{v}{M}) \right) = \eta_{t-1} (1 + 2\alpha_t \lambda_1 + \alpha_t^2 \bar{v})$$

Using the fact that $\mathbf{B}_0 = \mathbf{I}$, i.e., $\eta_0 = 1$ and $1 + x \leq \exp(x)$, the result follows. \square

Using Lemma 23 we next bound $\mathbb{E} \left[\text{Tr}(\mathbf{Q}_\perp^\text{T} \mathbf{B}_t \mathbf{B}_t^\text{T} \mathbf{Q}_\perp) \right]$. This will help us in bounding $\text{Tr}(\mathbf{Q}_\perp^\text{T} \mathbf{B}_t \mathbf{B}_t^\text{T} \mathbf{Q}_\perp)$ using Markov's inequality.

Lemma 24. *For all $t \geq 0$ and $\alpha_t \leq \frac{1}{\lambda_1}$,*

$$\mathbb{E} \left[\text{Tr}(\mathbf{Q}_\perp^\text{T} \mathbf{B}_t \mathbf{B}_t^\text{T} \mathbf{Q}_\perp) \right] \leq \exp \left(\sum_{m \in [t]} 2\alpha_m \lambda_2 + \alpha_m^2 \bar{v} \right) \left(d + \sum_{p \in [t]} \alpha_p^2 \exp \left(\sum_{m \in [p]} 2\alpha_m (\lambda_1 - \lambda_2) \right) \right).$$

Proof. Let $\eta_t = \mathbb{E} \left[\text{Tr}(\mathbf{Q}_\perp^\text{T} \mathbf{B}_t \mathbf{B}_t^\text{T} \mathbf{Q}_\perp) \right] = \left\langle \mathbb{E} \left[\mathbf{B}_t \mathbf{B}_t^\text{T} \right], \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} \right\rangle$. Now,

$$\begin{aligned} \mathbb{E} \left[\text{Tr}(\mathbf{Q}_\perp^\text{T} \mathbf{B}_t \mathbf{B}_t^\text{T} \mathbf{Q}_\perp) \right] &= \mathbb{E} \left[\text{Tr}(\mathbf{Q}_\perp^\text{T} (\mathbf{I} + \alpha_t \mathbf{A}_t) \mathbf{B}_{t-1} \mathbf{B}_{t-1}^\text{T} (\mathbf{I} + \alpha_t \mathbf{A}_t)^\text{T} \mathbf{Q}_\perp) \right] \\ &= \mathbb{E} \left\langle \mathbf{B}_{t-1} \mathbf{B}_{t-1}^\text{T}, (\mathbf{I} + \alpha_t \mathbf{A}_t) \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} (\mathbf{I} + \alpha_t \mathbf{A}_t)^\text{T} \right\rangle \\ &= \left\langle \mathbb{E} \left[\mathbf{B}_{t-1} \mathbf{B}_{t-1}^\text{T} \right], \mathbb{E} \left[(\mathbf{I} + \alpha_t \mathbf{A}_t) \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} (\mathbf{I} + \alpha_t \mathbf{A}_t)^\text{T} \right] \right\rangle \end{aligned} \quad (5.30)$$

Now,

$$\begin{aligned} &\mathbb{E} \left[(\mathbf{I} + \alpha_t \mathbf{A}_t) \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} (\mathbf{I} + \alpha_t \mathbf{A}_t)^\text{T} \right] \\ &= \mathbb{E} \left[(\mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} + \alpha_t \mathbf{A}_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T}) (\mathbf{I} + \alpha_t \mathbf{A}_t) \right] \\ &= \mathbb{E} \left[\mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} + \alpha_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} \mathbf{A}_t + \alpha_t \mathbf{A}_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} + \alpha_t^2 \mathbf{A}_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} \mathbf{A}_t \right] \\ &= \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} + \alpha_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} \Sigma + \alpha_t \Sigma \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} + \alpha_t^2 \mathbb{E} \left[\mathbf{A}_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} \mathbf{A}_t \right] \\ &= \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} + \alpha_t \Sigma \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} + \alpha_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} \Sigma + \alpha_t^2 \Sigma \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} \Sigma + \alpha_t^2 \mathbb{E} \left[(\mathbf{A}_t - \Sigma) \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} (\mathbf{A}_t - \Sigma) \right] \\ &\preceq \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} + \alpha_t \Sigma \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} + \alpha_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} \Sigma + \alpha_t^2 \Sigma \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} \Sigma + \alpha_t^2 \mathbb{E} \left[(\mathbf{A}_t - \Sigma) (\mathbf{A}_t - \Sigma) \right] \\ &\preceq \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} + \alpha_t \Sigma \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} + \alpha_t \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} \Sigma + \alpha_t^2 \Sigma \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} \Sigma + \alpha_t^2 \frac{v}{M} \mathbf{I} \end{aligned} \quad (5.31)$$

Now, since \mathbf{Q}_\perp is the matrix of eigenvectors orthogonal to \mathbf{q}_1 and second largest eigenvalue of Σ is λ_2 , from (5.10), we get

$$\begin{aligned} \mathbb{E} \left[(\mathbf{I} + \alpha_t \mathbf{A}_t) \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} (\mathbf{I} + \alpha_t \mathbf{A}_t)^\text{T} \right] &\preceq (1 + 2\alpha_t \lambda_2 + \alpha_t^2 \lambda_2^2) \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} + \alpha_t^2 \frac{v}{M} \mathbf{I} \\ &\preceq (1 + 2\alpha_t \lambda_2 + \alpha_t^2 \lambda_1^2 + \alpha_t^2 \frac{v}{M}) \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} + \alpha_t^2 \frac{v}{M} \mathbf{q}_1 \mathbf{q}_1^\text{T} \end{aligned}$$

Plugging the above in (5.30), we get

$$\begin{aligned} \eta_t &\leq (1 + 2\alpha_t \lambda_2 + \alpha_t^2 \lambda_1^2 + \alpha_t^2 \frac{v}{M}) \left\langle \mathbb{E} \left[\mathbf{B}_{t-1} \mathbf{B}_{t-1}^\text{T} \right], \mathbf{Q}_\perp \mathbf{Q}_\perp^\text{T} \right\rangle + \alpha_t^2 \frac{v}{M} \left\langle \mathbb{E} \left[\mathbf{B}_{t-1} \mathbf{B}_{t-1}^\text{T} \right], \mathbf{q}_1 \mathbf{q}_1^\text{T} \right\rangle \\ &\leq (1 + 2\alpha_t \lambda_2 + \alpha_t^2 \bar{v}) \eta_{t-1} + \alpha_t^2 \frac{v}{M} \left\| \mathbb{E} \left[\mathbf{B}_{t-1} \mathbf{B}_{t-1}^\text{T} \right] \right\| \\ &\leq \exp(2\alpha_t \lambda_2 + \alpha_t^2 \bar{v}) \eta_{t-1} + \frac{v}{M} \alpha_t^2 \exp \left(\sum_{m \in [t-1]} 2\alpha_m \lambda_1 + \alpha_m^2 \bar{v} \right), \end{aligned} \quad (5.32)$$

where last inequality follows from $1 + x \leq \exp(x)$ and Lemma 23. Recuring (5.32), we get

