Incremental Aggregated Riemannian Gradient Method for Distributed PCA

Xiaolu Wang* **Yuchen Jiao†** **Hoi-To Wai*** **Yuantao Gu†**
*Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong
†Department of Electronic Engineering, Tsinghua University

Abstract

We consider the problem of distributed principal component analysis (PCA) where the data samples are dispersed across different agents. Despite the rich literature on this problem under various specific settings, there is still a lack of efficient algorithms that are amenable to decentralized and asynchronous implementations. In this paper, we extend the incremental aggregated gradient (IAG) method in convex optimization to the nonconvex PCA problems based on an Riemannian gradient-type method named IARG-PCA. The IARG-PCA method admits low per-iteration computational and communication cost and can be readily implemented in a decentralized and asynchronous manner. Moreover, we show that the IARG-PCA method converges linearly to the leading eigenvector of the sample covariance of the whole dataset with a constant step size. The iteration complexity coincides with the best-known result of the IAG method in terms of the linear dependence on the number of agents. Meanwhile, the communication complexity is much lower than the state-of-the-art decentralized PCA algorithms if the eigengap of the sample covariance is moderate. Numerical experiments on synthetic and real datasets show that our IARG-PCA method exhibits substantially lower communication cost and comparable computational cost compared with other existing algorithms.

1 INTRODUCTION

Principal component analysis (PCA) is one of the most fundamental and long-standing problems in data analysis (Hotelling, 1933), which aims to identify a direction of a line that preserves the maximal variance of the dataset. Specifically, suppose that there are n data samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$, which are assumed to be mean-centered without loss of generality, i.e., $\frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i = \boldsymbol{0}$. The PCA can be formulated as the following nonconvex optimization problem:

$$\min_{\boldsymbol{w}\in\mathbb{S}^{d-1}} \left\{\mathcal{F}(\boldsymbol{w}) \coloneqq -\boldsymbol{w}^\top \boldsymbol{A}\boldsymbol{w}\right\}, \tag{1}$$

where $\mathbb{S}^{d-1} \coloneqq \{\boldsymbol{w} \in \mathbb{R}^d : \|\boldsymbol{w}\|_2 = 1\}$ is the unit sphere and $\boldsymbol{A} \coloneqq (1/n)\sum_{i=1}^n \boldsymbol{x}_i\boldsymbol{x}_i^\top$ is the sample covariance matrix. Due to the constant prevalence and significance of this problem in computer vision (Ma and Yuan, 2019), data clustering (Liu and Tan, 2019), neural science (Cunningham and Yu, 2014), genomics (Dorrity et al., 2020), large-scale climate modeling (Gittens et al., 2016), etc., there has been a large body of literature that tackles it under various specific settings (see Section 1.2 for detailed discussion).

In this work, we are interested in the setting where the data samples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ are dispersed across different agents. Specifically, suppose that there are N agents $\{1, \ldots, N\}$ and the agent i stores n_i local data samples represented by matrix $\boldsymbol{X}_i \coloneqq \{\boldsymbol{x}_i^1, \ldots, \boldsymbol{x}_i^{n_i}\} \in \mathbb{R}^{d\times n_i}$. Let $n \coloneqq \sum_{i=1}^N n_i$, we consider Problem (1) with $\boldsymbol{A} = (1/n)\sum_{i=1}^N \boldsymbol{X}_i\boldsymbol{X}_i^\top$. This setting is motivated by a wide range of real-world scenarios, where the data are acquired by multiple nodes, e.g., CPU cores, computing clusters, wireless sensors, and wearable devices, that are interconnected by networks (Assran et al., 2020). In practice, transmitting large volumes of data and process the whole dataset using a single agent can be exceedingly time/energy-consuming. This necessitates the development of efficient distributed algorithms for solving Problem (1). In large-scale distributed PCA, common algorithms require synchronization of the entire network after every round of communication of all agents, which will cause significant delay of the distributed systems in the presence of straggling nodes (Li et al., 2021).

1.1 Our Contributions

To address the aforementioned concerns, we propose the incremental aggregated Riemannian gradient method for

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

distributed PCA, which is referred to as IARG-PCA. We summarize the main contributions as follows:

- Our IARG-PCA method captures the manifold-constrained structure in Problem (1) by aggregating the outdated Riemannian gradient information stored by different agents in an incremental fashion. This extends the incremental aggregated gradient (IAG)-type methods (Blatt et al., 2007; Gurbuzbalaban et al., 2017; Vanli et al., 2018; Wai et al., 2018, 2020) for unconstrained convex optimization problems. On the computational side, the IARG-PCA method visits only one agent in each iteration and thus admits low per-iteration computational cost. On the communication side, the IARG-PCA method visits the agents according to a Hamiltonian walk[1] on the network and thus uses only one link in the network to transmit $\mathcal{O}(d)$ amount of data after each update. Since there is not a master agent that dominates all other agents and no synchronization is required in each round of communication, the IARG-PCA method is intrinsically decentralized and asynchronous and thus greatly alleviates the straggler's effect.
- The IARG-PCA method departs from the popular variance reduction approach for PCA such as (Garber et al., 2016; Shamir, 2015). Instead, it is built upon the IAG technique under a novel context of concave minimization with a manifold (non-convex) constraint. Note that existing convergence analyses of the IAG method for unconstrained strongly convex problems are no longer applicable to IARG-PCA. Specifically, we treat the IARG-PCA method as an inexact version of the Riemannian gradient descent (RGD) method with simultaneous multiplicative and additive perturbations. By carefully bounding the error terms that is shown to decrease to 0, we establish the linear convergence rate of IARG-PCA with $\mathcal{O}\left(\frac{N}{\Delta^2}\log\left(\frac{1}{\epsilon}\right)\right)$ iteration complexity to obtain an ϵ-suboptimal solution to Problem (1), where Δ is the eigengap (i.e., the difference between the two largest eigenvalues) of $\boldsymbol{A}$. This complexity matches the best-known convergence rate of the IAG method in terms of the linear dependence on N and is comparable to the variance-reduced methods such as (Garber et al., 2016; Shamir, 2015). Since only $\mathcal{O}(d)$ data should be transmitted *across the network* (as only one edge is used) after each iteration, the communication complexity of IARG-PCA is also $\mathcal{O}\left(\frac{N}{\Delta^2}\log\left(\frac{1}{\epsilon}\right)\right)$. The communication complexity of IARG-PCA is substantially better than other state-of-the-art decentralized algorithms for PCA.

1.2 Related Works

In this subsection, we review several closely related lines of research on PCA and discuss their connections with our proposed approach.

[1] A Hamiltonian walk on a connected network is a closed walk of minimal length which visits every node of a network (and may visit each node and link more than once).

Batch Algorithms for PCA: Since the optimal solution to Problem (1) is the eigenvector (up to a sign) associated with the largest eigenvalue of $\boldsymbol{A}$, one can invoke common numerical algebra algorithms, e.g., the power method and the Lanczos method (Golub and Van Loan, 2013), to solve it. Recently, there has been particular interests to solve Problem (1) using the RGD method (Absil et al., 2009). The convergence rate of RGD when solving Problem (1) is shown to be $\mathcal{O}\left(\frac{1}{\Delta^2}\log\left(\frac{1}{\epsilon}\right)\right)$ by Xu et al. (2018b) and later improved to be $\mathcal{O}\left(\frac{1}{\Delta}\log\left(\frac{1}{\epsilon}\right)\right)$ by Ding et al. (2020); Xu and Li (2021), which coincides with the rate of the power method. A major deficiency of batch algorithms is that they require full data pass in every iteration, which can result in slow initial improvement for large datasets (Bottou et al., 2018). By contrast, our IARG-PCA can be viewed as an approximate version of the RGD method, which requires considerably lower per-iteration cost by sacrificing the exactness of the first-order information.

Incremental Algorithms for PCA: The original incremental algorithms (including stochastic algorithms) for PCA can be traced back to the seminal Krasulina's method (Krasulina, 1969) and the Oja's method (Oja, 1982). Both methods take data points in an incremental fashion and are shown to converge at a sublinear rate of $\mathcal{O}\left(\frac{1}{\Delta^2\epsilon}\right)$ with diminishing step sizes in the online setting (Balsubramani et al., 2013; Jain et al., 2016). Analogous to our IARG-PCA method, the Krasulina's and Oja's methods can also be implemented in a distributed manner by taking the data samples according to a Hamiltonian walk on the network. Since only one link is used at a time for data transmission, with the communication complexity being $\mathcal{O}\left(\frac{1}{\Delta^2\epsilon}\right)$. If the data points are randomly sampled in each step, the Krasulina's and Oja's methods will exhibit high-variance stochastic gradients. As a remedy, there are a collection of works that employ the variance reduction technique in convex optimization (Johnson and Zhang, 2013) to solve Problem (1), including the VR-PCA (Shamir, 2015), shift-and-invert preconditioning-based power method (SIP-PM) (Garber et al., 2016), VR Power+M (Xu et al., 2018a), and VR Power (Kim and Klabjan, 2020). These variance-reduced algorithms require similar $\mathcal{O}\left(\left(N+\frac{1}{\Delta^2}\right)\text{polylog}\left(\frac{1}{\epsilon}\right)\right)$ iteration complexity. However, the stochastic PCA algorithms are not amenable to distributed implementations. Table 1 compares these methods against IARG-PCA and a more detailed discussion will be provided in Section 2.1.

Distributed Algorithms for PCA: Although there have been a host of works on distributed PCA, most recent works are based on the centralized master-slave framework (Alimisis et al., 2021; Grammenos et al., 2020; Huang and Pan, 2020; Li et al., 2021; Wu et al., 2018). The most related decentralized counterparts of our IARG-PCA include DePM (Wai et al., 2017), S-DOT (Gang et al., 2021), DRGTA (Chen et al., 2021), and DeEPCA

Table 1: Total runtime of state-of-the-art incremental algorithms for the PCA problem (1). Note that Δ is the eigengap (i.e., difference between the two largest eigenvalues) of $\boldsymbol{A}$.

	Distributed?	Total Runtime	Convergence
VR-PCA (Shamir, 2015)	✗	$\mathcal{O}\left(d\left(N+\frac{1}{\Delta^2}\right)\log\left(\frac{1}{\epsilon}\right)\right)$	Local, in Expectation
SIP-PM (Garber et al., 2016)	✗	$\mathcal{O}\left(d\left(N+\frac{1}{\Delta^2}\right)\text{polylog}\left(\frac{1}{\epsilon}\right)\right)$	Global, in Expectation
VR Power+M (Xu et al., 2018a)	✗	$\mathcal{O}\left(d\left(N+\frac{\sqrt{d}}{\Delta^2}\right)\log\left(\frac{1}{\epsilon}\right)\right)$	Local, in Expectation
VR Power (Kim and Klabjan, 2020)	✗	$\mathcal{O}\left(d\left(N+\frac{1}{\Delta^2}\right)\log\left(\frac{1}{\epsilon}\right)\right)$	Global, in Expectation
Krasulina/Oja (Balsubramani et al., 2013)	✓	$\mathcal{O}\left(\frac{d}{\Delta^2\epsilon}\right)$	Local, in Expectation & w.h.p.
IARG-PCA	✓	$\mathcal{O}\left(\frac{dN}{\Delta^2}\log\left(\frac{1}{\epsilon}\right)\right)$	Local, Deterministic

Table 2: Communication complexity of decentralized algorithms for PCA under common types of networks. The complexity of Krasulina's and Oja's methods is established in the online setting, where σ^2 is the variance of the gradient estimators. Note that under mild assumptions σ^2 is proportional to N in the offline setting. The complexity of DeEPCA is obtained by the second largest eigenvalues of the normalized Laplacian matrices of the networks (see Spielman (2015)).

	Asynchronous?	Ring	Erdős–Rényi	Complete	Dumbbell
DeEPCA	✗	$\mathcal{O}\left(\frac{N^2}{\Delta}\log\left(\frac{1}{\epsilon}\right)\right)$	$\mathcal{O}\left(\frac{N\log^2(N)}{\Delta}\log\left(\frac{1}{\epsilon}\right)\right)$	$\mathcal{O}\left(\frac{N^2}{\Delta}\log\left(\frac{1}{\epsilon}\right)\right)$	$\mathcal{O}\left(\frac{N^3}{\Delta}\log\left(\frac{1}{\epsilon}\right)\right)$
Krasulina's and Oja's	✓	$\mathcal{O}\left(\frac{\sigma^2}{\Delta^2\epsilon}\right)$			
IARG-PCA	✓	$\mathcal{O}\left(\frac{N}{\Delta^2}\log\left(\frac{1}{\epsilon}\right)\right)$			

(Ye and Zhang, 2021), which are all based on the consensus mechanism and thus the global synchronization is implicitly needed. Among them, DRGTA is applicable to the PCA problem but initially designed for the general manifold optimization problems, thus only a sublinear convergence rate is given. DeEPCA achieves the best-known $\mathcal{O}\left(\frac{|E|}{\sqrt{1-\lambda_2(\boldsymbol{W})}}\frac{1}{\Delta}\log\left(\frac{1}{\epsilon}\right)\right)$ communication complexity for decentralized PCA, where $|E|$ denotes the number of links in the network and $\lambda_2(\boldsymbol{W})$ is the second largest eigenvalue of the weight matrix $\boldsymbol{W}$ associated with the network. Since $|E| \geq N-1$ for connect networks and $\sqrt{1-\lambda_2(\boldsymbol{W})} \leq 1$, the communication complexity of DeEPCA cannot be better than that of IARG-PCA in terms of the dependence on N. We summarize the communication complexity of DeEPCA and IARG-PCA under different common types of networks in Table 2. As we can see, IARG-PCA not only is amenable to asynchronous implementation that is independent of the network topology, but also can be substantially more communication-efficient than DeEPCA if the eigengap Δ is not significantly small.

