Decentralized Learning: Theoretical Optimality and Practical Improvements

Yucheng Lu   YL2967@CORNELL.EDU
Department of Computer Science
Cornell University
Ithaca, NY 14850, USA

Christopher De Sa   CDESA@CS.CORNELL.EDU
Department of Computer Science
Cornell University
Ithaca, NY 14850, USA

Abstract

Decentralization is a promising method of scaling up parallel machine learning systems. In this paper, we provide a tight lower bound on the iteration complexity for such methods in a stochastic non-convex setting. Our lower bound reveals a theoretical gap in known convergence rates of many existing decentralized training algorithms, such as D-PSGD. We prove by construction this lower bound is tight and achievable. Motivated by our insights, we further propose DeTAG, a practical gossip-style decentralized algorithm that achieves the lower bound with only a logarithm gap. While a simple version of DeTAG with plain SGD and constant step size suffice for achieving theoretical limits, we additionally provide convergence bound for DeTAG under general non-increasing step size and momentum. Empirically, we compare DeTAG with other decentralized algorithms on multiple vision benchmarks, including CIFAR10/100 and ImageNet. We substantiate our theory and show DeTAG converges faster on unshuffled data and in sparse networks. Furthermore, we study a DeTAG variant, DeTAG*, that practically speeds up data-center-scale model training. This manuscript is the extended version for (Lu and De Sa, 2021).

Keywords: decentralization, parallel learning, non-convex optimization, lower bound, stochastic optimization

1. Introduction

Parallelism is a ubiquitous method to accelerate model training (Abadi et al., 2016; Alistarh, 2018; Alistarh et al., 2020; Lu et al., 2020). A parallel learning system usually consists of three layers (Table 1): an application to solve, a communication protocol deciding how parallel workers coordinate, and a network topology determining how workers are connected. Traditional design for these layers usually follows a centralized setup: in the application layer, training data is required to be shuffled and shared among parallel workers; while in the protocol and network layers, workers either communicate via a fault-tolerant single central node (e.g. Parameter Server) (Li et al., 2014a,b; Ho et al., 2013) or a fully-connected topology (e.g. AllReduce) (Gropp et al., 1999; Patarasuk and Yuan, 2009). This centralized design limits the scalability of learning systems in two aspects. First, in

Table 1: Design choice of centralization and decentralization in different layers of a parallel machine learning system. The protocol specifies how workers communicate. The topology refers to the overlay network that logically connects all the workers.

Layer	Centralized	Decentralized
Application	Shuffled (Homogeneous) Data (e.g. Data Center Training)	Unshuffled (Heterogeneous) Data (e.g. Federated Learning)
Protocol	AllReduce/AllGather Parameter Server	Gossip
Network Topology	Complete-(Bipartite) Graph	Arbitrary Graph

many scenarios, such as Federated Learning (Koloskova et al., 2019a; McMahan et al., 2016) and Internet of Things (IOT) (Kanawaday and Sane, 2017), a shuffled dataset or a complete (bipartite) communication graph is not possible or affordable to obtain. Second, a centralized communication protocol can significantly slow down the training, especially with a low-bandwidth or high-latency network (Lian et al., 2017b; Tang et al., 2019b; Yu et al., 2018).

The rise of decentralization.   To mitigate these limitations, decentralization comes to the rescue. Decentralizing the application and network allows workers to learn with unshuffled local datasets (Li et al., 2019) and arbitrary topologies (Seaman et al., 2017; Shanthamallu et al., 2017). Furthermore, the decentralized protocol, i.e. Gossip, helps to balance load, and has been shown to outperform centralized protocols in many cases (Lian et al., 2017a; Yu et al., 2019; Nazari et al., 2019; Lu and De Sa, 2020).

Understanding decentralization with layers.   Many decentralized training designs have been proposed, which can lead to confusion as the term "decentralization" is used inconsistently in the literature. Some works use "decentralized" to refer to approaches that can tolerate non-iid or unshuffled datasets (Li et al., 2019), while others use it to mean gossip communication (Lian et al., 2017a), and still others use it to mean a sparse topology graph (Wan et al., 2020). To eliminate this ambiguity, we formulate Table 1, which summarizes the different "ways" a system can be decentralized. Note that the choices to decentralize different layers are *independent*, e.g., the centralized protocol AllReduce can still be implemented on a decentralized topology like the Ring graph (Wan et al., 2020).

The theoretical limits of decentralization.   Despite the empirical success, the best convergence rates achievable by decentralized training—and how they interact with different notions of decentralization—remains an open question. Previous works often show complexity of a given decentralized algorithm with respect to the number of iterations T or the number of workers n, ignoring other factors including network topologies, function parameters or data distribution. Although a series of decentralized algorithms have been proposed showing theoretical improvements—such as using variance reduction (Tang et al., 2018b),
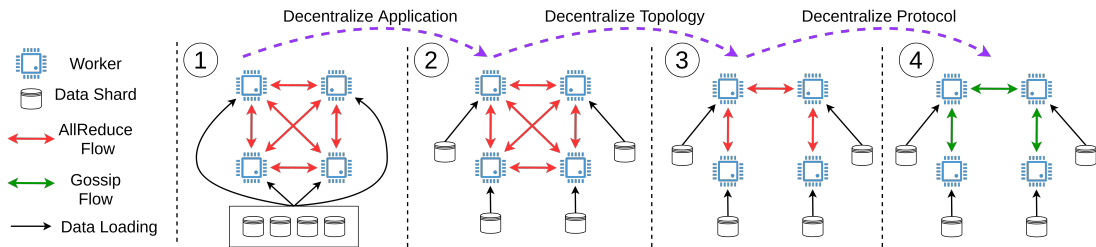
Figure 1: Figure illustrating how decentralization in different layers lead to different learning systems. From left to right: ①: A fully centralized system where workers sample from shared and shuffled data; ②: Based on ①, workers maintain their own data sources, making it decentralized in the application layer; ③: Based on ②, workers are decentralized in the topology layer; ④: A fully decentralized system in all three layers where the workers communicate via Gossip. Our framework and theory are applicable to all kinds of decentralized learning systems.

acceleration (Seaman et al., 2017), or matching (Wang et al., 2019)—we do not know how close they are to an "optimal" rate or whether further improvement is possible.