$$\begin{aligned}
\eta_t &\leq \exp\left(\sum_{m \in [t]} 2\alpha_m \lambda_2 + \alpha_m^2 \bar{v}\right) \eta_0 + \frac{v}{M} \sum_{p \in [t]} \alpha_p^2 \exp\left(\sum_{m \in [p]} 2\alpha_m \lambda_1 + \alpha_m^2 \bar{v}\right) \exp\left(\sum_{m=p+1}^t 2\alpha_m \lambda_2 + \alpha_m^2 \bar{v}\right) \\
&= \exp\left(\sum_{m \in [t]} 2\alpha_m \lambda_2 + \alpha_m^2 \bar{v}\right) \left(\eta_0 + \frac{v}{M} \sum_{p \in [t]} \alpha_p^2 \exp\left(\sum_{m \in [p]} 2\alpha_m \lambda_1 + \alpha_m^2 \bar{v}\right) \exp\left(-\sum_{m \in [p]} 2\alpha_m \lambda_2 - \alpha_m^2 \bar{v}\right)\right) \\
&= \exp\left(\sum_{m \in [t]} 2\alpha_m \lambda_2 + \alpha_m^2 \bar{v}\right) \left(\eta_0 + \frac{v}{M} \sum_{p \in [t]} \alpha_p^2 \exp\left(\sum_{m \in [p]} 2\alpha_m (\lambda_1 - \lambda_2)\right)\right).
\end{aligned}$$

Since $\mathbf{B}_0 = \mathbf{I}$, $\eta_0 = (d-1) \leq d$. Thus,

$$\eta_t \leq \exp\left(\sum_{m \in [t]} 2\alpha_m \lambda_2 + \alpha_m^2 \bar{v}\right) \left(d + \frac{v}{M} \sum_{p \in [t]} \alpha_p^2 \exp\left(\sum_{m \in [p]} 2\alpha_m (\lambda_1 - \lambda_2)\right)\right)$$

□

Next, we lower bound $\mathbf{q}_1^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{q}_1$. In Lemma 25 we lower bound $\mathbb{E} [\mathbf{q}_1^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{q}_1]$ and in Lemma 26 we upper bound $\text{Var} [\mathbf{q}_1^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{q}_1]$.

Lemma 25. *For all $t \geq 0$ and $\alpha_t \geq 0$, we have*

$$\mathbb{E} [\mathbf{q}_1^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{q}_1] \geq \exp\left(\sum_{m \in [t]} 2\alpha_m \lambda_1 - 4\alpha_m^2 \lambda_1^2\right)$$

Proof. Let $\eta_t = \mathbb{E} [\mathbf{q}_1^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{q}_1] = \mathbb{E} [\text{Tr}(\mathbf{q}_1^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{q}_1)]$. Since $\mathbf{B}_t = (\mathbf{I} + \alpha_t \mathbf{A}_t) \mathbf{B}_{t-1}$, we have

$$\begin{aligned}
\eta_t &= \mathbb{E} [\text{Tr}(\mathbf{q}_1^T (\mathbf{I} + \alpha_t \mathbf{A}_t) \mathbf{B}_{t-1} \mathbf{B}_{t-1}^T (\mathbf{I} + \alpha_t \mathbf{A}_t)^T \mathbf{q}_1)] \\
&= \left\langle \mathbb{E} [\mathbf{B}_{t-1} \mathbf{B}_{t-1}^T], \mathbb{E} [(\mathbf{I} + \alpha_t \mathbf{A}_t) \mathbf{q}_1 \mathbf{q}_1^T (\mathbf{I} + \alpha_t \mathbf{A}_t)^T] \right\rangle \\
&= \left\langle \mathbb{E} [\mathbf{B}_{t-1} \mathbf{B}_{t-1}^T], \mathbf{q}_1 \mathbf{q}_1^T + \alpha_t \Sigma \mathbf{q}_1 \mathbf{q}_1^T + \alpha_t \mathbf{q}_1 \mathbf{q}_1^T \Sigma + \alpha_t^2 \mathbb{E} [\mathbf{A}_t \mathbf{q}_1 \mathbf{q}_1^T \mathbf{A}_t] \right\rangle \\
&\geq \left\langle \mathbb{E} [\mathbf{B}_{t-1} \mathbf{B}_{t-1}^T], \mathbf{q}_1 \mathbf{q}_1^T + \alpha_t \Sigma \mathbf{q}_1 \mathbf{q}_1^T + \alpha_t \mathbf{q}_1 \mathbf{q}_1^T \Sigma \right\rangle \\
&= \left\langle \mathbb{E} [\mathbf{B}_{t-1} \mathbf{B}_{t-1}^T], \mathbf{q}_1 \mathbf{q}_1^T + 2\alpha_t \lambda_1 \mathbf{q}_1 \mathbf{q}_1^T \right\rangle \\
&= (1 + 2\alpha_t \lambda_1) \left\langle \mathbb{E} [\mathbf{B}_{t-1} \mathbf{B}_{t-1}^T], \mathbf{q}_1 \mathbf{q}_1^T \right\rangle = (1 + 2\alpha_t \lambda_1) \eta_{t-1}
\end{aligned} \tag{5.33}$$

Since $\mathbf{B}_0 = \mathbf{I}$, $\eta_0 = 1$. Proceeding recursively and using the fact that $1 + x \geq \exp(x - x^2)$ for all $x \geq 0$, we get

$$\mathbb{E} [\mathbf{q}_1^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{q}_1] \geq M \exp\left(\sum_{m \in [t]} 2\alpha_m \lambda_1 - 4\alpha_m^2 \lambda_1^2\right). \tag{5.34}$$

□

Lemma 26. *For $t \geq 0$ and $\alpha_t \leq \frac{1}{4r}, \forall t$,*

$$\mathbb{E} [(\mathbf{q}_1^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{q}_1)^2] \leq \exp\left(\sum_{m \in [t]} 4\alpha_m \lambda_1 + 10\alpha_m^2 \bar{v}\right)$$

Proof. Let $\mathbf{H}_{t,s} = (\mathbf{I} + \alpha_t \mathbf{A}_t) \dots (\mathbf{I} + \alpha_{t-s+1} \mathbf{A}_{t-s+1})$ and $\eta_s = \mathbb{E} \left[(\mathbf{q}_1^T \mathbf{H}_{t,s} \mathbf{H}_{t,s}^T \mathbf{q}_1)^2 \right]$. Note that $\mathbf{H}_{t,s} = \mathbf{B}_t$ and $\eta_t = \mathbb{E} \left[(\mathbf{q}_1^T \mathbf{H}_{t,t} \mathbf{H}_{t,t}^T \mathbf{q}_1)^2 \right]$. Now,