2 ALGORITHM DEVELOPMENT

In this section, we first develop the IARG-PCA method in the sequential update setting and then discuss its decentralized implementation.

Suppose that all the data blocks $\boldsymbol{X}_1,\ldots,\boldsymbol{X}_N$ are assembled in a single agent. Since Problem (1) possesses a unit-sphere manifold constraint, one may apply the following RGD method (Absil et al., 2009; Ding et al., 2020; Xu and Li, 2021) to solve it:

$$\boldsymbol{w}^{t+1} = \frac{\boldsymbol{w}^t - \eta\,\text{grad}\,\mathcal{F}(\boldsymbol{w}^t)}{\|\boldsymbol{w}^t - \eta\,\text{grad}\,\mathcal{F}(\boldsymbol{w}^t)\|_2} \text{ for } t\in\mathbb{N}, \qquad (2)$$

where the Riemannian gradient of $\mathcal{F}$ at $\boldsymbol{w}^t\in\mathbb{S}^{d-1}$ is

$$\text{grad}\,\mathcal{F}(\boldsymbol{w}^t) := -\frac{1}{n}\sum_{i=1}^{N}\left(\boldsymbol{I}-\boldsymbol{w}^t(\boldsymbol{w}^t)^\top\right)\boldsymbol{X}_i\boldsymbol{X}_i^\top\boldsymbol{w}^t$$

However, computing the Riemannian gradient $\text{grad}\,\mathcal{F}(\boldsymbol{w}^t)$ involves the whole dataset, which makes the RGD method to suffer from $\mathcal{O}(dn)$ per-iteration computational cost. Furthermore, the algorithm is not suitable for distributed implementations as the data blocks $\boldsymbol{X}_1,\ldots,\boldsymbol{X}_N$ will be stored at different agents. By contrast, the Krasulina's method (Krasulina, 1969) and the Oja's method (Oja, 1982) take one data block $\boldsymbol{X}_j$ for some $j\in[N]$ in iteration t ($t\geq 0$):

$$\text{(Krasulina) } \boldsymbol{w}^{t+1} = \boldsymbol{w}^t + \eta_t\left(\boldsymbol{X}_j\boldsymbol{X}_j^\top - \frac{\|\boldsymbol{X}_j^\top\boldsymbol{w}^t\|_2^2}{\|\boldsymbol{w}^t\|_2}\right)\boldsymbol{w}^t,$$

$$\text{(Oja)} \qquad \boldsymbol{w}^{t+1} = \frac{\boldsymbol{w}^t + \eta_t\boldsymbol{X}_j\boldsymbol{X}_j^\top\boldsymbol{w}^t}{\|\boldsymbol{w}^t + \eta_t\boldsymbol{X}_j\boldsymbol{X}_j^\top\boldsymbol{w}^t\|_2}.$$

These update rules result in a reduced $\mathcal{O}(d \max_{i\in[N]} n_i)$ per-iteration computational cost. However, the Krasulina's and Oja's methods are intrinsically stochastic gradient descent (SGD)-type methods (Bottou et al., 2018), which only have sublinear convergence rates with $\eta_t = \mathcal{O}(1/t)$ (Balsubramani et al., 2013; Jain et al., 2016) (see Table 1).

To develop an algorithm that exhibits both low per-iteration computational cost and fast convergence rate, we propose the IARG-PCA method. Specifically, in iteration t ($t \geq 0$), we approximate the Riemannian gradient $\operatorname{grad} \mathcal{F}(\boldsymbol{w}^t)$ by

$$\boldsymbol{g}^t \coloneqq -\frac{1}{n}\sum_{i=1}^{N} \boldsymbol{z}_i(\boldsymbol{w}^{\tau_i(t)}), \tag{3}$$

where $\boldsymbol{z}_i(\boldsymbol{w}) \coloneqq -\left(\boldsymbol{I} - \boldsymbol{w}\boldsymbol{w}^\top\right)\boldsymbol{X}_i\boldsymbol{X}_i^\top\boldsymbol{w}$ (referred to as the i-th component Riemannian gradient at point $\boldsymbol{w} \in \mathbb{S}^{d-1}$) and $\tau_i(t)$ is the most recent iteration count at which the data block $\boldsymbol{X}_i$ is used in $\boldsymbol{g}^t$. Then, the IARG-PCA method proceeds with the following update:

$$\boldsymbol{w}^{t+1} = \frac{\boldsymbol{w}^t - \eta\boldsymbol{g}^t}{\|\boldsymbol{w}^t - \eta\boldsymbol{g}^t\|_2}. \tag{4}$$

Compared with the original RGD iteration (2), the IARG-PCA iteration (4) uses an inexact version of the Riemmanian gradient, which aggregates the information that has been computed in the previous iterations.

To control the inexactness of $\boldsymbol{g}^t$, we shall ensure that each data block is sampled at least once every T iterations for some $T \geq 0$, i.e., $(t-T)_+ \leq \tau_i(t) \leq t$ for all $i \in [N]$. For example, in the sequential update setting, this can be satisfied if the data blocks are processed one by one in a cyclic order with $\tau_i(0) = 0$ for all $i \in [N]$. This means that $T = N$ and

$$\tau_i(t) = \begin{cases} t & \text{if } i = (t-1 \bmod N) + 1, \\ \tau_i(t-1) & \text{otherwise,} \end{cases}$$

for $i \in [N]$ and $t \in \mathbb{N}_+$. Under the cyclic sampling scheme, we observe that $\tau_j(t) = t$ for some $j \in [N]$. Then, to facilitate the efficient implementation of IARG-PCA, we rewrite (3) in a recursive form:

$$\boldsymbol{g}^t = \boldsymbol{g}^{t-1} + \frac{1}{n}\boldsymbol{z}_j(\boldsymbol{w}^{\tau_j(t-1)}) - \frac{1}{n}\boldsymbol{z}_j(\boldsymbol{w}^t), \tag{5}$$

which results in $\mathcal{O}(dn_i)$ per-iteration computational cost incurred by evaluating $\boldsymbol{z}_j(\boldsymbol{w}^t)$. Since each $\boldsymbol{z}_i(\boldsymbol{w}^{\tau_i(t)})$ takes $\mathcal{O}(dn_i)$ memory for $i \in [N]$, the IARG-PCA method needs to keep $\mathcal{O}(nd)$ memory to store all the N (outdated) component Riemannian gradients.

Lastly, we compare our IARG-PCA method with the VR-PCA (Shamir, 2015) and MASAGA (Babanezhad et al., 2018) methods that extend the variance reduction techniques in SVRG (Johnson and Zhang, 2013) and SAGA (Defazio et al., 2014) to PCA and general manifold optimization problems, respectively. Specifically, the VR-PCA update formulas in iteration t are

$$\begin{aligned} \widetilde{\boldsymbol{w}}^{t+1} &= \boldsymbol{w}^t + \eta\left(\boldsymbol{X}_i\boldsymbol{X}_i^\top(\boldsymbol{w}^t - \bar{\boldsymbol{w}}^s) + \boldsymbol{u}^s\right), \\ \boldsymbol{w}^{t+1} &= \widetilde{\boldsymbol{w}}^{t+1}/\|\widetilde{\boldsymbol{w}}^{t+1}\|_2, \end{aligned}$$

where $\bar{\boldsymbol{w}}^s$ is updated once every epoch ($\mathcal{O}(N)$ iterations) and $\boldsymbol{u}^{s-1} = \frac{1}{n}\sum_{j=1}^{n}\boldsymbol{X}_j\boldsymbol{X}_j^\top\bar{\boldsymbol{w}}^{s-1}$ is evaluated based on full data pass. The update rule in the t-th iteration of MASAGA takes the following form for solving Problem (1):

$$\begin{aligned} \boldsymbol{g}^t &= \boldsymbol{z}_i(\boldsymbol{w}^t) - \mathcal{T}_{\boldsymbol{w}^0}\left(\mathcal{T}_{\boldsymbol{w}^{\tau_j(t-1)}}\left(\boldsymbol{z}_j(\boldsymbol{w}^{\tau_j(t-1)})\right) - \boldsymbol{\mu}^t\right), \\ \boldsymbol{w}^{t+1} &= \operatorname{Exp}_{\boldsymbol{w}^t}(-\eta\boldsymbol{g}_t), \end{aligned}$$

where $\boldsymbol{\mu}^t = \frac{1}{n}\sum_{i=1}^{n}\mathcal{T}_{\boldsymbol{w}^{\tau_i(t)}}\left(\boldsymbol{z}_i(\boldsymbol{w}^{\tau_i(t)})\right)$, and $\mathcal{T}_{\boldsymbol{w}}$ and $\operatorname{Exp}_{\boldsymbol{w}}$ are the parallel transport and exponential map defined as $\mathcal{T}_{\boldsymbol{w}}(\boldsymbol{z}) \coloneqq -\boldsymbol{w}\sin(\|\boldsymbol{z}\|_2) + \frac{\boldsymbol{z}}{\|\boldsymbol{z}\|_2}\cos(\|\boldsymbol{z}\|_2)$ and $\operatorname{Exp}_{\boldsymbol{w}}(\boldsymbol{z}) \coloneqq \boldsymbol{w}\cos(\|\boldsymbol{z}\|_2) + \frac{\boldsymbol{z}}{\|\boldsymbol{z}\|_2}\sin(\|\boldsymbol{z}\|_2)$ for $\boldsymbol{w} \in \mathbb{S}^{d-1}$ and $\boldsymbol{z} \in \mathbb{R}^d$, respectively. Analogous to our IARG-PCA, MASAGA also needs to keep $\mathcal{O}(nd)$ memory. However, both methods require to take data samples uniformly at random, which make them not amenable to distributed implementations and are thus essentially different from our deterministic and distributed IARG-PCA method.

2.1 Distributed Implementation

Equipped with above development, we describe the distributed implementation of the IARG-PCA method, where the data blocks are hold by N interconnected agents. The agent i maintains the latest component Riemannian gradient $\boldsymbol{z}_i(\boldsymbol{w}^{\tau_i(t)})$ and perform update (5) once it receives $\boldsymbol{w}^t$ and $\boldsymbol{g}^{t-1}$ from one of its neighborhoods. Thus, the overall memory requirement of the IARG-PCA method is shared by N different agents. The agents are visited based on a deterministic order. For a ring network, the algorithm visits the agents cyclically and thus $T = N$. For a general connected network, the algorithm visits the agents following a Hamiltonian walk and thus $T \in [N, 2N-2]$[2]. IARG-PCA treats every agent equally and does not require a central agent to dominate the other agents. Besides, no global synchronization of the whole network is needed in each round of communication. As it turns out, IARG-PCA is intrinsically a deterministic, decentralized, and asynchronous algorithm.

A formal description of the distributed implementation of the IARG-PCA method is presented as Algorithm 1. In each iteration, the IARG-PCA method incurs $\mathcal{O}(dn_i)$ computational cost (Line 4–8) and $\mathcal{O}(d)$ communication cost (Line 9). Indeed, the Krasulina's and Oja's methods can

[2]The length of a Hamiltonian walk on a connected network lies in the interval $[N, 2N-2]$ (Punnim et al., 2007).

Algorithm 1 IARG-PCA

Input: Choose initial iterate $\boldsymbol{w}^0$ and step size η. Let $\boldsymbol{g}_0 = \boldsymbol{0}$, $\tau_i(0) = 0$ and $\boldsymbol{z}_i(\boldsymbol{w}^0) = \boldsymbol{0}$ for all $i \in [N]$.
for $t = 1, 2, \dots$ **do**
  Set $\tau_{j(t)}(t) \leftarrow t$ and $\tau_i(t) \leftarrow \tau_i(t-1)$ for all agent $i \neq j(t)$ (visit agent $j(t)$)
  Set $\boldsymbol{b} \leftarrow \boldsymbol{X}_{j(t)}^\top \boldsymbol{w}^t$
  Set $\boldsymbol{z}_{j(t)}(\boldsymbol{w}^t) \leftarrow \boldsymbol{X}_{j(t)}\boldsymbol{b} - \|\boldsymbol{b}\|_2^2 \boldsymbol{w}^t$
  Set $\boldsymbol{g}^t \leftarrow \boldsymbol{g}^{t-1} + \frac{1}{n}\boldsymbol{z}_{j(t)}(\boldsymbol{w}^{\tau_{j(t)}(t-1)}) - \frac{1}{n}\boldsymbol{z}_{j(t)}(\boldsymbol{w}^t)$
  Set $\widetilde{\boldsymbol{w}}^{t+1} \leftarrow \boldsymbol{w}^t - \eta \boldsymbol{g}^t$
  Set $\boldsymbol{w}^{t+1} \leftarrow \widetilde{\boldsymbol{w}}^{t+1} / \|\widetilde{\boldsymbol{w}}^{t+1}\|_2$
  Transmit $\boldsymbol{w}^{t+1}$ and $\boldsymbol{g}^t$ to agent $j(t+1)$
end for

be implemented in the same manner, while only $\boldsymbol{w}^{t+1}$ should be transmitted after the computation phase. This yields the same per-iteration computational and communication cost as the IARG-PCA method. On the other hand, other stochastic PCA algorithms listed in Table 1 are not amenable to distributed implementations, since the variance reduction technique they use require full data pass every a particular number of iterations and uniformly sampling from whole dataset is also troublesome in the distributed setting.