In light of this, a natural question is: *What is the optimal complexity in decentralized training? Has it been achieved by any algorithm yet?* Previous works have made initial attempts on this question, by analyzing this theoretical limit in a non-stochastic or (strongly) convex setting (Seaman et al., 2017; Scaman et al., 2018; Koloskova et al., 2020; Woodworth et al., 2018; Dvinskikh and Gasnikov, 2019; Sun and Hong, 2019). These results provide great heuristics but still leave the central question open, since stochastic methods are usually used in practice and many real-world problems of interest are non-convex (e.g. deep learning). In this paper we give the first full answer to this question: our contributions are as follows.

- In Section 4, we prove the first (to our knowledge) tight lower bound for decentralized training in a stochastic non-convex setting. Our results reveal an asymptotic gap between our lower bound and known convergence rates of existing algorithms.

- In Section 5, we prove our lower bound is tight by exhibiting an algorithm called DeFacto that achieves it—albeit while only being decentralized in the sense of the application and network layers.

- In Section 6, we propose DeTAG, a practical algorithm that achieves the lower bound with only a logarithm gap and is decentralized in all three layers. While we show plain SGD and constant step size suffice for optimality, we provide the convergence guarantee for DeTAG under general non-increasing step size and momentum.

- In Section 7, we experimentally evaluate DeTAG on the CIFAR benchmark and show it converges faster compared to decentralized learning baselines. We additionally propose DeTAG*, a practical version that allows speed up on data-center-scale model training.

2. Related Work

Decentralized Training. In the application layer, decentralized training usually denotes federated learning (Zhao et al., 2018). Research on decentralization in this sense investigates convergence where each worker samples only from a local dataset which is not independent and identically distributed to other workers' datasets (Bonawitz et al., 2019; Tran et al., 2019; Yang et al., 2019; Konečnỳ et al., 2016). Another line of research on decentralization focuses on the protocol layer—with average gossip (Boyd et al., 2005, 2006), workers communicate by averaging their parameters with neighbors on a graph. D-PSGD (Lian et al., 2017a) is one of the most basic algorithms that scales SGD with this protocol, achieving a linear parallel speed up. Additional works extend D-PSGD to asynchronous and variance-reduced cases (Lian et al., 2017b; Tang et al., 2018b; Tian et al., 2020; Zhang and You, 2019b; Hendrikx et al., 2019; Xin et al., 2021a). After those, Zhang and You (2019c); Xin et al. (2019, 2021b) propose adding gradient trackers to D-PSGD. Other works discuss the application of decentralization on specific tasks such as linear models or deep learning (He et al., 2018; Assran et al., 2018). Zhang and You (2019a) treats the case where only directed communication can be performed. Wang et al. (2019) proposes using matching algorithms to optimize the gossip protocol. Multiple works discuss using compression to decrease communication costs in decentralized training (Koloskova et al., 2019b,a; Lu and De Sa, 2020; Tang et al., 2019a, 2018a), and other papers connect decentralized training to other parallel methods and present a unified theory (Lu et al., 2020; Koloskova et al., 2020; Wang and Joshi, 2018). In some even earlier works like (Nedic and Ozdaglar, 2009; Duchi et al., 2010), full local gradients on a convex setting is investigated.

Lower Bounds in Stochastic Optimization. Lower bounds are a well studied topic in non-stochastic optimization, especially in convex optimization (Agarwal and Bottou, 2014; Arjevani and Shamir, 2015; Lan and Zhou, 2018; Fang et al., 2018; Arjevani and Shamir, 2017). In the stochastic setting, Allen-Zhu (2018) and Foster et al. (2019) discuss the complexity lower bound to find stationary points on convex problems. Other works study the lower bound in a convex, data-parallel setting (Diakonikolas and Guzmán, 2018; Balkanski and Singer, 2018; Tran-Dinh et al., 2019), and Colin et al. (2019) extends the result to a model-parallel setting. In the domain of non-convex optimization, Carmon et al. (2017, 2019) propose a zero-chain model that obtains tight bound for a first order method to obtain stationary points. Zhou and Gu (2019) extends this lower bound to a finite sum setting, and Arjevani et al. (2019) proposes a probabilistic zero-chain model that obtains tight lower bounds for first-order methods on stochastic and non-convex problems.

3. Setting

In this section, we introduce the notation and assumptions we will use. Throughout the paper, we consider the standard *data-parallel* training setup with n parallel workers. Each worker i stores a copy of the model $\boldsymbol{x} \in \mathbb{R}^d$ and a local dataset $\mathcal{D}_i$. The model copy and local dataset define a local loss function (or empirical risk) f_i. The ultimate goal of the parallel workers is to output a target model $\hat{\boldsymbol{x}}$ that minimizes the average over all the local

Table 2: Complexity comparison among different algorithms in the stochastic non-convex setting on arbitrary graphs. The blue text are the results from this paper. Definitions to all the parameters can be found in Section 3. Other algorithms like EXTRA (Shi et al., 2015) or MSDA (Scaman et al., 2017) are not comparable since they are designed for (strongly) convex problems. Additionally, Liu and Zhang (2021) provides alternative complexity bound for algorithms like D-PSGD which improves upon the spectral gap. However, the new bound would compromise the dependency on ϵ, which does not conflict with our comparison here. In the table, L.B. stands for lower bound while U.B. stands for upper bound.