$$\begin{aligned} \eta_t &= \text{Tr} \left(\mathbb{E} \left[(\mathbf{q}_1^T \mathbf{H}_{t,t} \mathbf{H}_{t,t}^T \mathbf{q}_1)^2 \right] \right) = \text{Tr} \left(\mathbb{E} \left[\mathbf{q}_1^T \mathbf{H}_{t,t} \mathbf{H}_{t,t}^T \mathbf{q}_1 \mathbf{q}_1^T \mathbf{H}_{t,t} \mathbf{H}_{t,t}^T \mathbf{q}_1 \right] \right) \\ &= \text{Tr} \left(\mathbb{E} \left[\mathbf{H}_{t,t}^T \mathbf{q}_1 \mathbf{q}_1^T \mathbf{H}_{t,t} \mathbf{H}_{t,t}^T \mathbf{q}_1 \mathbf{q}_1^T \mathbf{H}_{t,t} \right] \right) \\ &= \text{Tr} \left(\mathbb{E} \left[(\mathbf{I} + \alpha_1 \mathbf{A}_1)^T \mathbf{H}_{t,t-1}^T \mathbf{q}_1 \mathbf{q}_1^T \mathbf{H}_{t,t-1} (\mathbf{I} + \alpha_1 \mathbf{A}_1) (\mathbf{I} + \alpha_1 \mathbf{A}_1)^T \mathbf{H}_{t,t-1}^T \mathbf{q}_1 \mathbf{q}_1^T \mathbf{H}_{t,t-1} (\mathbf{I} + \alpha_1 \mathbf{A}_1) \right] \right) \\ &= \text{Tr} \left(\mathbb{E} \left[(\mathbf{I} + \alpha_1 \mathbf{A}_1) \mathbf{G}_{t-1} (\mathbf{I} + \alpha_1 \mathbf{A}_1) (\mathbf{I} + \alpha_1 \mathbf{A}_1) \mathbf{G}_{t-1} (\mathbf{I} + \alpha_1 \mathbf{A}_1) \right] \right), \end{aligned} \quad (5.35)$$

where $\mathbf{G}_{t-1} = \mathbf{H}_{t,t-1}^T \mathbf{q}_1 \mathbf{q}_1^T \mathbf{H}_{t,t-1} \in \mathbb{R}^{Md \times Md}$. To bound η_t , we first bound the above expression (5.35) for an arbitrary $\mathbf{G}_{t-1} = \mathbf{G}$. We take expectation over only \mathbf{A}_1 and then finally take an expectation over \mathbf{G}_{t-1} . For a fixed arbitrary \mathbf{G} , we have

$$\begin{aligned} &\text{Tr} \left(\mathbb{E} \left[(\mathbf{I} + \alpha_1 \mathbf{A}_1) \mathbf{G} (\mathbf{I} + \alpha_1 \mathbf{A}_1) (\mathbf{I} + \alpha_1 \mathbf{A}_1) \mathbf{G} (\mathbf{I} + \alpha_1 \mathbf{A}_1) \right] \right) \\ &= \text{Tr} \left(\mathbb{E} \left[(\mathbf{G} + \alpha_1 \mathbf{A}_1 \mathbf{G}) (\mathbf{I} + \alpha_1 \mathbf{A}_1) (\mathbf{G} + \alpha_1 \mathbf{A}_1 \mathbf{G}) (\mathbf{I} + \alpha_1 \mathbf{A}_1) \right] \right) \\ &= \text{Tr} \left(\mathbb{E} \left[(\mathbf{G} + \alpha_1 \mathbf{A}_1 \mathbf{G} + \alpha_1 \mathbf{G} \mathbf{A}_1 + \alpha_1^2 \mathbf{A}_1 \mathbf{G} \mathbf{A}_1)^2 \right] \right) \\ &= \text{Tr} \left(\mathbf{G}^2 + \alpha_1 \mathbb{E} \left[\mathbf{G} \mathbf{A}_1 \mathbf{G} \right] + \alpha_1 \left[\mathbf{G}^2 \mathbf{A}_1 \right] + \alpha_1^2 \left[\mathbf{G} \mathbf{A}_1 \mathbf{G} \mathbf{A}_1 \right] + \alpha_1 \left[\mathbf{A}_1 \mathbf{G}^2 \right] + \alpha_1^2 \left[\mathbf{A}_1 \mathbf{G} \mathbf{A}_1 \mathbf{G} \right] + \right. \\ &\quad \alpha_1^2 \left[\mathbf{A}_1 \mathbf{G}^2 \mathbf{A}_1 \right] + \alpha_1^3 \left[\mathbf{A}_1 \mathbf{G} \mathbf{A}_1 \mathbf{G} \mathbf{A}_1 \right] + \alpha_1 \left[\mathbf{G} \mathbf{A}_1 \mathbf{G} \right] + \alpha_1^2 \left[\mathbf{G} \mathbf{A}_1^2 \mathbf{G} \right] + \alpha_1^2 \left[\mathbf{G} \mathbf{A}_1 \mathbf{G} \mathbf{A}_1 \right] + \alpha_1^3 \left[\mathbf{G} \mathbf{A}_1^2 \mathbf{G} \mathbf{A}_1 \right] \\ &\quad \left. \alpha_1^2 \left[\mathbf{A}_1 \mathbf{G} \mathbf{A}_1 \mathbf{G} \right] + \alpha_1^3 \left[\mathbf{A}_1 \mathbf{G} \mathbf{A}_1^2 \mathbf{G} \right] + \alpha_1^3 \left[\mathbf{A}_1 \mathbf{G} \mathbf{A}_1 \mathbf{G} \mathbf{A}_1 \right] + \alpha_1^4 \left[\mathbf{A}_1 \mathbf{G} \mathbf{A}_1^2 \mathbf{G} \mathbf{A}_1 \right] \right) \end{aligned} \quad (5.36)$$

Proceeding similarly as Lemma 21 with $\mathbb{E} \left[\mathbf{A}_1^2 \right] \preceq \frac{v}{M}$ and $\|A_t\| \leq r$, we get

$$\begin{aligned} &\text{Tr} \left(\mathbb{E} \left[(\mathbf{I} + \alpha_1 \mathbf{A}_1) \mathbf{G} (\mathbf{I} + \alpha_1 \mathbf{A}_1) (\mathbf{I} + \alpha_1 \mathbf{A}_1) \mathbf{G} (\mathbf{I} + \alpha_1 \mathbf{A}_1) \right] \right) \\ &\leq (1 + 4\alpha_1 \lambda_1 + 6\alpha_1^2 (\lambda_1^2 + \frac{v}{M}) + 4\alpha_1^3 r (\lambda_1^2 + \frac{v}{M}) + \alpha_1^4 r^2 (\lambda_1^2 + \frac{v}{M})) \text{Tr}(\mathbf{G}^2) \\ &\leq (1 + 4\alpha_1 \lambda_1 + 10\alpha_1^2 \bar{v}) \text{Tr}(\mathbf{G}^2) \leq \exp(4\alpha_1 \lambda_1 + 10\alpha_1^2 \bar{v}) \text{Tr}(\mathbf{G}^2), \end{aligned}$$

where in the last line we used that $\alpha_t \leq \frac{1}{4r}$ and $1 + x \leq \exp(x)$. Now plugging the above in (5.35) and using $\mathbf{G} = \mathbf{G}_{t-1} = \mathbf{H}_{t,t-1}^T \mathbf{q}_1 \mathbf{q}_1^T \mathbf{H}_{t,t-1}$, we have

$$\eta_t = \text{Tr} \left(\mathbb{E} \left[(\mathbf{I} + \alpha_1 \mathbf{A}_1) \mathbf{G}_{t-1} (\mathbf{I} + \alpha_1 \mathbf{A}_1) (\mathbf{I} + \alpha_1 \mathbf{A}_1) \mathbf{G}_{t-1} (\mathbf{I} + \alpha_1 \mathbf{A}_1) \right] \right) \quad (5.37)$$

$$\leq \exp(4\alpha_1 \lambda_1 + 10\alpha_1^2 \bar{v}) \text{Tr}(\mathbf{G}_{t-1}^2) = \exp(4\alpha_1 \lambda_1 + 10\alpha_1^2 \bar{v}) \eta_{t-1} \quad (5.38)$$