In a nutshell, the algorithmic idea of the IARG-PCA method resembles but differs from that of the IAG-type methods for distributed convex optimization problems (Blatt et al., 2007; Gurbuzbalaban et al., 2017; Vanli et al., 2018; Wai et al., 2018, 2020). Different from the previous works that focus on unconstrained optimization in the Euclidean domain, the IARG-PCA method computes the Riemannian gradient and require a retraction operation (i.e., normalization in our setting (Absil et al., 2009)) in each iteration. As will be clear later in our theoretical analysis, this is crucial to make the gradient aggregation technique work on Riemannian manifolds.

Remark 1. *In practical implementations of IARG-PCA (as well as all other IAG-type methods), an agent is activated if it receives the "token" sent from its neighbor. However, the algorithm may break down if the token is lost. Although the specific communication protocols is beyond the scope of this paper, we provide a naive solution to this problem: The receiver will send an "acknowledgement" message back to the sender once it receive the token, and the sender will resend the token if it does not receive the acknowledgement.*

3 MAIN RESULTS

To present the main theoretical results, we introduce the following standing assumptions:

Assumption 1. *There exists a constant $R > 0$ such that* $\max_{1\le i\le N}\left\{\max_{1\le j\le n_i}\|\boldsymbol{x}_i^j\|_2^2\right\} \le R$.

Assumption 2. *The matrix $\boldsymbol{A}$ admits the eigendecomposition $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$, with the orthonormal matrix $\boldsymbol{V} := (\boldsymbol{v}_1, \dots, \boldsymbol{v}_d) \in \mathbb{R}^{d\times d}$ and $\boldsymbol{\Lambda} := \mathrm{Diag}(\lambda_1, \dots, \lambda_d)$ such that $\lambda_1 > \lambda_2 \ge \cdots \ge \lambda_d$.*

Assumption 3. *There exists a constant $T \ge 0$ such that for all $i \in [N]$ and $t \in \mathbb{N}$, $(t-T)_+ \le \tau_i(t) \le t$.*

Assumption 1 gives the upper bound on the norm of the data samples. In Assumption 2, the eigengap of the sample covariance is required to be positive. Notice that $\lambda_1 \le R^2$. Moreover, all agents should be visited in the past T iterations by Assumption 3, which can be satisfied if the agent activation order $\{j(1), j(2), \dots\}$ in Algorithm 1 induces a Hamiltonian walk on the network.

The suboptimality of the iterate $\boldsymbol{w}^t$ in Algorithm 1 is measured by $\mathcal{E}_t := 1 - \langle \boldsymbol{w}^t, \boldsymbol{v}_1\rangle^2$ for $t \in \mathbb{N}$. Then, we formally present the main theorem of this paper as follows:

Theorem 1. *Suppose that Assumptions 1, 2, and 3 hold. Let* $\bar{\eta} := \min\left\{\frac{1}{2\lambda_1}, \frac{1}{\sqrt{6C_1}}, \frac{\Delta}{24\sqrt{2}C_1}, \frac{24\Delta}{C_2(T+1)}, \frac{47}{\Delta}\right\}$, *where* $\Delta := \lambda_1 - \lambda_2$, $C_1 := 192\sqrt{2}R^2$, *and* $C_2 := 9216\sqrt{2}R\left(32R + \frac{\Delta}{12\sqrt{2}}\right)$. *If Algorithm 1 is initialized with $\boldsymbol{w}^0$ satisfying $\langle \boldsymbol{w}^0, \boldsymbol{v}_1\rangle \ge \frac{\sqrt{2}}{2}$ and step size satisfying $0 < \eta \le \bar{\eta}$, then it holds for all $t \ge 1$ that $\langle \boldsymbol{w}^t, \boldsymbol{v}_1\rangle \ge \frac{\sqrt{2}}{2}$ and*

$$\sqrt{\mathcal{E}_t} \le \left(1 - \frac{\Delta}{48}\eta\right)^t \sqrt{\mathcal{E}_0}. \tag{6}$$

As indicated by Theorem 1, the IARG-PCA method requires at most $\mathcal{O}\left(\frac{T}{\Delta^2}\log\left(\frac{1}{\epsilon}\right)\right)$ iterations to obtain a suboptimal solution satisfying $\langle \boldsymbol{w}^t, \boldsymbol{v}_1\rangle \ge 1 - \epsilon$ for $\epsilon \in (0, 1]$. Since $T = \mathcal{O}(N)$ for connected networks, the iteration complexity and communication complexity of IARG-PCA are both $\mathcal{O}\left(\frac{N}{\Delta^2}\log\left(\frac{1}{\epsilon}\right)\right)$.

We notice that the error bound in Theorem 1 is *deterministic* which differs from those in existing works such as (Balsubramani et al., 2013; Jain et al., 2016; Shamir, 2015). In contrast, the latter demonstrated the convergence of PCA algorithms in expectation or with high probability. The difference is due to the considerably weaker Assumption 3 required by Theorem 1 which bounds the delays deterministically. Meanwhile, prior works require unbiased (Riemannian) gradient estimates which may not be satisfied by IARG-PCA under the said assumptions.

Remark 2. *The linear convergence in Theorem 1 is based on a proper initial point. Indeed, the constant $\sqrt{2}/2$ is artificial for the conciseness of the proof. The global convergence with arbitrary initial points can be guaranteed (with high probability) by properly modifying our analysis. Specifically, for any initial point satisfying $\langle \boldsymbol{w}_0, \boldsymbol{v}_1\rangle^2 \ge 1 - \kappa$ (resp. $\langle \boldsymbol{w}_0, \boldsymbol{v}_1\rangle^2 \le \kappa - 1$) for some $\kappa > 0$, IARG-PCA will converge to $\boldsymbol{v}_1$ (resp. $-\boldsymbol{v}_1$) with a constant step*

size depending on κ and thus the linear convergence is preserved. The precise arguments will be similar to those in (Shamir, 2016, Theorem 2) and we omit it here.

4 PROOF OF THEOREM 1

In this section, we present the main steps in the proof of Theorem 1. We notice that a challenge in showing the convergence of IARG-PCA lies with the *biased* Riemannian gradients employed. To get over this, we treat the IARG-PCA method as a RGD method that carries errors, where the original linearly convergent term is perturbed by a multiplicative factor and an additive term. By carefully controlling the error terms that decrease to 0, we show that both the multiplicative and additive errors can be properly bounded and thus the linear convergence is guaranteed.

Specifically, to prove Theorem 1, it suffices to establish (6) using induction. The base case holds trivially since $\langle \boldsymbol{w}^0, \boldsymbol{v}_1\rangle \geq \sqrt{2}/2$ and thus $\mathcal{E}_0 = 1 - \langle \boldsymbol{w}^0, \boldsymbol{v}_1\rangle^2 \leq 1/2$. Suppose that $t \geq 0$ and it holds for all $\tau = 0, \ldots, t$ that

$$\langle \boldsymbol{w}^\tau, \boldsymbol{v}_1\rangle \geq \frac{\sqrt{2}}{2} \text{ and } \sqrt{\mathcal{E}_\tau} \leq \left(1 - \frac{\Delta}{48}\eta\right)^\tau \sqrt{\mathcal{E}_0}. \quad (7)$$

We will further show in this section that

$$\langle \boldsymbol{w}^{t+1}, \boldsymbol{v}_1\rangle \geq \frac{\sqrt{2}}{2} \text{ and } \sqrt{\mathcal{E}_{t+1}} \leq \left(1 - \frac{\Delta}{48}\eta\right)^{t+1} \sqrt{\mathcal{E}_0}. \quad (8)$$

We write the intermediate iterate $\widetilde{\boldsymbol{w}}^{t+1}$ in Algorithm 1 as

$$\begin{aligned}\widetilde{\boldsymbol{w}}^{t+1} &= \boldsymbol{w}^t - \eta \boldsymbol{g}^t \\ &= (\boldsymbol{I} + \eta \boldsymbol{A} - \eta \boldsymbol{w}^t (\boldsymbol{w}^t)^\top \boldsymbol{A}) \boldsymbol{w}^t + \eta \boldsymbol{e}^t, \end{aligned} \quad (9)$$

where the first term is the same as the RGD update rule (2) (without normalization) and

$$\boldsymbol{e}^t := \operatorname{grad} \mathcal{F}(\boldsymbol{w}^t) - \boldsymbol{g}^t \quad (10)$$

is the error term incurred by the gradient aggregation technique. Then, we have $\langle \widetilde{\boldsymbol{w}}^{t+1}, \boldsymbol{v}_\ell\rangle = a_\ell^t + \zeta_\ell^t$, where $a_\ell^t := \left(1 + \lambda_\ell \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t\right) \langle \boldsymbol{w}^t, \boldsymbol{v}_\ell\rangle$ and $\zeta_\ell^t := \eta \langle \boldsymbol{e}^t, \boldsymbol{v}_\ell\rangle$ for $\ell \in [d]$. Hence, it follows that

$$\begin{aligned}\mathcal{E}_{t+1} &= 1 - \langle \boldsymbol{w}^{t+1}, \boldsymbol{v}_1\rangle^2 = 1 - \frac{\langle \widetilde{\boldsymbol{w}}^{t+1}, \boldsymbol{v}_1\rangle^2}{\|\widetilde{\boldsymbol{w}}^{t+1}\|_2^2} \\ &= \frac{\sum_{\ell=2}^d \langle \widetilde{\boldsymbol{w}}^{t+1}, \boldsymbol{v}_\ell\rangle^2}{\sum_{\ell=1}^d \langle \widetilde{\boldsymbol{w}}^{t+1}, \boldsymbol{v}_\ell\rangle^2} = \frac{\sum_{\ell=2}^d \left(a_\ell^t + \zeta_\ell^t\right)^2}{\sum_{\ell=1}^d \left(a_\ell^t + \zeta_\ell^t\right)^2}. \end{aligned} \quad (11)$$

Then, taking square root for both sides of (11) and using the triangle inequality $\sqrt{\sum_{\ell=2}^d \left(a_\ell^t + \zeta_\ell^t\right)^2} \leq \sqrt{\sum_{\ell=2}^d (a_\ell^t)^2} + \sqrt{\sum_{\ell=2}^d \left(\zeta_\ell^t\right)^2}$ yield

$$\sqrt{\mathcal{E}_{t+1}} \leq \sqrt{\frac{\sum_{\ell=2}^d (a_\ell^t)^2}{\sum_{\ell=1}^d \left(a_\ell^t + \zeta_\ell^t\right)^2}} + \sqrt{\frac{\sum_{\ell=2}^d \left(\zeta_\ell^t\right)^2}{\sum_{\ell=1}^d \left(a_\ell^t + \zeta_\ell^t\right)^2}}. \quad (12)$$

Besides, using the inequality $(x+y)^2 \geq \frac{1}{1+\beta} x^2 - \frac{1}{\beta} y^2$ for all $x, y \in \mathbb{R}$ and $\beta > 0$, we have

$$\begin{aligned}\sum_{\ell=1}^d \left(a_\ell^t + \zeta_\ell^t\right)^2 &\geq \frac{1}{1+\beta} \sum_{\ell=1}^d (a_\ell^t)^2 - \frac{1}{\beta} \sum_{\ell=1}^d \left(\zeta_\ell^t\right)^2 \\ &= \sum_{\ell=1}^d (a_\ell^t)^2 \left(\frac{1}{1+\beta} - \frac{1}{\beta} \frac{\sum_{\ell=1}^d \left(\zeta_\ell^t\right)^2}{\sum_{\ell=1}^d (a_\ell^t)^2}\right) \end{aligned} \quad (13)$$

for all $\beta > 0$. Plugging (13) back into the first term of (12), it holds for all $\beta > 0$ that

$$\begin{aligned}\sqrt{\mathcal{E}_{t+1}} \leq &\sqrt{\frac{\sum_{\ell=2}^d (a_\ell^t)^2}{\sum_{\ell=1}^d (a_\ell^t)^2} \frac{1}{\frac{1}{1+\beta} - \frac{1}{\beta} \frac{\sum_{\ell=1}^d \left(\zeta_\ell^t\right)^2}{\sum_{\ell=1}^d (a_\ell^t)^2}}} \\ &+ \sqrt{\frac{\sum_{\ell=2}^d \left(\zeta_\ell^t\right)^2}{\sum_{\ell=1}^d \left(a_\ell^t + \zeta_\ell^t\right)^2}}. \end{aligned} \quad (14)$$

In (14), $\frac{\sum_{\ell=2}^d (a_\ell^t)^2}{\sum_{\ell=1}^d (a_\ell^t)^2}$ is determined by the full Riemannian gradient at $\boldsymbol{w}^t$, which is expected to converge linearly in the noiseless case. The terms $\frac{1}{1+\beta} - \frac{1}{\beta} \frac{\sum_{\ell=1}^d \left(\zeta_\ell^t\right)^2}{\sum_{\ell=1}^d (a_\ell^t)^2}$ and $\frac{\sum_{\ell=2}^d \left(\zeta_\ell^t\right)^2}{\sum_{\ell=1}^d \left(a_\ell^t + \zeta_\ell^t\right)^2}$ are respectively multiplicative and additive perturbations incurred by the error $\boldsymbol{e}^t$. Subsequently, we will bound them individually.