	Source	Protocol	Term with σ	Term with D, λ	Gap to L.B.
L.B.	Theorem 1	Central	$\Omega\left(\frac{\Delta L\sigma^2}{nB\epsilon^4}\right)$	$\Omega\left(\frac{\Delta LD}{\epsilon^2}\right)$	/
	Corollary 2	Decentral	$\Omega\left(\frac{\Delta L\sigma^2}{nB\epsilon^4}\right)$	$\Omega\left(\frac{\Delta L}{\epsilon^2\sqrt{1-\lambda}}\right)$	/
U.B.	DeFacto (Theorem 4)	Central	$O\left(\frac{\Delta L\sigma^2}{nB\epsilon^4}\right)$	$O\left(\frac{\Delta LD}{\epsilon^2}\right)$	$O(1)$
	DeTAG (Theorem 5)	Decentral	$O\left(\frac{\Delta L\sigma^2}{nB\epsilon^4}\right)$	$O\left(\frac{\Delta L\log\left(\frac{\varsigma_0 n}{\epsilon\sqrt{\Delta L}}\right)}{\epsilon^2\sqrt{1-\lambda}}\right)$	$O\left(\log\left(\frac{\varsigma_0 n}{\epsilon\sqrt{\Delta L}}\right)\right)$
	D-PSGD (Lian et al., 2017a)	Decentral	$O\left(\frac{\Delta L\sigma^2}{nB\epsilon^4}\right)$	$O\left(\frac{\Delta Ln\varsigma}{\epsilon^2(1-\lambda)^2}\right)$	$O\left(\frac{n\varsigma}{(1-\lambda)^{\frac{3}{2}}}\right)$
	SGP (Assran et al., 2019)	Decentral	$O\left(\frac{\Delta L\sigma^2}{nB\epsilon^4}\right)$	$O\left(\frac{\Delta Ln\varsigma}{\epsilon^2(1-\lambda)^2}\right)$	$O\left(\frac{n\varsigma}{(1-\lambda)^{\frac{3}{2}}}\right)$
	D^2 (Tang et al., 2018b)	Decentral	$O\left(\frac{\Delta L\sigma^2}{nB\epsilon^4}\right)$	$O\left(\frac{\lambda^2\Delta Ln\varsigma_0}{\epsilon^2(1-\lambda)^3}\right)$	$O\left(\frac{\lambda^2 n\varsigma_0}{(1-\lambda)^{\frac{5}{2}}}\right)$
	DSGT (Zhang and You, 2019c)	Decentral	$O\left(\frac{\Delta L\sigma^2}{nB\epsilon^4}\right)$	$O\left(\frac{\lambda^2\Delta Ln\varsigma_0}{\epsilon^2(1-\lambda)^3}\right)$	$O\left(\frac{\lambda^2 n\varsigma_0}{(1-\lambda)^{\frac{5}{2}}}\right)$
	GT-DSGD (Xin et al., 2021b)	Decentral	$O\left(\frac{\Delta L\sigma^2}{nB\epsilon^4}\right)$	$O\left(\frac{\lambda^2\Delta Ln\varsigma_0}{\epsilon^2(1-\lambda)^3}\right)$	$O\left(\frac{\lambda^2 n\varsigma_0}{(1-\lambda)^{\frac{5}{2}}}\right)$

loss functions, that is,

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d}\left[f(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}\underbrace{\mathbb{E}_{\xi_i\sim\mathcal{D}_i}f_i(\boldsymbol{x};\xi_i)}_{f_i(\boldsymbol{x})}\right]. \tag{1}$$

Here, ξ_i is a data sample from $\mathcal{D}_i$ and is used to compute a stochastic gradient via some oracle, e.g. back-propagation on a mini-batch of samples. The loss functions can (potentially) be non-convex so finding a global minimum is NP-Hard; instead, we expect the workers to output a point $\hat{\boldsymbol{x}}$ at which $f(\hat{\boldsymbol{x}})$ has a small gradient magnitude in expectation: $\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}})\| \leq \epsilon$, for

some small ϵ.[1] The assumptions our theoretical analysis requires can be categorized by the layers from Table 1: in each layer, "being decentralized" corresponds to certain assumptions (or lack of assumptions). We now describe these assumptions for each layer separately.

3.1 Application Layer

Application-layer assumptions comprise constraints on the losses f_i from (1) and the gradient oracle via which they are accessed by the learning algorithm, as these are constraints on the learning task itself.

Function class (Δ and L). As is usual in this space, we assume the local loss functions $f_i : \mathbb{R}^d \to \mathbb{R}$ are L-smooth,

$$\|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|, \ \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, \tag{2}$$

for some constant $L > 0$, and that the total loss f is range-bounded by Δ in the sense that $f(\boldsymbol{0}) - \inf_{\boldsymbol{x}} f(\boldsymbol{x}) \leq \Delta$. We let the function class $\mathcal{F}_{\Delta,L}$ denote the set of all functions that satisfy these conditions (for any dimension $d \in \mathbb{N}^+$).

Oracle class (σ^2). We assume each worker interacts with its local function f_i only via a stochastic gradient oracle $\tilde{g}_i$, and that when we query this oracle with model $\boldsymbol{x}$, it returns an independent unbiased estimator to $\nabla f_i(\boldsymbol{x})$ based on some random variable z with distribution Z (e.g. the index of a mini-batch randomly chosen for backprop). Formally,

$$\mathbb{E}_{z \sim Z}[\tilde{g}_i(\boldsymbol{x}, z)] = \nabla f_i(\boldsymbol{x}), \ \forall \boldsymbol{x} \in \mathbb{R}^d. \tag{3}$$

As per the usual setup, we additionally assume the local estimator has bounded variance: for some constant $\sigma > 0$,

$$\mathbb{E}_{z \sim Z}\|\tilde{g}_i(\boldsymbol{x}, z) - \nabla f_i(\boldsymbol{x})\|^2 \leq \sigma^2, \ \forall \boldsymbol{x} \in \mathbb{R}^d. \tag{4}$$

We let O denote a set of these oracles $\{\tilde{g}_i\}_{i \in [n]}$, and let the oracle class $\mathcal{O}_{\sigma^2}$ denote the class of all such oracle sets that satisfy these two assumptions.