Since $\eta_0 = 1$, we have

$$\mathbb{E} \left[(\mathbf{q}_1^T \mathbf{H}_{t,t} \mathbf{H}_{t,t}^T \mathbf{q}_1)^2 \right] \leq \exp \left(\sum_{m \in [t]} 4\alpha_m \lambda_1 + 10\alpha_m^2 \bar{v} \right) \quad (5.39)$$

□

Theorem 10. For any fixed $\delta > 0$ and $\alpha_t = \frac{\eta}{(\lambda_1 - \tilde{\lambda}_2)(\beta + t)}$ for $\eta > 0.5$ and

$$\beta = 20 \max \left(\frac{r\eta}{(\lambda_1 - \tilde{\lambda}_2)}, \frac{(v + \lambda_1^2)\eta^2}{(\lambda_1 - \tilde{\lambda}_2)^2 \log(1 + \frac{\delta}{100})} \right).$$

Then the output of the algorithm at every node \mathbf{x}_t converges to \mathbf{q}_1 as follows:

$$\sin^2(\mathbf{q}_1, \mathbf{x}_t) \leq \frac{C' \log(1/\delta)}{\delta^3} \left(d \left(\frac{\beta}{t} \right)^{2\eta} + \frac{v(\beta + 1)^2 \eta^2}{t\beta^2(2\eta - 1)(\lambda_1 - \tilde{\lambda}_2)^2} \right),$$

with probability at least $1 - \delta$, where C' is an absolute constant.

Proof. As discussed earlier, to prove the convergence we bound the terms $\mathbf{q}_1^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{q}_1$ and $\text{Tr}(\mathbf{Q}_\perp^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{Q}_\perp)$ that help bound the error. First, using Chebyshev's inequality, we get

$$P \left[|\mathbf{q}_1^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{q}_1 - \mathbb{E} [\mathbf{q}_1^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{q}_1]| \geq \frac{1}{\sqrt{\delta}} \sqrt{\text{Var} [\mathbf{q}_1^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{q}_1]} \right] < \delta.$$

So proceesing exactly as in Theorem 9, with probability at least $1 - \delta$, the following holds:

$$\mathbf{q}_1^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{q}_1 \geq \exp \left(2\lambda_1 \sum_{m \in [t]} \alpha_m - 4\lambda_1^2 \sum_{m \in [t]} \alpha_m^2 \right) \left(1 - \frac{1}{\sqrt{\delta}} \sqrt{\exp \left(18\bar{v} \sum_{m \in [t]} \alpha_m^2 \right) - 1} \right)$$

Also, using Markov's inequality, we have

$$P \left[\text{Tr}(\mathbf{Q}_\perp^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{Q}_\perp) \geq \frac{1}{\delta} \mathbb{E} [\text{Tr}(\mathbf{Q}_\perp^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{Q}_\perp)] \right] \leq \delta$$

Thus, with probability at least $1 - \delta$, we have

$$\begin{aligned} \text{Tr}(\mathbf{Q}_\perp^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{Q}_\perp) &\leq \frac{1}{\delta} \mathbb{E} [\text{Tr}(\mathbf{Q}_\perp^T \mathbf{B}_t \mathbf{B}_t^T \mathbf{Q}_\perp)] \\ &\leq \frac{1}{\delta} \exp \left(\sum_{m \in [t]} 2\alpha_m \lambda_2 + \alpha_m^2 \bar{v} \right) \left(d + \frac{v}{M} \sum_{p \in [t]} \alpha_p^2 \exp \left(\sum_{m \in [p]} 2\alpha_m (\lambda_1 - \lambda_2) \right) \right). \end{aligned}$$

Using Lemma 22, we have

$$\begin{aligned} \sin^2(\mathbf{q}_1, \mathbf{x}_t) &\leq \frac{C \log(1/\delta)}{\delta^3} \frac{\exp \left(2\lambda_2 \sum_{m \in [t]} \alpha_m + \bar{v} \sum_{m \in [t]} \alpha_m^2 \right) \left(d + \frac{v}{M} \sum_{m \in [t]} \alpha_m^2 \exp \left(2(\lambda_1 - \lambda_2) \sum_{p \in [m]} \alpha_p \right) \right)}{\exp \left(2\lambda_1 \sum_{m \in [t]} \alpha_m - 4\lambda_1^2 \sum_{m \in [t]} \alpha_m^2 \right) \left(1 - \frac{1}{\sqrt{\delta}} \sqrt{\exp \left(18\bar{v} \sum_{m \in [t]} \alpha_m^2 \right) - 1} \right)} \\ &\leq \frac{C \log(1/\delta)}{\delta^3} \frac{\exp \left(5\bar{v} \sum_{m \in [t]} \alpha_m^2 \right) \left(d \exp \left(2(\lambda_2 - \lambda_1) \sum_{m \in [t]} \alpha_m \right) + \frac{v}{M} \sum_{m \in [t]} \alpha_m^2 \exp \left(2(\lambda_2 - \lambda_1) \sum_{p=m+1}^t \alpha_p \right) \right)}{\left(1 - \frac{1}{\sqrt{\delta}} \sqrt{\exp \left(18\bar{v} \sum_{m \in [t]} \alpha_m^2 \right) - 1} \right)} \end{aligned}$$

Since $\alpha_t = \frac{\eta}{(\lambda_1 - \tilde{\lambda}_2)(\beta + t)}$, using the same bounds as in Theorem 9, we get Thus,

$$\sin^2(\mathbf{q}_1, \mathbf{x}_t) \leq \frac{C' \log(1/\delta)}{\delta^3} \left(d \exp \left(2(\lambda_2 - \lambda_1) \sum_{m \in [t]} \alpha_m \right) + \frac{v}{M} \sum_{m \in [t]} \alpha_m^2 \exp \left(2(\lambda_2 - \lambda_1) \sum_{p=m+1}^t \alpha_p \right) \right) \quad (5.40)$$

Now, since $\sum_{m \in [t]} \alpha_m$ is partial harmonic series, we have

$$\exp\left(2(\lambda_2 - \lambda_1) \sum_{m \in [t]} \alpha_m\right) \leq \left(\frac{\beta}{\beta + t}\right)^{2\eta} \quad (5.41)$$

Also,

$$\exp\left(2(\lambda_2 - \lambda_1) \sum_{p=m+1}^t \alpha_p\right) \leq \left(\frac{\beta + m + 1}{\beta + t + 1}\right)^{2\eta}$$