4.1 Bounding the Riemannian Gradient Term

Lemma 1. *Suppose that Assumption 2 holds. If $\langle \boldsymbol{w}^t, \boldsymbol{v}_1\rangle \geq \sqrt{2}/2$ and $\eta < 1/(2\lambda_1)$, then it holds for all $t \in \mathbb{N}$ that*

$$\frac{\sum_{\ell=2}^d (a_\ell^t)^2}{\sum_{\ell=1}^d (a_\ell^t)^2} \leq \left(1 - \frac{\Delta}{6}\eta\right) \mathcal{E}_t. \quad (15)$$

The proof is provided in Appendix A.2. Lemma 1 indicates that the Riemannian gradient term $\frac{\sum_{\ell=2}^d (a_\ell^t)^2}{\sum_{\ell=1}^d (a_\ell^t)^2}$ improves the suboptimality measure by a constant factor for sufficiently small η.

Remark 3. *Note that if $T = 0$ (i.e., $\tau_i(t) = t$ for all $i \in [N]$), then the IARG-PCA method reduces to the RGD method since $\zeta_\ell^t = \eta \langle \boldsymbol{e}^t, \boldsymbol{v}_\ell\rangle = 0$. In this case, the sequence $\{\mathcal{E}_t\}_{t\in\mathbb{N}}$ satisfies*

$$\mathcal{E}_{t+1} \leq \left(1 - \frac{\Delta\eta}{6}\right) \mathcal{E}_t, \quad (16)$$

which indicates that the RGD method converges Q-linearly provided that $\eta < \min\{6/\Delta, 1/(2\lambda_1)\}$. This gives iteration complexity of order $\mathcal{O}\left(\frac{1}{\Delta} \log\left(\frac{1}{\epsilon}\right)\right)$, which matches the best-known results given by Ding et al. (2020); Xu and Li (2021).[3] It could be of independent interest that the proof of Lemma 1 appears to be neater than the previous ones.

[3] Indeed, the state-of-the-art result in Xu and Li (2021) shows

4.2 Bounding the Perturbation Terms

We then upper bound the multiplicative and additive perturbations in (14). As a preparation, we bound $\|\boldsymbol{e}^t\|_2$ in the following lemma.

Lemma 2. *Suppose that Assumptions 1 and 3 hold. If $\eta \leq 1/(2R)$, then it holds for all $t \in \mathbb{N}$ that*

$$\|\boldsymbol{e}^t\|_2 \leq 16R\eta \sum_{s=(t-T)_+}^{t-1} \|\boldsymbol{g}^s\|_2, \tag{17}$$

$$\leq C_1 T\eta \max_{(t-2T)_+ \leq j \leq t} \sqrt{\mathcal{E}_j}, \tag{18}$$

where C_1 is the same constant as in Theorem 1.

The proof is deferred to Appendix A.3. Lemma 2 gives two useful upper bounds on $\boldsymbol{e}^t$. The tighter bound (17) will be used to establish final recurrence relation while the bound (18) will be used to control the perturbation terms.

Remark 4. *The proof of Lemma 2 crucially relies on the fact that $\|\operatorname{grad} \mathcal{F}(\boldsymbol{w}^s)\|_2 \leq \mathcal{O}\left(\sqrt{\mathcal{E}_s}\right)$, which leads to an upper bound on the error norm $\|\boldsymbol{e}^t\|_2$ that diminishes to 0. This property theoretically motivates the use of Riemannian gradients in our approach as oppose to the Euclidean ones, which is one of the distinct features of our IARG-PCA method compared to the IAG-type methods.*

Equipped with Lemma 2, we obtain the following lemma that bounds the multiplicative and additive perturbations incurred by $\boldsymbol{e}^t$.

Lemma 3. *Suppose that Assumptions 1 and 3 hold. If $\eta \leq \min\{1/(2R), 1/(2\lambda_1), 1/\sqrt{6C_1}\}$, then it holds for all $t \in \mathbb{N}$ and $\beta > 0$ that*

$$\frac{1}{1+\beta} - \frac{1}{\beta}\frac{\sum_{\ell=1}^d (\zeta_\ell^t)^2}{\sum_{\ell=1}^d (a_\ell^t)^2} \geq 1 - \beta - \frac{2C_1^2T^2}{\beta}\eta^4, \tag{19}$$

$$\frac{\sum_{\ell=2}^d (\zeta_\ell^t)^2}{\sum_{\ell=1}^d (a_\ell^t + \zeta_\ell^t)^2} \leq 4\eta^2\|\boldsymbol{e}^t\|_2^2, \tag{20}$$

where C_1 is the same constant as in Theorem 1.

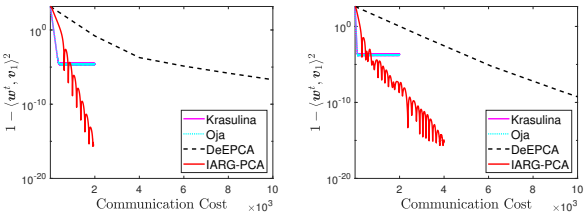
4.3 Solving the Recurrence Relation

Armed with (14) and Lemmas 1–3, we can obtain the following recurrence relation of the sequence $\{\sqrt{\mathcal{E}_t}\}_{t\in\mathbb{N}}$.

Lemma 4. *Suppose that Assumptions 1–3 hold. If $\eta \leq \min\left\{1/(2\lambda_1), 1/\sqrt{6C_1}, \Delta/(24\sqrt{2}C_1)\right\}$, then it holds for all $t \in \mathbb{N}$ that*

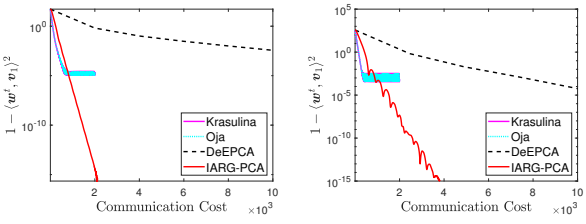
$$\sqrt{\mathcal{E}_{t+1}} \leq \left(1 - \frac{\Delta}{48}\eta\right)\sqrt{\mathcal{E}_t} - \frac{\Delta}{192\sqrt{2}R}\eta\|\boldsymbol{g}^t\|_2 + \left(32R + \frac{\Delta}{12\sqrt{2}}\right)\eta^2 \sum_{s=(t-T)_+}^{t} \|\boldsymbol{g}^s\|_2. \tag{21}$$

The proof of Lemma 4 is given in Appendix A.5. Equipped with Lemma 4, it suffices to apply (Aytekin et al., 2016, Lemma 1) to solve the recurrence (21), which leads the iteration complexity to scale linearly with the delay T. Indeed, the perturbation terms in (21) represented by sequence $\{\|\boldsymbol{g}^t\|_2\}_{t\in\mathbb{N}}$ do not disrupt the exponential decrease of the system $\sqrt{\mathcal{E}_{t+1}} \leq \left(1 - \frac{\Delta}{48}\eta\right)\sqrt{\mathcal{E}_t}$. The detailed derivation that completes the proof of Theorem 1 is deferred to Appendix A.6.

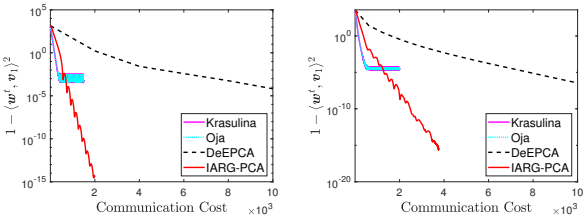


(a) Ring (b) Erdős–Rényi

Figure 1: Decentralized PCA on synthetic data.



(a) Ring (a9a) (b) Erdős–Rényi (w8a)



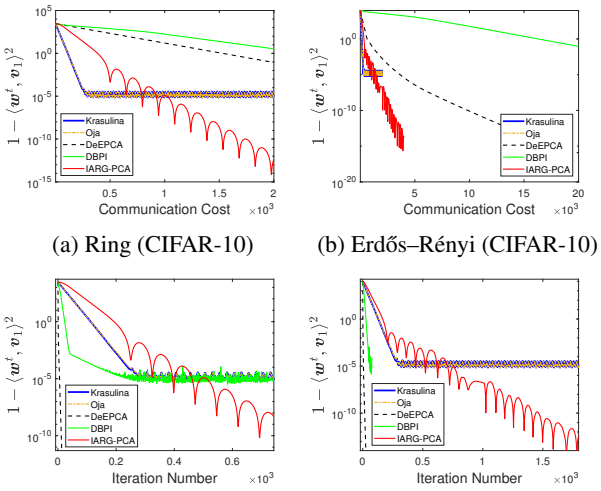
(c) Ring (w8a) (d) Erdős–Rényi (a9a)

Figure 2: Decentralized PCA on a9a and w8a datasets.

5 EXPERIMENTS

In this section, we test the numerical performance of our IARG-PCA method on both synthetic and real datasets and compare it with that of the state-of-the-art distributed PCA algorithms and incremental PCA algorithms, respectively. For each experiment on synthetic data, we independently generate n data points according to the spiked model for

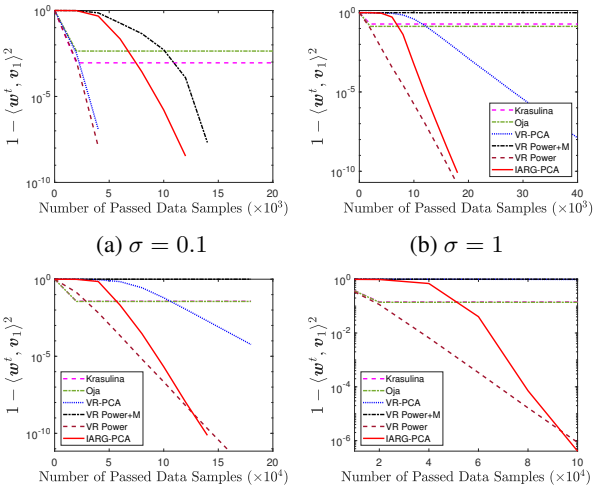
that the iteration complexity of the RGD method for PCA is $\mathcal{O}\left(\frac{1}{\max\{\Delta,\epsilon\}}\log\left(\frac{1}{\epsilon}\right)\right)$, which is the same as ours in the high-precision regime where $\Delta > \epsilon$.



(a) Ring (CIFAR-10) (b) Erdős–Rényi (CIFAR-10)

(c) Ring (CIFAR-10) (d) Erdős–Rényi (CIFAR-10)

Figure 3: Decentralized PCA on CIFAR-10 dataset.



(a) $\sigma = 0.1$ (b) $\sigma = 1$

(c) $\sigma = 0.5$ (d) $\sigma = 1$

Figure 5: Incremental PCA on synthetic data (first row: $n = 5000, d = 100$; second row: $n = 50000, d = 1000$).

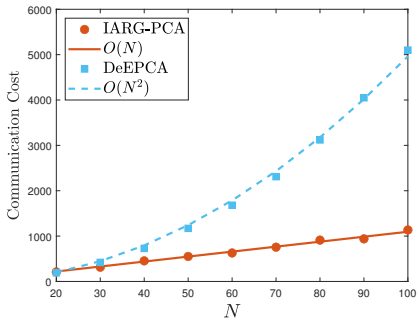


Figure 4: Dependence of communication cost on N.

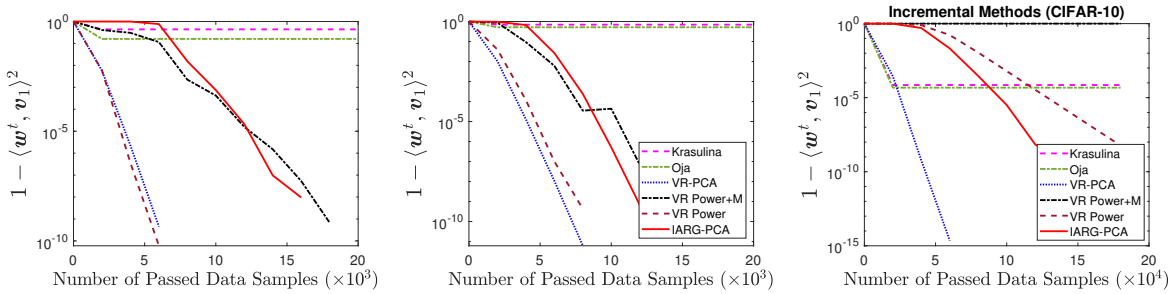
PCA (Johnstone, 2001): $\boldsymbol{x}_i = r_i \boldsymbol{w}^* + \boldsymbol{\varepsilon}_i$ for $i = 1, \ldots, n$, where $\boldsymbol{w}^* \in \mathbb{R}^d$ is a random vector uniformly sampled over the unit sphere $\mathbb{S}^{d-1}$, $r_i \sim \mathcal{N}(0, 1)$, and $\boldsymbol{\varepsilon}_i$ is the noise vector composed of i.i.d. $\mathcal{N}(0, \sigma^2)$ entries with $\sigma > 0$ being the noise level. For the real-data experiments, we use the datasets from the LIBSVM library (Chang and Lin, 2011). Our code is available at https://github.com/xwangcu/iarg-pca.

5.1 Numerical Results of Distributed PCA

In this subsection, we focus on the performance of IARG-PCA in the distributed setting and compare it with the state-of-the-art decentralized PCA algorithm, i.e., DeEPCA (Ye and Zhang, 2021). We also implement the Krasulina's and Oja's methods in a distributed manner as baselines. We compare their performance in terms of communication cost on two common types of networks, namely the ring network and the Erdős–Rényi network (with connectivity probability $p = 0.5$). The weight matrix associated with network is defined as $\boldsymbol{W} = \boldsymbol{I} - \boldsymbol{L} / \max_{i \in [N]} D_i$, where D_i denotes the degree of agent i and $\boldsymbol{L}$ is the Laplacian matrix of the network. Then, we have $\lambda_2(\boldsymbol{W}) = 0.99$ for the ring network and $\lambda_2(\boldsymbol{W}) = 0.70$ for the Erdős–Rényi network. We define the communication cost as the number of d-dimensional vectors transmitted on the networks.