Data shuffling (ς^2 and ς_0^2). At this point, an analysis with a centralized application layer would make the additional assumption that all the f_i are equal and the $\tilde{g}_i$ are identically distributed: this roughly corresponds to the assumption that the data all comes independently from a single centralized source. *We do not make this assumption*, and lacking such an assumption is what makes an analysis decentralized in the application layer. Still, some assumption that bounds the f_i relative to each other somehow is needed: we now discuss two such assumptions used in the literature, from which we use the weaker (and more decentralized) one.

One commonly made assumption (Lian et al., 2017a; Koloskova et al., 2019b,a; Lu and De Sa, 2020; Tang et al., 2018a) in decentralized training is

$$\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2 \leq \varsigma^2, \ \forall \boldsymbol{x} \in \mathbb{R}^d, \tag{5}$$

---

1. There are many valid stopping criteria. We adopt ϵ-stationary point as the success signal. $\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}})\|^2 \leq \epsilon^2$ is another commonly used criterion; we adopt the non-squared one following (Carmon et al., 2019). Other criterions regarding stationary points can be converted to hold in our theory.

for some constant ς, which is said to bound the "outer variance" among workers. This is often unreasonable, as it suggests the local datasets on workers must have close distribution: in practice, ensuring this often requires some sort of shuffling or common centralized data source. We **do not assume** (5) but instead adopt the much weaker assumption

$$\frac{1}{n} \sum_{i=1}^{n} \|\nabla f_i(\mathbf{0}) - \nabla f(\mathbf{0})\|^2 \leq \varsigma_0^2, \tag{6}$$

for constant $\varsigma_0 > 0$.[2] This assumption only requires a bound at point $\mathbf{0}$, which is, to the best of our knowledge, the weakest assumption of this type used in the literature (Tang et al., 2018b; Zhang and You, 2019c). Requiring such a weak assumption allows workers to (potentially) sample from different distributions or vary largely in their loss functions (e.g. in a federated learning environment).

3.2 Protocol Layer

Protocol-layer assumptions comprise constraints on the parallel learning algorithm itself, and especially on the way that the several workers communicate to approach consensus.

Algorithm class (B). We consider algorithms A that divide training into multiple iterations, and between two adjacent iterations, there must be a synchronization process among workers (e.g. a barrier) such that they start each iteration simultaneously.[3] Each worker running A has a local copy of the model, and we let $\boldsymbol{x}_{t,i} \in \mathbb{R}^d$ denote this model on worker i at iteration t. We assume without loss of generality that A initializes each local model at zero: $\boldsymbol{x}_{0,i} = \mathbf{0}$ for all i. At each iteration, each worker makes *at most B* queries to its gradient oracle $\tilde{g}_i$, for some constant $B \in \mathbb{N}^+$, and then uses the resulting gradients to update its model. We do not make any explicit rules for output and allow the output of the algorithm $\hat{\boldsymbol{x}}_t$ at the end of iteration t (the model that A would output if it were stopped at iteration t) to be any linear combination of all the local models, i.e.

$$\hat{\boldsymbol{x}}_t \in \operatorname{span}(\{\boldsymbol{x}_{t,j}\}_{j\in[n]}) = \{\textstyle\sum_{j=1}^{n} c_j \boldsymbol{x}_{t,j} \mid c_j \in \mathbb{R}\}. \tag{7}$$

Beyond these basic properties, we further require A to satisfy the following "zero-respecting" property from Carmon et al. (2017). Specifically, if $\boldsymbol{z}$ is any vector worker i queries its gradient oracle with at iteration t, then for any $k \in [d]$, if $\boldsymbol{e}_k^\top \boldsymbol{z} \neq 0$, then there exists a $s \leq t$ and a $j \in [n]$ such that either $j = i$ or j is a neighbor of i in the network connectivity graph G (i.e. $(i,j) \in \{(i,i)\} \cup G$) and $(\boldsymbol{e}_k^\top \boldsymbol{x}_{s,j}) \neq 0$. More informally, the worker will not query its gradient oracle with a nonzero value for some weight unless that weight was already nonzero in the model state of the worker or one of its neighbors at some point in the past. Similarly, for any $k \in [d]$, if $(\boldsymbol{e}_k^\top \boldsymbol{x}_{t+1,i}) \neq 0$, then either there exists an $s \leq t$ and j such that $(i,j) \in \{(i,i)\} \cup G$ and $(\boldsymbol{e}_k^\top \boldsymbol{x}_{s,j}) \neq 0$, or one of the gradient oracle's outputs $\boldsymbol{v}$ on worker i at iteration t has $\boldsymbol{e}_k^\top \boldsymbol{v} \neq 0$. Informally, a worker's model will not have a nonzero weight unless either (1) that weight was nonzero on that worker or one of its neighbors at a previous

---

2. As we only use ς_0 for upper bounds, not lower bounds, we do not define a "class" that depends on this parameter.

3. We consider synchronous algorithms only here for simplicity of presentation; further discussion of extension to asynchronous algorithms is included in the supplementary material.

iteration, or (2) the corresponding entry in one of the gradients the worker sampled at that iteration was nonzero.