Thus,

$$\sum_{m \in [t]} \alpha_m^2 \exp\left(2(\lambda_2 - \lambda_1) \sum_{p=m+1}^t \alpha_p\right) \leq \frac{(\beta + 1)^2 \eta^2}{\beta^2 (2\eta - 1) (\lambda_1 - \lambda_2)^2 (\beta + t + 1)}, \quad (5.42)$$

where ζ_1 is by bounding the sum using the corresponding integral. Plugging (5.41) and (5.42) in (5.40), we get

$$\begin{aligned} \sin^2(\mathbf{q}_1, \mathbf{x}_t) &\leq \frac{C' \log(1/\delta)}{\delta^3} \left(d \left(\frac{\beta}{\beta + t}\right)^{2\eta} + \frac{v(\beta + 1)^2 \eta^2}{M \beta^2 (2\eta - 1) (\lambda_1 - \lambda_2)^2 (\beta + t + 1)} \right) \\ &\leq \frac{C' \log(1/\delta)}{\delta^3} \left(d \left(\frac{\beta}{t}\right)^{2\eta} + \frac{v(\beta + 1)^2 \eta^2}{M t \beta^2 (2\eta - 1) (\lambda_1 - \lambda_2)^2} \right) \end{aligned}$$

This shows that in a federated learning setup, when exact averaging is possible (as compared to inexact averaging in DIEGO) there is an increase in rate as $\mathcal{O}(1/Mt)$. Also, a notable result here is that this improvement in rate is possible as long as $M = \mathcal{O}(t^{2\eta-1})$. In the next section, we present extensive numerical results that further support the claims. \square

5.6 Numerical Results

In this section, we demonstrate the efficiency of the proposed solution through extensive experiments on synthetic as well as real world data. The whole idea of collaboration in the network is motivated from the fact since a total of M samples will be processed in a network of M nodes (assuming one sample is processed at each node per time instant t), the exchange of information between the connected nodes should in turn be equivalent to processing more than one sample at each node per time instant. Since the convergence rate of distributed PCA depends on the number of samples processed, this distributed learning would improve the rate of convergence. We compare our results for centralized PCA solution for eigenvector estimation, generalized Hebbian estimation (GHA), our proposed algorithm (DIEGO) that does inexact averaging and the exact averaging case of federated learning. The step size used for all experiments is $\alpha_t = \frac{\alpha}{t}$. The y-axis of all the plots denote the average angle between the estimated eigenvectors $\mathbf{x}_{i,k,t}$

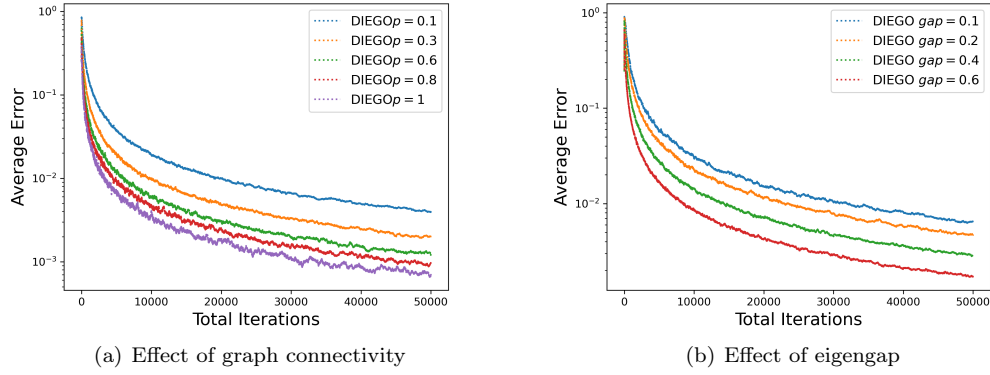


Figure 5.1: Effect of different parameters on the performance of DIEGO for $K = 1$

and the true eigenvectors $\pm \mathbf{q}_k$ of the population covariance matrix $\mathbf{\Sigma}$ across all the M nodes in the network and is given by

$$\mathcal{E} = \frac{1}{MK} \sum_{i=1}^M \sum_{k=1}^K \left(1 - \left(\frac{\mathbf{x}_{i,k,t}^T \mathbf{q}_k}{\|\mathbf{x}_{i,k,t}\|} \right)^2 \right). \quad (5.43)$$

5.6.1 Synthetic Data

There are certain parameters like eigengap, graph connectivity, number of samples etc. that decide the performance of our proposed algorithm for distributed PCA in streaming setting. We first study the effect of these parameters on the performance of our algorithm. In the following experiments we generate samples from multivariate Gaussian distribution and for each experiment we perform 50 Monte-Carlo trials.

Effect of Graph Connectivity

We simulate an Erdos-Renyi graph of $M = 20$ nodes with different connectivity factor $p \in \{0.1, 0.3, 0.6, 0.8, 1\}$. We generate synthetic data samples of dimension $d = 20$ from a multivariate Gaussian distribution with zero mean and fixed covariance matrix $\mathbf{\Sigma}$ such that $\lambda_1 - \lambda_2 = 0.2$. Figure 5.1(a) shows that stronger the connectivity of the graph, faster is the convergence. Also, the case of $p = 1$ is when the graph is strongly connected. In such a case, there will be an exact averaging at every node in every iteration. Therefore, the performance when $p = 1$ is same as the performance for the case of federated learning.

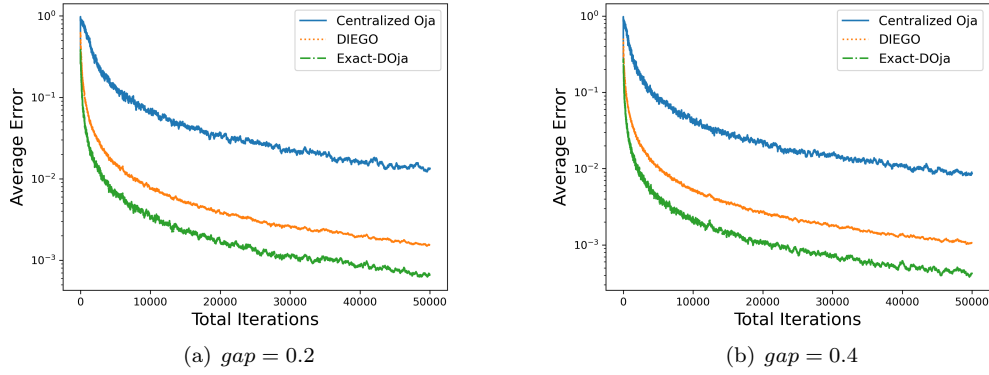


Figure 5.2: Performance comparison of DIEGO with centralized Oja and the Distributed Oja in federated learning setting in a network of $M = 20$ nodes.

Effect of Eigengap

We simulate an Erdos-Renyi graph of $M = 20$ nodes with a fixed connectivity factor p and different eigengaps $\lambda_1 - \lambda_2 = \{0.1, 0.2, 0.4, 0.6\}$. Figure 5.1(b) shows that as eigengap increases, so does the convergence which is in line with the performance of PCA algorithms.