For synthetic data, we set $d = 50$, $N = 50$, and $n_i = 50$ for all $i \in [N]$. For real-world datasets 'a9a' ($d = 300$ and $n = 49749$) (resp. 'w8a' ($d = 123$ and $n = 32561$)), we equally assign 995 (resp. 650) data samples to agents $1, \ldots, N-1$. The remaining data samples are assigned to agent N. Figures 1 and 2 present how the suboptimality measure $1 - \langle \boldsymbol{w}^t, \boldsymbol{v}_1 \rangle^2$ of different algorithms decreases with the communication cost on synthetic and real datasets, respectively. As we can see, the baseline Krasulina's and Oja's methods convergence sublinearly. By contrast, the IARG-PCA and DeEPCA methods exhibit linear rates of convergence. It is noteworthy that IARG-PCA takes significantly fewer communication cost than DeEPCA to reach certain high precision. This indicates that our IARG-PCA method is substantially more communication-efficient than other decentralized PCA algorithms.

We provide additional numerical results on the larger CIFAR-10 dataset with $n = 50000$ and $d = 3072$ in Figure 3. The data samples are distributed in the same manner as done for the 'w8a' and 'a9a' datasets. Moreover, we implement the recently proposed distributed Banach-Picard iteration (DBPI) (Andrade et al., 2023) that can also be applied to solve the PCA problem. We observe from Figures 3(a) and 3(b) that IARG-PCA achieves up to 10^{-15} precision with the least communication cost than all other algorithms. However, Figures 3(c) and 3(d) show that IARG-PCA converges slower than DeEPCA and DBPI. This is because the latter two algorithms requires consensus communication in each iteration, which facilitate the convergence



(a) a1a (b) w1a (c) CIFAR-10

Figure 6: Incremental PCA on real datasets.

while bring huge communication cost in the meantime.

Moreover, we evaluate the expected communication cost of the IARG-PCA and DeEPCA methods to reach 10^{-12} precision for different numbers of agents N. We conduct experiments on synthetic data over ring networks. Figure 4 shows that the communication cost of DeEPCA scales with $\mathcal{O}(N^2)$ while IARG-PCA scales with $\mathcal{O}(N)$, which verifies the theoretical results presented in Table 2. Therefore, our IARG-PCA method exhibits high scalability and thus enjoys low communication cost for large networks.

5.2 Numerical Results of Incremental PCA

Although our IARG-PCA method is developed for distributed environments, it can be implemented on a single machine as discussed in Section 2. Thus, we report the convergence performance of IARG-PCA together with some representative incremental PCA algorithms, including the Krasulina's method (Krasulina, 1969), the Oja's method (Oja, 1982), VR-PCA (Shamir, 2015), VR-Power+M (Xu et al., 2018a), and VR Power (Kim and Klabjan, 2020). For the Krasulina's and Oja's methods, we adopt diminishing step sizes $\eta_t = \theta/t$ and best tune the hyperparameter θ. For VR-PCA, we set the epoch length and the step size as the recommended values n and $\sqrt{n}/\sum_{i=1}^n \|\boldsymbol{x}_i\|_2^2$, respectively. For VR Power+M method, we set its parameter β as the optimal value $\lambda_2^2/4$. For VR Power and IARG-PCA, we best tune their constant step sizes. Lastly, all algorithms take only one data sample in each iteration.

Figure 5 presents the convergence of $1-\langle \boldsymbol{w}^t, \boldsymbol{v}_1\rangle^2$ on synthetic data with $n = 5000$, $d = 100$ and $n = 50000$, $d = 1000$. IARG-PCA is comparable with VR-Power+M and inferior to VR-PCA and VR-Power if the noise level is small (the eigengap is large), while becomes much faster than VR-PCA and VR-Power+M if the noise is large (the eigengap is small). This is because the iteration complexity of all incremental algorithms is dominated by the $1/\Delta^2$ term if the eigengap Δ is small (see Table 1), which may lead to significant performance deterioration of the stochastic PCA algorithms. By contrast, our IARG-PCA appears to be more robust to the decrease of eigengap.

Figure 6 presents the results on two real datasets 'a1a' ($d = 123, n = 1605$) and 'w1a' ($d = 300, n = 2477$). The performance of our IARG-PCA method is basically comparable with that of the VR Power+M method. The numerical results on synthetic and real data illustrate that our IARG-PCA method is computationally efficient when it runs on a single machine.

6 CONCLUSION & DISCUSSION

We have developed an efficient PCA algorithm that is amenable to decentralized and asynchronous implementations. We have proved that the proposed IARG-PCA method exhibits $\mathcal{O}\left(\frac{N}{\Delta^2}\log\left(\frac{1}{\epsilon}\right)\right)$ iteration/communication complexity. The effectiveness of our proposed method has also been justified through numerical experiments.

We leave several interesting problems for future work. Although the communication complexity of IARG-PCA can be much better than that of the DeEPCA if the eigengap Δ is not very small (see Table 2), it would be more favorable if we can improve it to $\mathcal{O}\left(\frac{N}{\Delta}\log\left(\frac{1}{\epsilon}\right)\right)$. Nevertheless, the $1/\Delta^2$ factor seems to be fundamental to incremental PCA algorithms. This is evidenced in Table 1, where all the state-of-the-art incremental PCA algorithms require the total runtime to scale with $\mathcal{O}(1/\Delta^2)$. Moreover, this paper only focused on the one-dimensional PCA problem. Although 1-PCA methods can be used to obtain other principal components by repeatedly applying the well-known *deflation* technique, it would be meaningful to extend our algorithm to directly tackle the general k-PCA problem ($k > 1$) with the Stiefel manifold constraint.

Acknowledgements

The work of X. Wang and H.-T. Wai was supported in part by the HKRGC Project #24203520. The work of Y. Jiao and Y. Gu was supported in part by the National Natural Science Foundation of China under Grant U2230201 and 61971266, Grant from the Guoqiang Institute, Tsinghua University, and in part by the Clinical Medicine Development Fund of Tsinghua University.

References

P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009.

F. Alimisis, P. Davies, B. Vandereycken, and D. Alistarh. Distributed principal component analysis with limited communication. In *Advances in Neural Information Processing Systems 34*, pages 2823–2834, 2021.

F. Andrade, M. A. Figueiredo, and J. ao Xavier. Distributed Banach-Picard iteration: Application to distributed parameter estimation and PCA. *IEEE Transactions on Signal Processing*, 2023.

M. Assran, A. Aytekin, H. R. Feyzmahdavian, M. Johansson, and M. G. Rabbat. Advances in asynchronous parallel and distributed optimization. *Proceedings of the IEEE*, 108(11):2013–2031, 2020.

A. Aytekin, H. R. Feyzmahdavian, and M. Johansson. Analysis and implementation of an asynchronous optimization algorithm for the parameter server. *arXiv preprint arXiv:1610.05507*, 2016.

R. Babanezhad, I. H. Laradji, A. Shafaei, and M. Schmidt. MASAGA: A linearly-convergent stochastic first-order method for optimization on manifolds. In *Proceedings of the 2018 Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 344–359. Springer, 2018.

A. Balsubramani, S. Dasgupta, and Y. Freund. The fast convergence of incremental PCA. 2013.

D. Blatt, A. O. Hero, and H. Gauchman. A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, 18(1):29–51, 2007.

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60 (2):223–311, 2018.

C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, 2011.

S. Chen, A. Garcia, M. Hong, and S. Shahrampour. Decentralized Riemannian gradient descent on the Stiefel manifold. In *Proceedings of the 38th International Conference on Machine Learning*, pages 1594–1605. PMLR, 2021.

J. P. Cunningham and B. M. Yu. Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, 17(11):1500–1509, 2014.

A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems 27*, 2014.

Q. Ding, K. Zhou, and J. Cheng. Tight convergence rate of gradient descent for eigenvalue computation. In *Proceedings of the 29th International Joint Conferences on Artificial Intelligence*, pages 3276–3282, 2020.

M. W. Dorrity, L. M. Saunders, C. Queitsch, S. Fields, and C. Trapnell. Dimensionality reduction by UMAP to visualize physical and genetic interactions. *Nature Communications*, 11(1):1–6, 2020.

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*, 7:699–715, 2021.

D. Garber, E. Hazan, C. Jin, C. Musco, P. Netrapalli, A. Sidford, et al. Faster eigenvector computation via shift-and-invert preconditioning. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2626–2634. PMLR, 2016.

A. Gittens, A. Devarakonda, E. Racah, M. Ringenburg, L. Gerhardt, J. Kottalam, J. Liu, K. Maschhoff, S. Canon, J. Chhugani, et al. Matrix factorizations at scale: A comparison of scientific data analytics in spark and C+MPI using three case studies. In *Proceedings of 2016 IEEE International Conference on Big Data*, pages 204–213. IEEE, 2016.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. JHU Press, 2013.

A. Grammenos, R. Mendoza Smith, J. Crowcroft, and C. Mascolo. Federated principal component analysis. In *Advances in Neural Information Processing Systems 33*, pages 6453–6464, 2020.

M. Gurbuzbalaban, A. Ozdaglar, and P. A. Parrilo. On the convergence rate of incremental aggregated gradient algorithms. *SIAM Journal on Optimization*, 27(2):1035–1048, 2017.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933.

L.-K. Huang and S. Pan. Communication-efficient distributed PCA by Riemannian optimization. In *Proceedings of the 37th International Conference on Machine Learning*, pages 4465–4474. PMLR, 2020.

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 *Proceedings of the 29th Conference on Learning theory*, pages 1147–1164. PMLR, 2016.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, 2013.

I. M. Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *The Annals of Statistics*, 29(2):295–327, 2001.

C. Kim and D. Klabjan. Stochastic variance-reduced algorithms for PCA with arbitrary mini-batch sizes. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, pages 4302–4312. PMLR, 2020.

T. Krasulina. The method of stochastic approximation for the determination of the least eigenvalue of a symmetrical matrix. *USSR Computational Mathematics and Mathematical Physics*, 9(6):189–195, 1969.

X. Li, S. Wang, K. Chen, and Z. Zhang. Communication-efficient distributed SVD via local power iterations. In *Proceedings of the 38th International Conference on Machine Learning*, pages 6504–6514. PMLR, 2021.

Z. Liu and V. Y. Tan. The informativeness of k-means for learning mixture models. *IEEE Transactions on Information Theory*, 65(11):7460–7479, 2019.

J. Ma and Y. Yuan. Dimension reduction of image deep feature using PCA. *Journal of Visual Communication and Image Representation*, 63:102578, 2019.

E. Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, 1982.

N. Punnim, V. Saenpholphat, and S. Thaithae. Almost Hamiltonian cubic graphs. *International Journal of Computer Science and Network Security*, 7(1):83–86, 2007.

O. Shamir. A stochastic PCA and SVD algorithm with an exponential convergence rate. In *Proceedings of the 18th International Conference on Machine Learning*, pages 144–152. PMLR, 2015.

O. Shamir. Fast stochastic algorithms for SVD and PCA: Convergence properties and convexity. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 248–256. PMLR, 2016.

D. Spielman. *Course Notes on Spectral Graph Theory*. 2015. URL http://www.cs.yale.edu/homes/spielman/561/.

N. D. Vanli, M. Gurbuzbalaban, and A. Ozdaglar. Global convergence rate of proximal incremental aggregated gradient methods. *SIAM Journal on Optimization*, 28 (2):1282–1300, 2018.

H.-T. Wai, A. Scaglione, J. Lafond, and E. Moulines. Fast and privacy preserving distributed low-rank regression. In *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4451–4455. IEEE, 2017.

H.-T. Wai, N. M. Freris, A. Nedic, and A. Scaglione. SUCAG: Stochastic unbiased curvature-aided gradient method for distributed optimization. In *Proceedings of the 2018 IEEE Conference on Decision and Control*, pages 1751–1756. IEEE, 2018.

H.-T. Wai, W. Shi, C. A. Uribe, A. Nedić, and A. Scaglione. Accelerating incremental gradient optimization with curvature information. *Computational Optimization and Applications*, 76(2):347–380, 2020.

S. X. Wu, H.-T. Wai, L. Li, and A. Scaglione. A review of distributed algorithms for principal component analysis. *Proceedings of the IEEE*, 106(8):1321–1340, 2018.

P. Xu, B. He, C. De Sa, I. Mitliagkas, and C. Re. Accelerated stochastic power iteration. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, pages 58–67. PMLR, 2018a.

Z. Xu and P. Li. A comprehensively tight analysis of gradient descent for PCA. In *Advances in Neural Information Processing Systems 34*, pages 21935–21946, 2021.

Z. Xu, X. Cao, and X. Gao. Convergence analysis of gradient descent for eigenvector computation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2933–2939, 2018b.

H. Ye and T. Zhang. DeEPCA: Decentralized exact PCA with linear convergence rate. *Journal of Machine Learning Research*, 22(238):1–27, 2021.

Appendix

A MISSING PROOFS

In this section, we provide the proofs that are missing in the main text. We use $\|\cdot\|_2$ to denote either the ℓ_2-norm of a vector or the spectral norm of a matrix.