Intuitively, we are requiring that algorithm A will not modify those coordinates that remain zero in all previous oracle outputs and neighboring models.[4] This lets A use a wide space of accessible information in communication and allows our class to cover first-order methods including SGD (Ghadimi and Lan, 2013), Momentum SGD (Nesterov, 1983), Adam (Kingma and Ba, 2014), RMSProp (Tieleman and Hinton, 2012), Adagrad (Ward et al., 2018), and AdaDelta (Zeiler, 2012). We let algorithm class $\mathcal{A}_B$ denote the set of all algorithms A that satisfy these assumptions.

So far our assumptions in this layer cover both centralized and decentralized protocols. Decentralized protocols, however, must satisfy the additional assumption that they communicate via *gossip* (see Section 2) (Boyd et al., 2005, 2006). A single step of gossip protocol can be expressed as

$$\boldsymbol{z}_{t,i} \leftarrow \sum_{j \in \mathcal{N}_i} \boldsymbol{y}_{t,j} \boldsymbol{W}_{ji}, \ \forall i \in [n] \tag{8}$$

for some constant doubly stochastic matrix $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ called the *communication matrix* and $\boldsymbol{y}$ and $\boldsymbol{z}$ are the input and output of the gossip communication step, respectively. The essence of a single Gossip step is to take weighted average over the neighborhood specified by a fixed matrix. To simplify later discussion, we further define the gossip matrix class $\mathcal{W}_n$ as the set of all matrices $\boldsymbol{W} \in \mathbb{R}^{n \times n}$, where $\boldsymbol{W}$ is doubly stochastic and $\boldsymbol{W}_{ij} \neq 0$ only if $(i, j) \in G$. We call every $\boldsymbol{W} \in \mathcal{W}_n$ a gossip matrix and we use $\lambda = \max\{|\lambda_2|, |\lambda_n|\} \in [0, 1)$ to denote its general second-largest eigenvalue, where λ_i denotes the i-th largest eigenvalue of $\boldsymbol{W}$. We let gossip algorithm class $\mathcal{A}_{B, \boldsymbol{W}}$ denote the set of all algorithms $A \in \mathcal{A}_B$ that only communicate via gossip using a single matrix $\boldsymbol{W} \in \mathcal{W}_n$. It trivially holds that $\mathcal{A}_{B, \boldsymbol{W}} \subset \mathcal{A}_B$.

3.3 Topology Layer

Topology-layer assumptions comprise constraints on how workers are connected topologically. We let the graph class $\mathcal{G}_{n,D}$ denote the class of graphs G connecting n workers (vertices) with diameter D, where diameter of a graph measures the maximum distance between two arbitrary vertices (so $1 \leq D \leq n - 1$). A centralized analysis here typically will also require that G be either complete or complete-bipartite (with parameter servers and workers as the two parts): lacking this requirement and allowing arbitrary graphs is what makes an analysis decentralized in the topology layer.

3.4 Complexity Measures

Now that we have defined the classes we are interested in, we can use them to define the complexity measures we will bound in our theoretical results. Given a loss function $f \in \mathcal{F}_{\Delta,L}$, a set of underlying oracles $O \in \mathcal{O}_{\sigma^2}$, a graph $G \in \mathcal{G}_{n,D}$, and an algorithm $A \in \mathcal{A}_B$, let $\hat{\boldsymbol{x}}_t^{A,f,O,G}$ denote the output of algorithm A at the end of iteration t under this setting. Then

---

4. On the other hand, it is possible to even drop the zero-respecting requirement and extend A to all the deterministic (not in the sense of sampling but the actual executions) algorithms. At a cost, we would need the function class to follow an "orthogonal invariant" property, and the model dimension needs to be large enough. We leave this discussion to the appendix.

the *iteration complexity* of A solving f under O and G is defined as

$$T_\epsilon(A, f, O, G) = \min\left\{t \in \mathbb{N} \,\middle|\, \mathbb{E}\left\|\nabla f(\hat{\boldsymbol{x}}_t^{A,f,O,G})\right\| \leq \epsilon\right\},$$

that is, the least number of iterations required by A to find a ϵ-stationary-in-expectation point of f.

4. Lower Bound

Given the setup in Section 3, we can now present and discuss our lower bound on the iteration complexity. Note that in the formulation of protocol layer, the algorithm class $\mathcal{A}_B$ only specifies the information available for each worker, and thus $\mathcal{A}_B$ covers both centralization and decentralization in the protocol layer. Here, we show our lower bound in two parts: first a general bound where an arbitrary protocol that follows $\mathcal{A}_B$ is allowed, and then a corollary bound for the case where only decentralized protocol is allowed.

4.1 Lower Bound for Arbitrary Protocol

We start from the general bound. We expect this lower bound to show given arbitrary setting (functions, oracles and graph), the smallest iteration complexity we could obtain from $\mathcal{A}_B$, i.e.

$$\inf_{A \in \mathcal{A}_B} \sup_{f \in \mathcal{F}_{\Delta,L}} \sup_{O \in \mathcal{O}_{\sigma^2}} \sup_{G \in \mathcal{G}_{n,D}} T_\epsilon(A, f, O, G), \tag{9}$$

it suffices to construct a hard instance containing a loss function $\hat{f} \in \mathcal{F}_{\Delta,L}$, a graph $\hat{G} \in \mathcal{G}_{n,D}$ and a set of oracles $\hat{O} \in \mathcal{O}_{\sigma^2}$ and obtain a valid lower bound on $\inf_{A \in \mathcal{A}_B} T_\epsilon(A, \hat{f}, \hat{O}, \hat{G})$ since Equation (9) is always lower bounded by $\inf_{A \in \mathcal{A}_B} T_\epsilon(A, \hat{f}, \hat{O}, \hat{G})$.