Comparison with other methods

Next, we show the performance comparison of our proposed method DIEGO with centralized solution as well as generalized distributed Oja in the federated learning setup (exact averaging case). We simulate an Erdos-Renyi graph of $M \in \{20, 100\}$ nodes with a fixed connectivity factor p and different two eigengaps in each case. First, we show the performance comparison when estimating only the dominant eigenvector. Figure 5.2 shows the comparison for the case of $M = 20$ nodes. Figure 5.3 shows similar comparison for $M = 100$ nodes. The gap between the performance of centralized Oja and distributed methods increase with the increase in the number of nodes, since effectively more samples are processed per node in a larger network.

Next, we show the performance comparison when multiple top eigenvectors are being estimated. As Figure 5.4 clearly shows that even for estimation of multiple eigenvectors, the algorithms show similar performance.

5.6.2 Real World Data

We also provide some results for the real-world dataset of MNIST [71] and Higgs [79]. For MNIST dataset, we simulate the distributed setup with an Erdos-Renyi graph with $p = 0.5$ and $M = 10$ nodes. MNIST dataset has $N = 60,000$ samples with each sample having a dimension

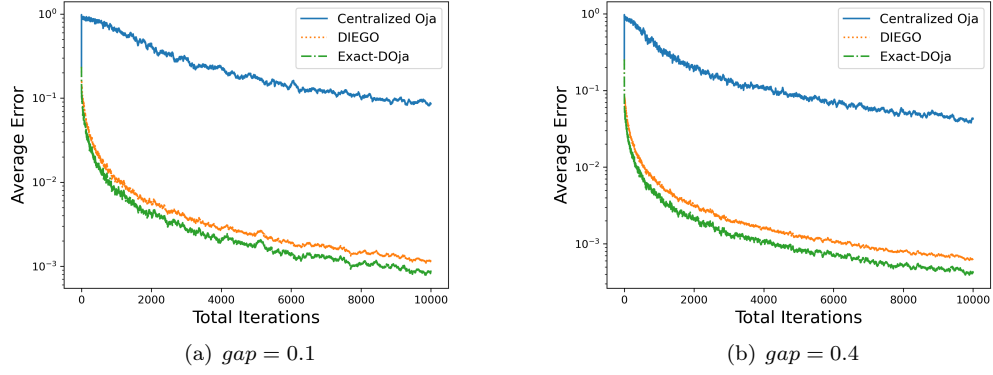


Figure 5.3: Performance comparison of DIEGO with centralized Oja and the Distributed Oja in federated learning setting in a network of $M = 100$ nodes.

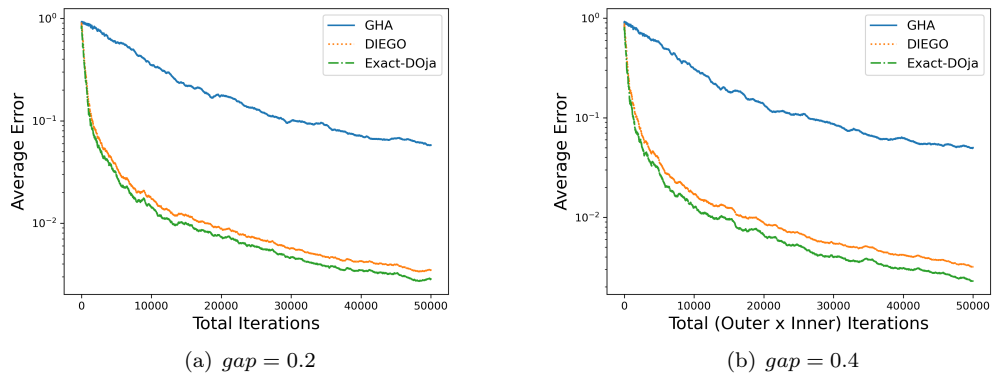


Figure 5.4: Performance comparison of DIEGO with centralized Oja and the Distributed Oja in federated learning setting while estimating $K = 5$ eigenvectors.

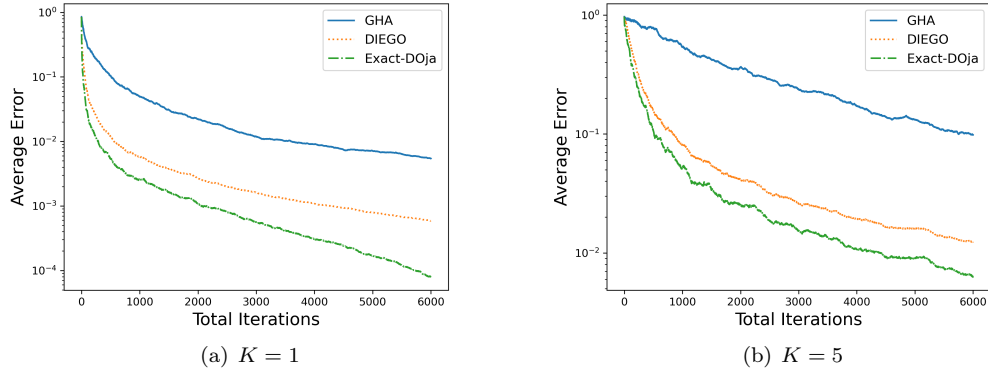


Figure 5.5: Performance comparison of DIEGO with centralized Oja and the Distributed Oja in federated learning setting while estimating $K = 5$ eigenvectors for MNIST dataset.

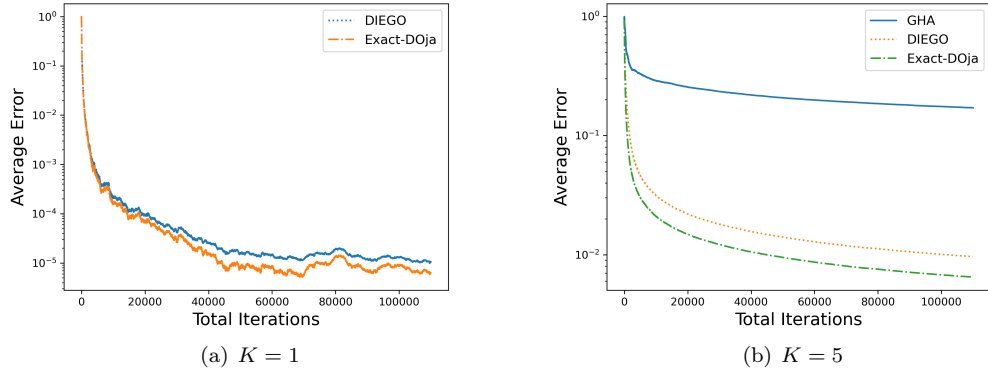


Figure 5.6: Performance comparison of DIEGO with centralized Oja and the Distributed Oja in federated learning setting while estimating $K = 5$ eigenvectors for Higgs dataset.

of $d = 784$. Figure 5.5 shows the comparison of the centralized and distributed PCA algorithms for MNIST dataset when $K \in \{1, 5\}$ dominant eigenvectors are estimated. The results shown are averaged over 50 trials for different random initializations and random data shuffling. The initial step size α used for the two cases were $\alpha = 0.5$ and $\alpha = 10$ respectively. For Higgs dataset, we simulate the distributed setup with an Erdos-Renyi graph with $p = 0.5$ and $M = 100$ nodes. Higgs dataset has $N = 1.1 \times 10^7$ samples with each sample having a dimension of $d = 28$. Figure 5.6 shows the comparison of the centralized and distributed PCA algorithms for MNIST dataset when $K \in \{1, 5\}$ dominant eigenvectors are estimated. The results shown are averaged over 50 trials for different random initializations and random data shuffling. The initial step size α used for the two cases is $\alpha = 2$.