A.1 Technical Lemmas

We introduce the following technical lemmas that will be used multiple times in our analysis.

Lemma 5. *Suppose that Assumption 1 holds. If $\boldsymbol{u}, \boldsymbol{u}' \in \mathbb{R}^d$ satisfy $\|\boldsymbol{u}\|_2 = \|\boldsymbol{u}'\|_2 = 1$ and $\boldsymbol{B} \in \mathbb{R}^{d\times d}$ satisfies $\|\boldsymbol{B}\|_2 \leq M$ for some $M > 0$, then it holds that*

$$\left\|(\boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^\top)\boldsymbol{B}\boldsymbol{u} - (\boldsymbol{I} - \boldsymbol{u}'(\boldsymbol{u}')^\top)\boldsymbol{B}\boldsymbol{u}'\right\|_2 \leq 4M\|\boldsymbol{u} - \boldsymbol{u}'\|_2.$$

Proof. We first note that

$$\begin{aligned}
\left\|\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{B}\boldsymbol{u} - \boldsymbol{u}'(\boldsymbol{u}')^\top\boldsymbol{B}\boldsymbol{u}'\right\|_2 &= \left\|\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{B}(\boldsymbol{u} - \boldsymbol{u}') + \boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{B}\boldsymbol{u}' - \boldsymbol{u}'(\boldsymbol{u}')^\top\boldsymbol{B}\boldsymbol{u}'\right\|_2 \\
&= \left\|\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{B}(\boldsymbol{u} - \boldsymbol{u}') + (\boldsymbol{u} - \boldsymbol{u}')\boldsymbol{u}^\top\boldsymbol{B}\boldsymbol{u}' + \boldsymbol{u}'(\boldsymbol{u} - \boldsymbol{u}')^\top\boldsymbol{B}\boldsymbol{u}'\right\|_2 \\
&\leq \|\boldsymbol{u}\boldsymbol{u}^\top\|_2\|\boldsymbol{B}\|_2\|\boldsymbol{u} - \boldsymbol{u}'\|_2 + \|\boldsymbol{u} - \boldsymbol{u}'\|_2\|\boldsymbol{u}\|_2\|\boldsymbol{B}\|_2\|\boldsymbol{u}'\|_2 + \|\boldsymbol{u}'\|_2\|\boldsymbol{u} - \boldsymbol{u}'\|_2\|\boldsymbol{B}\|_2\|\boldsymbol{u}'\|_2 \\
&\leq 3M\|\boldsymbol{u} - \boldsymbol{u}'\|_2,
\end{aligned}$$

where the second inequality is due to $\|\boldsymbol{u}\|_2 = \|\boldsymbol{u}'\|_2 = 1$, $\|\boldsymbol{u}\boldsymbol{u}^\top\|_2 = \|\boldsymbol{u}\|_2^2 = 1$, and $\|\boldsymbol{B}\|_2 \leq M$. This implies that

$$\begin{aligned}
\left\|(\boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^\top)\boldsymbol{B}\boldsymbol{u} - (\boldsymbol{I} - \boldsymbol{u}'(\boldsymbol{u}')^\top)\boldsymbol{B}\boldsymbol{u}'\right\|_2 &\leq \|\boldsymbol{B}(\boldsymbol{u} - \boldsymbol{u}')\|_2 + \left\|\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{B}\boldsymbol{u} - \boldsymbol{u}'(\boldsymbol{u}')^\top\boldsymbol{B}\boldsymbol{u}'\right\|_2 \\
&\leq \|\boldsymbol{B}\|_2\|\boldsymbol{u} - \boldsymbol{u}'\|_2 + \left\|\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{B}\boldsymbol{u} - \boldsymbol{u}'(\boldsymbol{u}')^\top\boldsymbol{B}\boldsymbol{u}'\right\|_2 \\
&\leq 4M\|\boldsymbol{u} - \boldsymbol{u}'\|_2,
\end{aligned}$$

as desired. □

We then introduce the following lemma that will be used to bound the error norm $\|\boldsymbol{e}^t\|_2$.

Lemma 6. *Suppose that Assumption 1 holds and $\eta < 1/(2R)$. Then, it holds that*

$$\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|_2 \leq 4\eta\|\boldsymbol{g}^t\|_2.$$

Proof. We first upper bound $\|\boldsymbol{g}^t\|_2$ as

$$\begin{aligned}
\|\boldsymbol{g}^t\|_2 &= \left\|\frac{1}{n}\sum_{i=1}^N \boldsymbol{z}_i(\boldsymbol{w}^{\tau_i(t)})\right\|_2 \\
&\leq \frac{1}{n}\sum_{i=1}^N \left\|\left(\boldsymbol{I} - \boldsymbol{w}^{\tau_i(t)}(\boldsymbol{w}^{\tau_i(t)})^\top\right)\boldsymbol{X}_i\boldsymbol{X}_i^\top\boldsymbol{w}^{\tau_i(t)}\right\|_2 \\
&\leq \frac{1}{n}\sum_{i=1}^N \left\|\boldsymbol{I} - \boldsymbol{w}^{\tau_i(t)}(\boldsymbol{w}^{\tau_i(t)})^\top\right\|_2 \left\|\boldsymbol{X}_i\boldsymbol{X}_i^\top\right\|_2 \|\boldsymbol{w}^{\tau_i(t)}\|_2 \\
&\leq \frac{1}{n}\sum_{i=1}^N n_i R = R,
\end{aligned} \tag{22}$$

where the last inequality holds because $\left\|\boldsymbol{I} - \boldsymbol{w}^{\tau_i(t)}(\boldsymbol{w}^{\tau_i(t)})^\top\right\|_2 = 1$, $\left\|\boldsymbol{X}_i\boldsymbol{X}_i^\top\right\|_2 \leq \sum_{j=1}^{n_i}\|\boldsymbol{x}_i^j(\boldsymbol{x}_i^j)^\top\|_2 = \sum_{j=1}^{n_i}\|\boldsymbol{x}_i^j\|_2^2 \leq n_i R$ by Assumption 1, and $\|\boldsymbol{w}^{\tau_i(t)}\|_2^2 = 1$, the last equality is due to $\sum_{i=1}^N n_i = n$. Then, it follows that

$$\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|_2 = \left\|\frac{\boldsymbol{w}^t - \eta\boldsymbol{g}^t}{\|\boldsymbol{w}^t - \eta\boldsymbol{g}^t\|_2} - \boldsymbol{w}^t\right\|_2$$

$$
\begin{aligned}
&= \left\| \left(\frac{1}{\|\boldsymbol{w}^t - \eta \boldsymbol{g}^t\|_2} - 1 \right) \boldsymbol{w}^t + \frac{\eta \boldsymbol{g}^t}{\|\boldsymbol{w}^t - \eta \boldsymbol{g}^t\|_2} \right\|_2 \\
&\le \left(\frac{1}{\|\boldsymbol{w}^t - \eta \boldsymbol{g}^t\|_2} - 1 \right) \|\boldsymbol{w}^t\|_2 + \frac{\eta \|\boldsymbol{g}^t\|_2}{\|\boldsymbol{w}^t - \eta \boldsymbol{g}^t\|_2} \\
&\le \frac{1}{\|\boldsymbol{w}^t\| - \eta \|\boldsymbol{g}^t\|_2} - 1 + \frac{\eta \|\boldsymbol{g}^t\|_2}{\|\boldsymbol{w}^t\| - \eta \|\boldsymbol{g}^t\|_2} \\
&= \frac{2\eta \|\boldsymbol{g}^t\|_2}{1 - \eta \|\boldsymbol{g}^t\|_2} \le 4\eta \|\boldsymbol{g}^t\|_2,
\end{aligned}
$$

where the last inequality holds because (22) and $\eta \le 1/(2R)$ implies that $1/(1 - \eta\|\boldsymbol{g}^t\|_2) \le 2$. □

A.2 Proof of Lemma 1

Proof. By Assumption 2, we have $\sum_{\ell=1}^d \langle \boldsymbol{w}^t, \boldsymbol{v}_\ell \rangle^2 = \boldsymbol{w}^t \boldsymbol{V}\boldsymbol{V}^\top \boldsymbol{w}^t = \|\boldsymbol{w}^t\|_2 = 1$, which implies that

$$
\sum_{\ell=2}^d \langle \boldsymbol{w}^t, \boldsymbol{v}_\ell \rangle^2 = 1 - \langle \boldsymbol{w}^t, \boldsymbol{v}_1 \rangle^2 = \mathcal{E}_t.
$$

This, together with the definition of a_ℓ^t for $\ell \in [d]$, gives

$$
\begin{aligned}
\sum_{\ell=1}^d (a_\ell^t)^2 &= \left(1 + \lambda_1 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t\right)^2 \langle \boldsymbol{w}^t, \boldsymbol{v}_1 \rangle^2 + \sum_{\ell=2}^d \left(1 + \lambda_\ell \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t\right)^2 \langle \boldsymbol{w}^t, \boldsymbol{v}_\ell \rangle^2 \\
&\le \left(1 + \lambda_1 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t\right)^2 \langle \boldsymbol{w}^t, \boldsymbol{v}_1 \rangle^2 + \left(1 + \lambda_2 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t\right)^2 \sum_{\ell=2}^d \langle \boldsymbol{w}^t, \boldsymbol{v}_\ell \rangle^2 \\
&= \left(1 + \lambda_1 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t\right)^2 (1 - \mathcal{E}_t) + \left(1 + \lambda_2 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t\right)^2 \mathcal{E}_t.
\end{aligned}
$$

Therefore, we have

$$
\begin{aligned}
\frac{(a_1^t)^2}{\sum_{\ell=1}^d (a_\ell^t)^2} &\ge \frac{\left(1 + \lambda_1 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t\right)^2 (1 - \mathcal{E}_t)}{\left(1 + \lambda_1 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t\right)^2 (1 - \mathcal{E}_t) + \left(1 + \lambda_2 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t\right)^2 \mathcal{E}_t} \\
&\ge \frac{1 - \mathcal{E}_t}{1 - \mathcal{E}_t + \left(\frac{1 + \lambda_2 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t}{1 + \lambda_1 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t}\right)^2 \mathcal{E}_t} \\
&\ge \frac{1 - \mathcal{E}_t}{1 - \mathcal{E}_t + \frac{1 + \lambda_2 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t}{1 + \lambda_1 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t} \mathcal{E}_t} \qquad (23) \\
&= \frac{1 - \mathcal{E}_t}{1 - \left(1 - \frac{1 + \lambda_2 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t}{1 + \lambda_1 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t}\right) \mathcal{E}_t}, \qquad (24)
\end{aligned}
$$

where (23) holds because $\lambda_1 > \lambda_2$ implies that $\left(\frac{1 + \lambda_2 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t}{1 + \lambda_1 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t}\right)^2 < \frac{1 + \lambda_2 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t}{1 + \lambda_1 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t} < 1$. Since $\left(1 - \frac{1 + \lambda_2 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t}{1 + \lambda_1 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t}\right) \mathcal{E}_t < 1$, then applying inequality $1/(1 - x) \ge 1 + x$ for all $x < 1$ to (24) gives

$$
\begin{aligned}
\frac{(a_1^t)^2}{\sum_{\ell=1}^d (a_\ell^t)^2} &\ge (1 - \mathcal{E}_t) \left(1 + \left(1 - \frac{1 + \lambda_2 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t}{1 + \lambda_1 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t}\right) \mathcal{E}_t\right) \\
&= (1 - \mathcal{E}_t) \left(1 + \frac{(\lambda_1 - \lambda_2)\eta}{1 + \lambda_1 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t} \mathcal{E}_t\right) \\
&= 1 - \mathcal{E}_t + \frac{(\lambda_1 - \lambda_2)\eta}{1 + \lambda_1 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t} \mathcal{E}_t - \frac{(\lambda_1 - \lambda_2)\eta}{1 + \lambda_1 \eta - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t} \mathcal{E}_t^2. \qquad (25)
\end{aligned}
$$

Since $0 \le (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t \le \lambda_1$ due to the Courant-Fischer theorem, we have

$$
1 \le 1 + \eta \lambda_1 - \eta (\boldsymbol{w}^t)^\top \boldsymbol{A} \boldsymbol{w}^t \le 1 + \eta \lambda_1. \qquad (26)
$$

Combining (25) and (26) gives that

$$\begin{aligned}\frac{\sum_{\ell=2}^{d}(a_\ell^t)^2}{\sum_{\ell=1}^{d}(a_\ell^t)^2} &= 1-\frac{(a_1^t)^2}{\sum_{\ell=1}^{d}(a_\ell^t)^2}\\ &\le \mathcal{E}_t-\frac{(\lambda_1-\lambda_2)\eta}{1+\lambda_1\eta-\eta(\boldsymbol{w}^t)^\top\boldsymbol{A}\boldsymbol{w}^t}\mathcal{E}_t+\frac{(\lambda_1-\lambda_2)\eta}{1+\lambda_1\eta-\eta(\boldsymbol{w}^t)^\top\boldsymbol{A}\boldsymbol{w}^t}\mathcal{E}_t^2\\ &\le \mathcal{E}_t-\frac{(\lambda_1-\lambda_2)\eta}{1+\lambda_1\eta}\mathcal{E}_t+(\lambda_1-\lambda_2)\eta\mathcal{E}_t^2\\ &= \left(1-\frac{(\lambda_1-\lambda_2)\eta}{1+\lambda_1\eta}+(\lambda_1-\lambda_2)\eta\mathcal{E}_t\right)\mathcal{E}_t\\ &\le \left(1-\frac{2(\lambda_1-\lambda_2)\eta}{3}+\frac{(\lambda_1-\lambda_2)\eta}{2}\right)\mathcal{E}_t \qquad (27)\\ &= \left(1-\frac{\Delta\eta}{6}\right)\mathcal{E}_t,\end{aligned}$$

where (27) holds because $\mathcal{E}_t = 1-\langle \boldsymbol{w}^t, \boldsymbol{v}_1\rangle^2 \le 1/2$ and $1+\lambda_1\eta \le 3/2$ due to $\eta < 1/(2\lambda_1)$. □