For the construction, we follow the idea of probabilistic zero-chain model Carmon et al. (2017, 2019); Arjevani et al. (2019); Zhou and Gu (2019), which is a special loss function where adjacent coordinates are closely dependent on each other like a "chain." Our main idea is to use this function as f and split this chain onto different workers. Then the workers must conduct a sufficient number of optimization steps and rounds of communication to make progress.[5] From this, we obtain the following lower bound.

Theorem 1 *For function class $\mathcal{F}_{\Delta,L}$, oracle class $\mathcal{O}_{\sigma^2}$ and graph class $\mathcal{G}_{n,D}$ defined with any $\Delta > 0$, $L > 0$, $n \in \mathbb{N}^+$, $D \in \{1, 2, \ldots, n-1\}$, $\sigma > 0$, and $B \in \mathbb{N}^+$, there exists $f \in \mathcal{F}_{\Delta,L}$, $O \in \mathcal{O}_{\sigma^2}$, and $G \in \mathcal{G}_{n,D}$, such that no matter what $A \in \mathcal{A}_B$ is used, $T_\epsilon(A, f, O, G)$ will always be lower bounded by*

$$\Omega\left(\frac{\Delta L \sigma^2}{n B \epsilon^4} + \frac{\Delta L D}{\epsilon^2}\right). \tag{10}$$

Dependency on the parameters. The bound in Theorem 1 consists of a sample complexity term, which is the dominant one for small ϵ, and a communication complexity term. We can see the increase of query budget B will only reduce the sample complexity. On the other hand, as the diameter D of a graph will generally increase as the number of

---

5. For brevity, we leave details in the supplementary material.

vertices n increases, we can observe a trade-off between two terms when the system scales up: when more workers join the system, the communication complexity will gradually become the dominant term.

Consistency with the literature. Theorem 1 is tightly aligned with the state-of-the-art bounds in many settings. With $n = B = D = 1$, we recover the tight bound for sequential stochastic non-convex optimization $\Theta(\Delta L\sigma^2\epsilon^{-4})$ as shown in Arjevani et al. (2019). With $\sigma = 0, D = 1$, we recover the tight bound for sequential non-stochastic non-convex optimization $\Theta(\Delta L\epsilon^{-2})$ as shown in Carmon et al. (2019). With $B = 1, D = 1$, we recover the tight bound for centralized training $\Theta(\Delta L\sigma^2(n\epsilon^4)^{-1})$ given in Li et al. (2014b).

Improvement upon previous results. Previous works like Seaman et al. (2017); Scaman et al. (2018) provide similar lower bounds in a convex setting which relates to the diameter. However, these results treat D as a fixed value, i.e., $D = n - 1$, and thus makes the bound to be only tight on linear graph. By comparison, Theorem 1 allows D to be chosen independently to n.

4.2 Lower Bound for Decentralized Protocol

The bound in Theorem 1 holds for both centralized and decentralized protocols. A natural question is: *How would the lower bound adapt if the protocol is restricted to be decentralized?* i.e., the quantity of

$$\inf_{A\in\mathcal{A}_{B,\boldsymbol{W}}} \sup_{f\in\mathcal{F}_{\Delta,L}} \sup_{O\in\mathcal{O}_{\sigma^2}} \sup_{G\in\mathcal{G}_{n,D}} T_\epsilon(A, f, O, G),$$

we can extend the lower bound to Gossip in the following corollary.

Corollary 2 *For every $\Delta > 0$, $L > 0$, $n \in \{2, 3, 4, \cdots\}$, $\sigma > 0$, and $B \in \mathbb{N}^+$, there exists a loss function $f \in \mathcal{F}_{\Delta,L}$, a set of underlying oracles $O \in \mathcal{O}_{\sigma^2}$, a gossip matrix $\boldsymbol{W} \in \mathcal{W}_n$ with second largest eigenvalue being $\lambda = \cos(\pi/n)$, and a graph $G \in \mathcal{G}_{n,D}$, such that no matter what $A \in \mathcal{A}_{B,\boldsymbol{W}}$ is used, $T_\epsilon(A, f, O, G)$ will always be lower bounded by*

$$\Omega\left(\frac{\Delta L\sigma^2}{nB\epsilon^4} + \frac{\Delta L}{\epsilon^2\sqrt{1-\lambda}}\right). \tag{11}$$

Gap in the existing algorithms. Comparing this lower bound with many state-of-the-art decentralized algorithms (Table 2), we can see they match on the sample complexity but leave a gap on the communication complexity. In many cases, the spectral gap significantly depends on the number of workers n and thus can be arbitrarily large. For example, when the graph G is a cycle graph or a linear graph, the gap of those baselines can increase by up to $O(n^6)$ Brooks et al. (2011); Gerencsér (2011)!

Extended Discussion on the Graph Type. Notice that in Corollary 2, the lower bound holds on the linear graph, which follows our definition since the we need to choose the "hardest" graph from the class. In a recent work (Yuan et al., 2022), the authors discuss leveraging the Ring-Lattice graph to admit general weight matrices ($\lambda \in [0, \cos(\pi/n)]$) while maintaining the optimal relation between the graph diameter and weight matrix connectivity.