Bibliography

- [1] S. B. Kotsiantis, I. Zaharakis, P. Pintelas *et al.*, “Supervised machine learning: A review of classification techniques,” *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.
- [2] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, “An introductory review of deep learning for prediction models with big data,” *Frontiers in Artificial Intelligence*, vol. 3, p. 4, 2020.
- [3] H. Hotelling, “Analysis of a complex of statistical variables into principal components,” *J. Educational Psychology*, vol. 24, no. 6, pp. 417–441, 1933.
- [4] R. A. FISHER, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
- [5] A. Fischer and C. Igel, “An introduction to restricted boltzmann machines,” in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, L. Alvarez, M. Mejail, L. Gomez, and J. Jacobo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 14–36.
- [6] P. Baldi and K. Hornik, “Neural networks and principal component analysis: Learning from examples without local minima,” *Neural Netw.*, vol. 2, no. 1, p. 53–58, Jan. 1989.
- [7] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: a review and new perspectives,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, p. 1798–1828, August 2013. [Online]. Available: <https://doi.org/10.1109/TPAMI.2013.50>
- [8] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob, “Big IoT data analytics: Architecture, opportunities, and open research challenges,” *IEEE Access*, vol. 5, pp. 5247–5261, 2017.
- [9] M. M. Rathore, A. Paul, W.-H. Hong, H. Seo, I. Awan, and S. Saeed, “Exploiting iot and big data analytics: Defining smart digital city using real-time urban data,” *Sustainable Cities and Society*, vol. 40, pp. 600–610, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2210670717309782>

- [10] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 609–616. [Online]. Available: <https://doi.org/10.1145/1553374.1553453>
- [11] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, “A neural probabilistic language model,” *J. Mach. Learn. Res.*, vol. 3, no. null, p. 1137–1155, mar 2003.
- [12] E. Oja and J. Karhunen, “On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix,” *J. Math. Anal. Applicat.*, vol. 106, no. 1, pp. 69 – 84, 1985.
- [13] T. D. Sanger, “Optimal unsupervised learning in a single-layer linear feedforward neural network,” *Neural Netw.*, vol. 2, no. 6, pp. 459 – 473, 1989.
- [14] D. O. Hebb, *The Organization of Behavior : A Neuropsychological Theory*. Wiley New York, 1949.
- [15] A. Gang, H. Raja, and W. U. Bajwa, “Fast and communication-efficient distributed PCA,” in *Proc. 2019 IEEE International Conf. Acoustics, Speech and Signal Process. (ICASSP)*, 2019, pp. 7450–7454.
- [16] A. Gang and W. U. Bajwa, “A linearly convergent algorithm for distributed principal component analysis,” *Signal Processing*, vol. 193, p. 108408, 2022.
- [17] —, “FAST-PCA: A fast and exact algorithm for distributed principal component analysis,” *arXiv preprint arXiv:2108.12373*, 2021.
- [18] K. Pearson, “On lines and planes of closest fit to systems of points in space,” *Philosophical Mag.*, vol. 2, pp. 559–572, 1901.
- [19] G. H. Golub and C. F. Van Loan, *Matrix Computations (3rd Ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.
- [20] C. Lanczos, “An iteration method for the solution of the eigenvalue problem of linear differential and integral operators,” *J. Research Nat. Bureau Standards*, 1950.
- [21] T. P. Krasulina, “Method of stochastic approximation in the determination of the largest eigenvalue of the mathematical expectation of random matrices,” *Autom. Remote Control*, vol. 1970, pp. 215–221, 1970.
- [22] H. Robbins and S. Monro, “A Stochastic Approximation Method,” *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400 – 407, 1951. [Online]. Available: <https://doi.org/10.1214/aoms/1177729586>

- [23] Z. Yi, M. Ye, J. C. Lv, and K. K. Tan, "Convergence analysis of a deterministic discrete time system of Oja's PCA learning algorithm," *IEEE Trans. Neural Netw.*, vol. 16, no. 6, pp. 1318–1328, Nov 2005.
- [24] J. C. Lv, Z. Yi, and K. K. Tan, "Global convergence of GHA learning algorithm with nonzero-approaching adaptive learning rates," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1557–1571, 2007.
- [25] C. Tang, "Exponentially convergent stochastic k-PCA without variance reduction," in *NeurIPS*, 2019.
- [26] O. Shamir, "A stochastic PCA and SVD algorithm with an exponential convergence rate," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, p. 144–152.
- [27] P. Xu, B. He, C. De Sa, I. Mitliagkas, and C. Re, "Accelerated stochastic power iteration," in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Storkey and F. Perez-Cruz, Eds., vol. 84. PMLR, 09–11 Apr 2018, pp. 58–67.
- [28] A. Balsubramani, S. Dasgupta, and Y. Freund, "The fast convergence of incremental PCA," in *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013.
- [29] I. Mitliagkas, C. Caramanis, and P. Jain, "Memory limited, streaming pca," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013. [Online]. Available: <https://proceedings.neurips.cc/paper/2013/file/76cf99d3614e23eabab16fb27e944bf9-Paper.pdf>
- [30] M. Hardt and E. Price, "The noisy power method: A meta algorithm with applications," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, p. 2861–2869.
- [31] P. Jain, C. Jin, S. M. Kakade, P. Netrapalli, and A. Sidford, "Streaming pca: Matching matrix bernstein and near-optimal finite sample guarantees for oja's algorithm," in *29th Annual Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, V. Feldman, A. Rakhlin, and O. Shamir, Eds., vol. 49. Columbia University, New York, New York, USA: PMLR, 23–26 Jun 2016, pp. 1147–1164.
- [32] J. Tropp, "User-friendly tail bounds for sums of random matrices," *Foundations of Computational Mathematics - FoCM*, vol. 12, 04 2010.