A.3 Proof of Lemma 2

Proof. We first note that

$$\begin{aligned}\|\boldsymbol{e}^t\|_2 &= \|\operatorname{grad}\mathcal{F}(\boldsymbol{w}^t)-\boldsymbol{g}^t\|_2\\ &= \left\|\frac{1}{n}\sum_{i=1}^{N}\left((\boldsymbol{I}-\boldsymbol{w}^{\tau_i(t)}(\boldsymbol{w}^{\tau_i(t)})^\top)\boldsymbol{X}_i\boldsymbol{X}_i^\top\boldsymbol{w}^{\tau_i(t)}-(\boldsymbol{I}-\boldsymbol{w}^t(\boldsymbol{w}^t)^\top)\boldsymbol{X}_i\boldsymbol{X}_i^\top\boldsymbol{w}^t\right)\right\|_2\\ &\le \frac{1}{n}\sum_{i=1}^{N}\left\|(\boldsymbol{I}-\boldsymbol{w}^{\tau_i(t)}(\boldsymbol{w}^{\tau_i(t)})^\top)\boldsymbol{X}_i\boldsymbol{X}_i^\top\boldsymbol{w}^{\tau_i(t)}-(\boldsymbol{I}-\boldsymbol{w}^t(\boldsymbol{w}^t)^\top)\boldsymbol{X}_i\boldsymbol{X}_i^\top\boldsymbol{w}^t\right\|_2\\ &\le \frac{4R}{n}\sum_{i=1}^{N}n_i\|\boldsymbol{w}^{\tau_i(t)}-\boldsymbol{w}^t\|_2, \qquad (28)\end{aligned}$$

where (28) follows from $\|\boldsymbol{X}_i\boldsymbol{X}_i^\top\|_2 \le \sum_{j=1}^{n_i}\|\boldsymbol{x}_i^j(\boldsymbol{x}_i^j)^\top\|_2 = n_i\|\boldsymbol{x}_i^j\|_2^2 \le n_iR$ by Assumption 1 and then using Lemma 5. Since $\boldsymbol{w}^{\tau_i(t)}-\boldsymbol{w}^t = \sum_{s=\tau_i(t)}^{t-1}(\boldsymbol{w}^{s+1}-\boldsymbol{w}^s)$, then repeatedly applying triangle inequality gives

$$\begin{aligned}\|\boldsymbol{e}^t\|_2 &\le \frac{4R}{n}\sum_{i=1}^{N}n_i\sum_{s=\tau_i(t)}^{t-1}\|\boldsymbol{w}^{s+1}-\boldsymbol{w}^s\|_2\\ &\le \frac{4R}{n}\sum_{i=1}^{N}n_i\sum_{s=(t-T)_+}^{t-1}\|\boldsymbol{w}^{s+1}-\boldsymbol{w}^s\|_2 \qquad (29)\\ &\le \frac{4R}{n}\sum_{i=1}^{N}n_i\sum_{s=(t-T)_+}^{t-1}4\eta\|\boldsymbol{g}^s\|_2 \qquad (30)\\ &\le 16R\eta\sum_{s=(t-T)_+}^{t-1}\|\boldsymbol{g}^s\|_2, \qquad (31)\end{aligned}$$

where (29) follows from Assumption 3, (30) is implied by $\eta < 1/(2R)$ and Lemma 6, and (31) is due to $\sum_{i=1}^{N}n_i = n$. To proceed, we plug $\boldsymbol{g}^t = \operatorname{grad}\mathcal{F}(\boldsymbol{w}^t)+\boldsymbol{e}^t$ into (31) and using (28), we have

$$\|\boldsymbol{e}^t\|_2 \le 16R\eta\sum_{s=(t-T)_+}^{t-1}(\|\operatorname{grad}\mathcal{F}(\boldsymbol{w}^s)\|_2+\|\boldsymbol{e}^s\|_2)$$

$$\leq 16R\eta \sum_{s=(t-T)_+}^{t-1} \left(\|\operatorname{grad} \mathcal{F}(\boldsymbol{w}^s)\|_2 + \frac{4R}{n} \sum_{i=1}^{N} n_i \|\boldsymbol{w}^{\tau_i(s)} - \boldsymbol{w}^s\|_2 \right). \tag{32}$$

Then, we need to upper bound the terms $\|\operatorname{grad} \mathcal{F}(\boldsymbol{w}^s)\|_2$ and $\|\boldsymbol{w}^{\tau_i(s)} - \boldsymbol{w}^t\|_2$ in (32). Since $(\boldsymbol{I} - \boldsymbol{v}_1\boldsymbol{v}_1^\top)\boldsymbol{A}\boldsymbol{v}_1 = (\boldsymbol{I} - \boldsymbol{v}_1\boldsymbol{v}_1^\top)\lambda_1\boldsymbol{v}_1 = \boldsymbol{0}$ and $\|\boldsymbol{A}\|_2 = \|\frac{1}{n}\sum_{i=1}^N \boldsymbol{X}_i\boldsymbol{X}_i^\top\|_2 \leq \frac{1}{n}\sum_{i=1}^N n_i R = R$ by Assumption 1, then it holds for all $s = (t-T)_+, \ldots, t-1$ that

$$\begin{aligned}
\|\operatorname{grad} \mathcal{F}(\boldsymbol{w}^s)\|_2 &= \left\|(\boldsymbol{I} - \boldsymbol{w}^s(\boldsymbol{w}^s)^\top)\boldsymbol{A}\boldsymbol{w}^s\right\|_2 \\
&= \left\|(\boldsymbol{I} - \boldsymbol{w}^s(\boldsymbol{w}^s)^\top)\boldsymbol{A}\boldsymbol{w}^s - (\boldsymbol{I} - \boldsymbol{v}_1\boldsymbol{v}_1^\top)\boldsymbol{A}\boldsymbol{v}_1\right\|_2 \\
&\leq 4R\|\boldsymbol{w}^s - \boldsymbol{v}_1\|_2 \qquad (33) \\
&= 4R\sqrt{2\left(1 - \langle \boldsymbol{w}^s, \boldsymbol{v}_1\rangle\right)} \\
&\leq 4\sqrt{2}R\sqrt{1 - \langle \boldsymbol{w}^s, \boldsymbol{v}_1\rangle^2} \qquad (34) \\
&= 4\sqrt{2}R\sqrt{\mathcal{E}_s}, \qquad (35)
\end{aligned}$$

where the (33) follows from Lemma 5 and (34) holds due to $\langle \boldsymbol{w}^t, \boldsymbol{v}_1\rangle \leq 1$. Moreover, it holds for all $s = (t-T)_+, \ldots, t-1$ that

$$\begin{aligned}
\|\boldsymbol{w}^{\tau_i(s)} - \boldsymbol{w}^s\|_2 &\leq \|\boldsymbol{w}^{\tau_i(s)} - \boldsymbol{v}_1\|_2 + \|\boldsymbol{w}^s - \boldsymbol{v}_1\|_2 \\
&= \sqrt{2\left(1 - \langle \boldsymbol{w}^{\tau_i(s)}, \boldsymbol{v}_1\rangle\right)} + \sqrt{2\left(1 - \langle \boldsymbol{w}^s, \boldsymbol{v}_1\rangle\right)} \\
&\leq \sqrt{2\left(1 - \langle \boldsymbol{w}^{\tau_i(s)}, \boldsymbol{v}_1\rangle^2\right)} + \sqrt{2\left(1 - \langle \boldsymbol{w}^s, \boldsymbol{v}_1\rangle^2\right)} \qquad (36) \\
&= \sqrt{2\mathcal{E}_{\tau_i(s)}} + \sqrt{2\mathcal{E}_s}, \qquad (37)
\end{aligned}$$

where the (36) is due to $\langle \boldsymbol{w}^{\tau_i(s)}, \boldsymbol{v}_1\rangle \leq 1$ and $\langle \boldsymbol{w}^s, \boldsymbol{v}_1\rangle \leq 1$.

Plugging (35) and (37) back into (32) gives

$$\begin{aligned}
\|\boldsymbol{e}^t\|_2 &\leq 16R\eta \sum_{s=(t-T)_+}^{t-1} \left(4\sqrt{2}R\sqrt{\mathcal{E}_s} + \frac{4\sqrt{2}R}{n} \sum_{i=1}^{N} n_i \left(\sqrt{\mathcal{E}_{\tau_i(s)}} + \sqrt{\mathcal{E}_s} \right) \right) \\
&\leq 64\sqrt{2}R^2\eta \left(\sum_{s=(t-T)_+}^{t-1} \sqrt{\mathcal{E}_s} + \sum_{s=(t-T)_+}^{t-1} \frac{1}{n} \sum_{i=1}^{N} n_i \left(\sqrt{\mathcal{E}_{\tau_i(s)}} + \sqrt{\mathcal{E}_s} \right) \right) \\
&\leq 64\sqrt{2}R^2\eta \sum_{s=(t-T)_+}^{t-1} \left(\sqrt{\mathcal{E}_s} + \frac{1}{n} \sum_{i=1}^{N} n_i \left(\max_{1\leq i\leq n} \sqrt{\mathcal{E}_{\tau_i(s)}} + \sqrt{\mathcal{E}_s} \right) \right) \\
&\leq 64\sqrt{2}R^2\eta \sum_{s=(t-T)_+}^{t-1} \left(\sqrt{\mathcal{E}_s} + \frac{2}{n} \sum_{i=1}^{N} n_i \max_{1\leq i\leq n} \sqrt{\mathcal{E}_{\tau_i(s)}} \right) \\
&\leq 64\sqrt{2}R^2\eta \sum_{s=(t-T)_+}^{t-1} \left(\sqrt{\mathcal{E}_s} + 2 \max_{(s-T)_+\leq j\leq s} \sqrt{\mathcal{E}_j} \right) \qquad (38) \\
&\leq 192\sqrt{2}R^2\eta \sum_{s=(t-T)_+}^{t-1} \max_{(s-T)_+\leq j\leq s} \sqrt{\mathcal{E}_j} \\
&\leq C_1 T\eta \max_{(t-2T)_+\leq j\leq t} \sqrt{\mathcal{E}_j}, \qquad (39)
\end{aligned}$$

where (38) is due to $\sum_{i=1}^N n_i = n$ and (39) holds by setting $C_1 := 192\sqrt{2}R^2$. □

A.4 Proof of Lemma 3

Proof. By (26), we have

$$\sum_{\ell=1}^{d}(a_\ell^t)^2 \geq (a_1^t)^2 = \left(1+\lambda_1\eta-\eta(\boldsymbol{w}^t)^\top \boldsymbol{A}\boldsymbol{w}^t\right)^2 \langle \boldsymbol{w}^t, \boldsymbol{v}_1\rangle^2 \geq \langle \boldsymbol{w}^t, \boldsymbol{v}_1\rangle^2 \geq \frac{1}{2}. \tag{40}$$

Since $\eta < 1/(2R)$, then using (18) in Lemma 2 and the fact that $\mathcal{E}_\tau \leq 1$ for all $\tau = 0, \ldots, t$, we have

$$\begin{aligned}
\sum_{\ell=1}^{d}\left(\zeta_\ell^t\right)^2 &= \sum_{\ell=1}^{d}\eta^2\left\langle \boldsymbol{e}^t, \boldsymbol{v}_\ell\right\rangle^2 \\
&= \eta^2(\boldsymbol{e}^t)^\top\left(\sum_{\ell=1}^{d}\boldsymbol{v}_\ell\boldsymbol{v}_\ell^\top\right)\boldsymbol{e}^t \\
&= \eta^2\|\boldsymbol{e}^t\|_2^2 \\
&\leq \eta^2\left(C_1T\eta\max_{(t-2T)_+\leq j\leq t}\sqrt{\mathcal{E}_j}\right)^2 \\
&\leq C_1^2T^2\eta^4\max_{(t-2T)_+\leq j\leq t}\mathcal{E}_j \\
&\leq C_1^2T^2\eta^4.
\end{aligned} \tag{41}$$

Then, combining (40) and (41) gives

$$\begin{aligned}
\frac{1}{1+\beta}-\frac{1}{\beta}\frac{\sum_{\ell=1}^{d}\left(\zeta_\ell^t\right)^2}{\sum_{\ell=1}^{d}(a_\ell^t)^2} &\geq \frac{1}{1+\beta}-\frac{1}{\beta}2C_1^2T^2\eta^4 \\
&\geq 1-\beta-\frac{2C_1^2T^2}{\beta}\eta^4,
\end{aligned}$$

where the second inequality is due to $1/(1+\beta)\geq 1-\beta$ for all $\beta>0$. This proves (19).