Algorithm 1 Decentralized Stochastic Gradient Descent with Factorized Consensus Matrices (DeFacto) on worker i

---

Input: initialized model $\boldsymbol{x}_{0,i}$, a copy of model $\tilde{\boldsymbol{x}}_{0,i} \leftarrow \boldsymbol{x}_{0,i}$, gradient buffer $\boldsymbol{g} = \boldsymbol{0}$, step size $\{\alpha_k\}_{k\geq 0}$, a sequence of communication matrices $\{\boldsymbol{W}_r\}_{1\leq r\leq R}$ of size R, number of iterations T, neighbor list $\mathcal{N}_i$

1: **for** $t = 0, 1, \cdots, T-1$ **do**
2:      $k \leftarrow \lfloor t/2R \rfloor$.
3:      $r \leftarrow t \bmod 2R$.
4:      **if** $0 \leq r < R$ **then**
5:          Spend all B oracle budgets to compute stochastic gradient $\tilde{\boldsymbol{g}}$ at point $\boldsymbol{x}_{k,i}$ and accumulate it to gradient buffer: $\boldsymbol{g} \leftarrow \boldsymbol{g} + \tilde{\boldsymbol{g}}$.
6:      **else**
7:          Update model copy with the r-th matrix in $\{\boldsymbol{W}_r\}_{1\leq r\leq R}$:

$$\tilde{\boldsymbol{x}}_{t+1,i} \leftarrow \sum_{j\in\mathcal{N}_i\cup\{i\}} \tilde{\boldsymbol{x}}_{t,j}[\boldsymbol{W}_r]_{ji} \tag{12}$$

8:      **end if**
9:      **if** $r = 2R-1$ **then**
10:          Update Model: $\boldsymbol{x}_{t+1,i} \leftarrow \tilde{\boldsymbol{x}}_{t+1,i} - \alpha_k\boldsymbol{g}/R$.
11:          Reinitialize gradient buffer: $\boldsymbol{g} \leftarrow \boldsymbol{0}$.
12:          Copy the current model: $\tilde{\boldsymbol{x}}_{t+1,i} \leftarrow \boldsymbol{x}_{t+1,i}$.
13:      **end if**
14: **end for**
15: **return** $\hat{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_{T,i}$

---

5. DeFacto: Optimal Complexity in Theory

In the previous section we show the existing algorithms have a gap compared to the lower bound. This gap could indicate the algorithms are suboptimal, but it could also be explained by our lower bound being loose. In this section we answer this by proposing DeFacto, an example algorithm showing the lower bound is achievable, which verifies the tightness of our lower bound—showing that (10) would hold with equality and $\Theta(\cdot)$, not just $\Omega(\cdot)$.

We start with the following insight on the theoretical gap: the goal of communication is to let all the workers obtain information from neighbors. Ideally, the workers would, at each iteration, perform (8) with $\boldsymbol{W}^* = \mathbf{1}_n\mathbf{1}_n^\top/n$, where $\mathbf{1}_n$ is the n-dimensional all-one vector. We call this matrix the *Average Consensus* matrix. The Average Consensus is statistically equivalent to centralized communication (All-Reduce operation). However, due to the graph constraints, we can not use this $\boldsymbol{W}^*$ unless workers are fully connected; instead, a general method is to repeatedly apply a sequence communication matrices in consecutive iterations and let workers achieve or approach the Average Consensus. Previous work uses Gossip matrix $\boldsymbol{W}$ and expect $\prod_{r=1}^{R} \boldsymbol{W} \approx \mathbf{1}_n\mathbf{1}_n^\top/n$ for some R. This R is known to be proportional to the mixing time of the Markov Chain $\boldsymbol{W}$ defines Lu et al. (2020); Lu and De Sa (2020), which is related to the inverse of its spectral gap Levin and Peres (2017). This limits convergence depending on the spectrum of the $\boldsymbol{W}$ chosen. The natural question to ask here is: can we do better? What are the limits of how fast we can reach average consensus on a connectivity graph G? This question is answered by the following lemma.

Algorithm 2 Decentralized Stochastic Gradient Tracking with By-Phase Accelerated Gossip (DeTAG) on worker i

---

Input: initialized model $\boldsymbol{x}_{0,i}$, a copy of model $\tilde{\boldsymbol{x}}_{0,i} \leftarrow \boldsymbol{x}_{0,i}$, gradient tracker $\boldsymbol{y}_{0,i}$, gradient buffer $\boldsymbol{g}_{(0)} = \boldsymbol{g}_{(-1)} = \boldsymbol{0}$, step size $\{\alpha_k\}_{k \geq 0}$, a gossip matrix $\boldsymbol{W}$, number of iterations T, neighbor list $\mathcal{N}_i$

1: **for** $t = 0, 1, \cdots, T - 1$ **do**
2:      $k \leftarrow \lfloor t/R \rfloor$.
3:      $r \leftarrow t \bmod R$.
4:      Perform the r-th step in Accelerated Gossip:
$$\tilde{\boldsymbol{x}}_{t+1,i} \leftarrow AG(\tilde{\boldsymbol{x}}_{t,i}, \boldsymbol{W}, \mathcal{N}_i, i) \tag{13}$$
$$\boldsymbol{y}_{t+1,i} \leftarrow AG(\boldsymbol{y}_{t,i}, \boldsymbol{W}, \mathcal{N}_i, i) \tag{14}$$
5:      Spend all B oracle budgets to compute stochastic gradient $\tilde{\boldsymbol{g}}$ at point $\boldsymbol{x}_{k,i}$ and accumulate it to gradient buffer: $\boldsymbol{g}_{(k)} \leftarrow \boldsymbol{g}_{(k)} + \tilde{\boldsymbol{g}}/R$.
6:      **if** $r = R - 1$ **then**
7:          Update gradient tracker and model:
$$\boldsymbol{x}_{t+1,i} \leftarrow \tilde{\boldsymbol{x}}_{t+1,i} - \alpha_k \boldsymbol{y}_i \tag{15}$$
$$\boldsymbol{y}_{t+1,i} \leftarrow \boldsymbol{y}_{t+1,i} + \boldsymbol{g}_{(k)} - \boldsymbol{g}_{(k-1)} \tag{16}$$
8:          Reinitialize gradient buffer: $\boldsymbol{g}_{(k-1)} \leftarrow \boldsymbol{g}_{(k)}$ and then $\boldsymbol{g}_{(k)} \leftarrow \boldsymbol{0}$.
9:          Copy the current model: $\tilde{\boldsymbol{x}}_{t+1,i} \leftarrow \boldsymbol{x}_{t+1,i}$.
10:      **end if**
11: **end for**
12: **return** $\hat{\boldsymbol{x}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_{T,i}$