- [33] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," in *Compressed Sensing*, 2012.
- [34] P. Yang, C.-J. Hsieh, and J. ling Wang, "History PCA: A new algorithm for streaming PCA," *arXiv: Machine Learning*, 2018.
- [35] Z. Allen-Zhu and Y. Li, "First efficient convergence for streaming k-PCA: A global, gap-free, and near-optimal rate," in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, 2017, pp. 487–492.
- [36] R. Arora, A. Cotter, and N. Srebro, "Stochastic optimization of PCA with capped MSG," in *Advances Neural Inform. Process. Sys.*, 2013, pp. 1815–1823.
- [37] D. Zhang and L. Balzano, "Global convergence of a grassmannian gradient descent algorithm for subspace estimation," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Gretton and C. C. Robert, Eds., vol. 51. Cadiz, Spain: PMLR, 09–11 May 2016, pp. 1460–1468.
- [38] C. D. Sa, C. Re, and K. Olukotun, "Global convergence of stochastic gradient descent for some non-convex matrix problems," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 2332–2341.
- [39] M.-F. Balcan, V. Kanchanapally, Y. Liang, and D. Woodruff, "Improved distributed principal component analysis," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, p. 3113–3121.
- [40] J. Fan, D. Wang, K. Wang, and Z. Zhu, "Distributed Estimation of Principal Eigenspaces," *ArXiv e-prints*, Feb. 2017.
- [41] C. Boutsidis, D. P. Woodruff, and P. Zhong, "Optimal principal component analysis in distributed and streaming models." New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: <https://doi.org/10.1145/2897518.2897646>
- [42] D. Garber, O. Shamir, and N. Srebro, "Communication-efficient algorithms for distributed stochastic principal component analysis," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, p. 1203–1212.
- [43] S. X. Wu, H.-T. Wai, L. Li, and A. Scaglione, "A review of distributed algorithms for principal component analysis," *Proc. IEEE*, vol. 106, no. 8, pp. 1321–1340, 2018.
- [44] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," *J. Comput. and Syst. Sci.*, vol. 74, no. 1, pp. 70 – 83, 2008.

- [45] A. Scaglione, R. Pagliari, and H. Krim, "The decentralized estimation of the sample covariance," in *Proc. 2008 42nd Asilomar Conf. on Signals, Syst. and Comput.*, 2008, pp. 1722–1726.
- [46] L. Li, A. Scaglione, and J. H. Manton, "Distributed principal subspace estimation in wireless sensor networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 725–738, Aug 2011.
- [47] A. Gang, B. Xiang, and W. U. Bajwa, "Distributed principal subspace analysis for partitioned big data: Algorithms, analysis, and implementation," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 7, pp. 699–715, 2021.
- [48] H. Straková, W. N. Gansterer, and T. Zemen, "Distributed QR factorization based on randomized algorithms," in *Proc. Int. Conf. Parallel Process. and Appl. Math.* Springer, 2011, pp. 235–244.
- [49] H. Raja and W. U. Bajwa, "Cloud K-SVD: Computing data-adaptive representations in the cloud," in *Proc. 2013 51st Annual Allerton Conf. Commun., Control and Computing (Allerton)*, 2013, pp. 1474–1481.
- [50] H. Raja and W. U. Bajwa, "Cloud-K-SVD: A collaborative dictionary learning algorithm for big, distributed data," *IEEE Trans. Signal Process.*, vol. 64, no. 1, pp. 173–188, Jan 2016.
- [51] H. Wai, A. Scaglione, J. Lafond, and E. Moulines, "Fast and privacy preserving distributed low-rank regression," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process., (ICASSP)*, 2017, pp. 4451–4455.
- [52] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.
- [53] H. Ye and T. Zhang, "DeEPCA: Decentralized exact PCA with linear convergence rate," *CoRR*, vol. abs/2102.03990, 2021. [Online]. Available: <https://arxiv.org/abs/2102.03990>
- [54] H. Raja and W. U. Z. Bajwa, "Distributed stochastic algorithms for high-rate streaming principal component analysis," *ArXiv*, vol. abs/2001.01017, 2020.
- [55] M. Hong, D. Hajinezhad, and M.-M. Zhao, "Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks," in *Proc. 34th Int. Conf. Mach. Learning*, vol. 70. PMLR, 06–11 Aug 2017, pp. 1529–1538.
- [56] P. Bianchi and J. Jakubowicz, "Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization," *IEEE Trans. Autom. Control*, vol. 58, no. 2, pp. 391–405, 2013.
- [57] H. Wai, A. Scaglione, J. Lafond, and E. Moulines, "A projection-free decentralized algorithm for non-convex optimization," in *Proc. 2016 IEEE Global Conf. Signal and Inform. Process. (GlobalSIP)*, 2016, pp. 475–479.

- [58] S. Chen, A. Garcia, M. Hong, and S. Shahrampour, “Decentralized riemannian gradient descent on the stiefel manifold,” *arXiv preprint arXiv:2102.07091*, 2021.
- [59] F. L. Andrade, M. A. Figueiredo, and J. Xavier, “Distributed picard iteration,” *arXiv preprint arXiv:2104.00131*, 2021.
- [60] —, “Distributed picard iteration: Application to distributed em and distributed pca,” *arXiv preprint arXiv:2106.10665*, 2021.
- [61] M. Nokleby, H. Raja, and W. U. Bajwa, “Scaling-up distributed processing of data streams for machine learning,” *Proc. IEEE*, vol. 108, no. 11, pp. 1984–2012, Nov. 2020.
- [62] W. U. Bajwa, V. Cevher, D. Papailiopoulos, and A. Scaglione, “Machine learning from distributed, streaming data,” *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 11–13, May 2020.
- [63] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan 2009.
- [64] M. K. Warmuth and D. Kuzmin, “Randomized PCA algorithms with regret bounds that are logarithmic in the dimension,” in *Proc. Advances Neural Inform. Process. Syst.*, 2007, pp. 1481–1488.
- [65] W. Shi, Q. Ling, G. Wu, and W. Yin, “EXTRA: an exact first-order algorithm for decentralized consensus optimization,” *SIAM J. Optim.*, vol. 25, no. 2, pp. 944–966, 2015.
- [66] F. S. Cattivelli and A. H. Sayed, “Diffusion LMS strategies for distributed estimation,” *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, 2010.
- [67] S. Kar and J. M. Moura, “Consensus + innovations distributed inference over networks: cooperation and sensing in networked systems,” *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 99–109, 2013.
- [68] K. Yuan, Q. Ling, and W. Yin, “On the convergence of decentralized gradient descent,” *SIAM J. Optim.*, vol. 26, no. 3, pp. 1835–1854, 2016.
- [69] S. Boyd, P. Diaconis, and L. Xiao, “Fastest mixing markov chain on a graph,” *SIAM REVIEW*, vol. 46, pp. 667–689, 2003.
- [70] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. USA: Princeton University Press, 2007.
- [71] Y. LeCun, C. Cortes, and C. Burges, “MNIST handwritten digit database,” *ATT Labs*, vol. 2, 2010.
- [72] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Tech. Rep., 2009.

- [73] P. D. Lorenzo and G. Scutari, “NEXT: In-network nonconvex optimization,” *IEEE Trans. Signal Inform. Process. Netw.*, vol. 2, no. 2, pp. 120–136, 2016.
- [74] G. Qu and N. Li, “Harnessing smoothness to accelerate distributed optimization,” *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, 2018.
- [75] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: <http://tensorflow.org/>
- [76] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.
- [77] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms,” *CoRR*, vol. abs/1708.07747, 2017. [Online]. Available: <http://arxiv.org/abs/1708.07747>
- [78] M. K. Warmuth and D. Kuzmin, “Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension,” *Journal of Machine Learning Research*, vol. 9, no. 75, pp. 2287–2320, 2008. [Online]. Available: <http://jmlr.org/papers/v9/warmuth08a.html>
- [79] P. Baldi, P. Sadowski, and D. Whiteson, “Searching for Exotic Particles in High-Energy Physics with Deep Learning,” *Nature Commun.*, vol. 5, p. 4308, 2014.