Then, it suffices to upper bound $\sum_{\ell=2}^{d}2a_\ell^t\zeta_\ell^t+\sum_{\ell=2}^{d}\left(\zeta_\ell^t\right)^2$ and lower bound $\sum_{\ell=1}^{d}\left(a_\ell^t+\zeta_\ell^t\right)^2$. By (26), we have

$$1-\mathcal{E}_t \leq (a_1^t)^2 = \left(1+\lambda_1\eta-\eta(\boldsymbol{w}^t)^\top\boldsymbol{A}\boldsymbol{w}^t\right)^2\langle\boldsymbol{w}^t,\boldsymbol{v}_1\rangle^2 \leq 1+\lambda_1\eta. \tag{42}$$

Besides, it follows from (18) in Lemma 2 that

$$|\zeta_1^t| = \left|\eta\left\langle\boldsymbol{e}^t,\boldsymbol{v}_1\right\rangle\right| \leq C_1T\eta^2\max_{(t-2T)_+\leq j\leq t}\sqrt{\mathcal{E}_j}. \tag{43}$$

Combining the second inequality in (42) and (43) gives

$$\begin{aligned}
\left|2a_1^t\zeta_1^t\right| &\leq 2\left(1+\lambda_1\eta\right)\langle\boldsymbol{w}^t,\boldsymbol{v}_1\rangle C_1T\eta^2\max_{(t-2T)_+\leq j\leq t}\sqrt{\mathcal{E}_j} \\
&= 2C_1\left(1+\lambda_1\eta\right)\eta^2\sqrt{1-\mathcal{E}_t}\max_{(t-2T)_+\leq j\leq t}\sqrt{\mathcal{E}_j}.
\end{aligned} \tag{44}$$

Combining the first inequality in (42) and (44), we have

$$\begin{aligned}
\sum_{\ell=1}^{d}\left(a_\ell^t+\zeta_\ell^t\right)^2 &\geq (a_1^t)^2+2a_1^t\zeta_1^t \\
&\geq 1-\mathcal{E}_t-2C_1\left(1+\lambda_1\eta\right)\eta^2\sqrt{1-\mathcal{E}_t}\max_{(t-2T)_+\leq j\leq t}\sqrt{\mathcal{E}_j} \\
&\geq 1-\mathcal{E}_t-3C_1\eta^2\sqrt{1-\mathcal{E}_j}\max_{(t-2T)_+\leq j\leq t}\sqrt{\mathcal{E}_{i_t}} \qquad (45) \\
&\geq 1-\frac{1}{2}-\frac{3C_1\eta^2}{2} \qquad (46)
\end{aligned}$$

$$\geq \frac{1}{4}, \tag{47}$$

where (45) follows from $\eta < 1/(2\lambda_1)$ implies that $1 + \lambda_1\eta \leq 3/2$, (46) follows from $0 \leq \mathcal{E}_0, \ldots, \mathcal{E}_t \leq 1/2$ by the induction hypothesis (7), and (47) is implied by $\eta < 1/\sqrt{6C_1}$. Moreover, we have

$$\sum_{\ell=2}^{d} \left(\zeta_\ell^t\right)^2 \leq \sum_{\ell=1}^{d} \left(\zeta_\ell^t\right)^2 = \sum_{\ell=1}^{d} \eta^2 \left\langle \boldsymbol{e}^t, \boldsymbol{v}_\ell \right\rangle^2 = \eta^2 (\boldsymbol{e}^t)^\top \boldsymbol{V}\boldsymbol{V}^\top \boldsymbol{e}^t = \eta^2 \|\boldsymbol{e}^t\|_2^2. \tag{48}$$

Combining (47) and (48), we obtain

$$\frac{\sum_{\ell=2}^{d} \left(\zeta_\ell^t\right)^2}{\sum_{\ell=1}^{d} \left(a_\ell^t + \zeta_\ell^t\right)^2} \leq 4\eta^2 \|\boldsymbol{e}^t\|_2^2,$$

which proves (20). □

A.5 Proof of Lemma 4

Proof. Since $\eta < \min\{1/(2\lambda_1), 1/\sqrt{6C_1}\}$, then incorporating (15) in Lemma 1, and (19) and (20) in Lemma 3 into (14), we obtain

$$\sqrt{\mathcal{E}_{t+1}} \leq \sqrt{\frac{1 - \frac{\Delta}{6}\eta}{1 - \beta - \frac{2C_1^2T^2}{\beta}\eta^4}}\sqrt{\mathcal{E}_t} + 2\eta\|\boldsymbol{e}^t\|_2 \tag{49}$$

for all $\beta > 0$. Taking $\beta = \alpha\eta$ for some $\alpha > 0$ gives

$$\sqrt{\mathcal{E}_{t+1}} \leq \sqrt{\frac{1 - \frac{\Delta}{6}\eta}{1 - \alpha\eta - \frac{2C_1^2T^2}{\alpha}\eta^3}}\sqrt{\mathcal{E}_t} + 2\eta\|\boldsymbol{e}^t\|_2. \tag{50}$$

Besides, it holds for all $\alpha < \frac{\lambda_1 - \lambda_2}{12}$ that

$$\begin{aligned} &\eta \leq h(\alpha) := \sqrt{\frac{\left(\frac{\Delta}{12} - \alpha\right)\alpha}{2C_1^2T^2}} \\ \iff &1 - \frac{\Delta}{6}\eta \leq 1 - \left(\frac{\Delta}{12} + \alpha\right)\eta - \frac{2C_1^2T^2}{\alpha}\eta^3 \\ \implies &1 - \frac{\Delta}{6}\eta \leq \left(1 - \alpha\eta - \frac{2C_1^2T^2}{\alpha}\eta^3\right)\left(1 - \frac{\Delta}{12}\eta\right). \end{aligned} \tag{51}$$

It is easy to verify that

$$\max_{\alpha > 0} h(\alpha) = h\left(\frac{\Delta}{24}\right) = \frac{\Delta}{24\sqrt{2}C_1}.$$

Since $\eta \leq \frac{\Delta}{24\sqrt{2}C_1}$, then taking $\alpha = \frac{\Delta}{24}$, it follows from (51) that

$$\frac{1 - \frac{\Delta}{6}\eta}{1 - \alpha\eta - \frac{2C_1^2T^2}{\alpha}\eta^3} \leq 1 - \frac{\Delta}{12}\eta. \tag{52}$$

Hence, plugging (52) back into (49) and using the fact that $\sqrt{1-x} \leq 1 - x/2$ for all $x \leq 1$ give

$$\begin{aligned} \sqrt{\mathcal{E}_{t+1}} &\leq \sqrt{1 - \frac{\Delta}{12}\eta} \cdot \sqrt{\mathcal{E}_t} + 2\eta\|\boldsymbol{e}^t\|_2 \\ &\leq \left(1 - \frac{\Delta}{24}\eta\right)\sqrt{\mathcal{E}_t} + 2\eta\|\boldsymbol{e}^t\|_2 \end{aligned}$$

$$= \left(1 - \frac{\Delta}{48}\eta\right)\sqrt{\mathcal{E}_t} - \frac{\Delta}{48}\eta\sqrt{\mathcal{E}_t} + 2\eta\|\boldsymbol{e}^t\|_2. \tag{53}$$

It follows from (35) and (10) that

$$\sqrt{\mathcal{E}_t} \geq \frac{1}{4\sqrt{2}R}\|\operatorname{grad}\mathcal{F}(\boldsymbol{w}^t)\|_2 = \frac{1}{4\sqrt{2}R}\|\boldsymbol{g}^t - \boldsymbol{e}^t\|_2 \geq \frac{1}{4\sqrt{2}R}\left(\|\boldsymbol{g}^t\|_2 - \|\boldsymbol{e}^t\|_2\right).$$

Plugging this back into (53) and using inequality (31) give

$$\begin{aligned}
\sqrt{\mathcal{E}_{t+1}} &\leq \left(1 - \frac{\Delta}{48}\eta\right)\sqrt{\mathcal{E}_t} - \frac{\Delta}{48}\eta\frac{1}{4\sqrt{2}R}\left(\|\boldsymbol{g}^t\|_2 - \|\boldsymbol{e}^t\|_2\right) + 2\eta\|\boldsymbol{e}^t\|_2 \\
&= \left(1 - \frac{\Delta}{48}\eta\right)\sqrt{\mathcal{E}_t} - \frac{\Delta}{192\sqrt{2}R}\eta\|\boldsymbol{g}^t\|_2 + \left(2 + \frac{\Delta}{192\sqrt{2}R}\right)\eta\|\boldsymbol{e}^t\|_2 \\
&\leq \left(1 - \frac{\Delta}{48}\eta\right)\sqrt{\mathcal{E}_t} - \frac{\Delta}{192\sqrt{2}R}\eta\|\boldsymbol{g}^t\|_2 + \left(32R + \frac{\Delta}{12\sqrt{2}}\right)\eta^2\sum_{s=(t-T)_+}^{t}\|\boldsymbol{g}^s\|_2,
\end{aligned} \tag{54}$$

as desired. □

A.6 Proof of Theorem 1

To prove Theorem 1, we use the following result provided in Aytekin et al. (2016) to solve the recurrence (21) of the sequence $\{\mathcal{E}_t\}_{t\in\mathbb{N}}$ in Lemma 4.

Lemma 7. *Let $\{V_t\}_{t\in\mathbb{N}}$ and $\{W_t\}_{t\in\mathbb{N}}$ be sequences of non-negative real numbers satisfying*

$$V_{t+1} \leq aV_t - bW_t + c\sum_{s=(t-T_0)_+}^{t} W_s,\ t \in \mathbb{N},$$

for some real numbers $a \in (0,1)$ and $b, c \geq 0$ and some integer $T_0 \geq 0$. Assume that the following holds:

$$\frac{c}{1-a}\frac{1-a^{T_0+1}}{a^{T_0}} \leq b.$$

Then, $V_t \leq a^t V_0$ for all $t \in \mathbb{N}$.

To apply Lemma 7, we let $a = 1 - \frac{\Delta}{48}\eta$, $b = \frac{\Delta}{192\sqrt{2}R}\eta$, and $c = \left(32R + \frac{\Delta}{12\sqrt{2}}\right)\eta^2$. Then, we have

$$\begin{aligned}
&\frac{c}{1-a}\frac{1-a^{T+1}}{a^T} \leq b \qquad (55)\\
\iff &\frac{\left(32R + \frac{\Delta}{12\sqrt{2}}\right)\eta^2}{\frac{\Delta}{48}\eta}\frac{1-\left(1-\frac{\Delta}{48}\eta\right)^{T+1}}{\left(1-\frac{\Delta}{48}\eta\right)^{T}} \leq \frac{\Delta}{192\sqrt{2}R}\eta \\
\iff &\frac{\frac{1}{\left(1-\frac{\Delta}{48}\eta\right)^{T+1}} - 1}{\frac{1}{1-\frac{\Delta}{48}\eta}} \leq \frac{\Delta^2}{9216\sqrt{2}R}\frac{1}{32R + \frac{\Delta}{12\sqrt{2}}} \\
\iff &\frac{1}{\left(1-\frac{\Delta}{48}\eta\right)^{T+1}} - 1 \leq \frac{1}{1-\frac{\Delta}{48}\eta}\frac{\Delta^2}{C_2}, \qquad (56)
\end{aligned}$$

where $C_2 = 9216\sqrt{2}R\left(32R + \frac{\Delta}{12\sqrt{2}}\right)$. Since $(1+x)^\alpha \leq e^{\alpha x} \leq 1 + 2\alpha x$ for $\alpha \in \mathbb{R}$ and $x \leq 1/\alpha$, then we have

$$\begin{aligned}
\frac{1}{\left(1-\frac{\Delta}{48}\eta\right)^{T+1}} - 1 &= \left(1 + \frac{\frac{\Delta}{48}\eta}{1-\frac{\Delta}{48}\eta}\right)^{T+1} - 1 \\
&\leq 2(T+1)\frac{\frac{\Delta}{48}\eta}{1-\frac{\Delta}{48}\eta}.
\end{aligned}$$

Hence, to let (56) hold, it suffices to let

$$\begin{aligned} & 2(T+1)\frac{\frac{\Delta}{48}\eta}{1-\frac{\Delta}{48}\eta} \leq \frac{1}{1-\frac{\Delta}{48}\eta}\frac{\Delta^2}{C_2} \\ \Longleftrightarrow\ & \eta \leq \frac{24\Delta}{C_2(T+1)}. \end{aligned}$$

Since $\sqrt{6C_1} > 2R$, we have

$$\begin{aligned} \eta &\leq \min\left\{\frac{1}{2\lambda_1}, \frac{1}{\sqrt{6C_1}}, \frac{\Delta}{24\sqrt{2}C_1}, \frac{24\Delta}{C_2(T+1)}, \frac{47}{\Delta}\right\} \\ &= \min\left\{\frac{1}{2R}, \frac{1}{2\lambda_1}, \frac{1}{\sqrt{6C_1}}, \frac{\Delta}{24\sqrt{2}C_1}, \frac{24\Delta}{C_2(T+1)}, \frac{47}{\Delta}\right\}. \end{aligned}$$

Thus, the recurrence (21) and condition (55) hold. Then, it follows from Lemma 7 that

$$\sqrt{\mathcal{E}_{t+1}} \leq \left(1-\frac{\Delta}{48}\eta\right)^{t+1}\sqrt{\mathcal{E}_0} \leq \sqrt{\mathcal{E}_0} \leq \frac{1}{\sqrt{2}},$$

which further implies that

$$\langle \boldsymbol{w}^{t+1}, \boldsymbol{v}_1\rangle = \sqrt{1-\mathcal{E}_{t+1}} \geq \sqrt{1-\frac{1}{2}} = \frac{\sqrt{2}}{2}.$$

This completes the proof of the induction step (8), and thus Theorem 1.