Algorithm 3 Accelerated Gossip (AG) with R steps

Input: $\boldsymbol{z}_{0,i}$, $\boldsymbol{W}$, $\mathcal{N}_i$, i

1: $\boldsymbol{z}_{-1,i} \leftarrow \boldsymbol{z}_{0,i}$
2: $\eta \leftarrow \frac{1 - \sqrt{1 - \lambda^2}}{1 + \sqrt{1 - \lambda^2}}$
3: **for** $r = 0, 1, 2, \cdots, R - 1$ **do**
4:      $\boldsymbol{z}_{r+1,i} \leftarrow (1 + \eta) \sum_{j \in \mathcal{N}_i \cup \{i\}} \boldsymbol{z}_{r,j} \boldsymbol{W}_{ji} - \eta \boldsymbol{z}_{r-1,i}$
5: **end for**
6: **return** $\boldsymbol{z}_{R,i}$

Lemma 3 *For any $G \in \mathcal{G}_{n,D}$, let $\mathcal{W}_G$ denote the set of $n \times n$ matrices such that for all $\boldsymbol{W} \in \mathcal{W}_G$, $\boldsymbol{W}_{ij} = 0$ if edge (i, j) does not appear in G. There exists a sequence of R matrices $\{\boldsymbol{W}_r\}_{r \in [R]}$ that belongs to $\mathcal{W}_G$ such that $R \in \{D, D + 1, \cdots, 2D\}$ and*

$$\boldsymbol{W}_{R-1} \boldsymbol{W}_{R-2} \cdots \boldsymbol{W}_0 = \frac{\mathbf{1}_n \mathbf{1}_n^\top}{n} = \boldsymbol{W}^*.$$

Lemma 3 is a classic result in the literature of graph theory. The formal proof and detailed methods to identify these matrices can be found in many previous works Georgopoulos (2011); Ko (2010); Hendrickx et al. (2014). Here we treat this as a black box procedure.[6]

Lemma 3 shows that we can achieve the exact average consensus by factorizing the matrix $\mathbf{1}_n \mathbf{1}_n^\top / n$, and we can obtain the factors from a preprocessing step. From here, the path to obtain an optimal rate becomes clear: starting from $t = 0$, workers first spend

---

6. We cover specific algorithms and details in the supplementary.

R iterations only computing stochastic gradients and then another R iterations to reach consensus communicating via factors from Lemma 3; they then repeat this process until a stationary point is found. We call this algorithm DeFacto (Algorithm 1).

DeFacto is statistically equivalent to centralized SGD operating $T/2R$ iterations with a mini-batch size of BR. It can be easily verified that DeFacto holds membership in $\mathcal{A}_B$. A straightforward analysis gives the convergence rate of DeFacto shown in the following Theorem.

Theorem 4 *Let A_1 denote Algorithm 1. For $\mathcal{F}_{\Delta,L}$, $\mathcal{O}_{\sigma^2}$ and $\mathcal{G}_{n,D}$ defined with any $\Delta > 0$, $L > 0$, $n \in \mathbb{N}^+$, $D \in \{1,2,\ldots,n-1\}$, $\sigma > 0$, and $B \in \mathbb{N}^+$, the convergence rate of A_1 running on any loss function $f \in \mathcal{F}_{\Delta,L}$, any graph $G \in \mathcal{G}_{n,D}$, and any oracles $O \in \mathcal{O}_{\sigma^2}$ is bounded by*

$$T_\epsilon(A_1, f, O, G) \leq O\left(\frac{\Delta L \sigma^2}{nB\epsilon^4} + \frac{\Delta LD}{\epsilon^2}\right). \tag{17}$$

Comparing Theorem 1 and Theorem 4, DeFacto achieves the optimal rate asymptotically. *This shows that our lower bound in Theorem 1 is tight.*

6. DeTAG: Optimal Complexity in Practice

In previous section, we propose DeFacto that achieves the lower bound. Despite its optimality, the design of DeFacto is unsatisfying in three aspects: (1) It compromises the throughput[7] by a factor of two because in each iteration, a worker either communicates with neighbors or computes gradients but not both. This fails to overlap communication and computation and creates extra idle time for the workers. (2) It needs to iterate over all the factor matrices before it can query the gradient oracle at subsequent parameters. When diameter D increases, the total time to finish such round will increase proportionally. (3) DeFacto works with decentralized data and arbitrary graph, achieving decentralization in both application and topology layers. However, the matrices used in Lemma 3 are not Gossip matrices as defined in $\mathcal{W}_n$, and thus it fails to be decentralized in the protocol-layer sense.

6.1 Towards Optimality with Three-Layer Decentralization

To address the limitations of DeFacto, a natural idea is to replace all the factor matrices in Lemma 3 with a gossip matrix $\boldsymbol{W}$. The new algorithm after this mild modification is statistically equivalent to a D-PSGD variant: every R iterations, it updates the model the same as one iteration in D-PSGD with a mini-batch size of BR and communicate with a matrix $\boldsymbol{W}'$ whose second largest eigenvalue $\lambda' = \lambda^R$, with T/R iterations in total. However, even with arbitrarily large R, the communication complexity in this "updated D-PSGD" is still $O(\Delta Ln\varsigma\epsilon^{-2})$ (Table 2), leaving an $O(n\varsigma)$ gap compared to our lower bound.

To close this gap, we adopt two additional techniques:[8] one is a gradient tracker $\boldsymbol{y}$ that is used as reference capturing gradient difference in the neighborhood; the other is using acceleration in gossip as specified in Algorithm 3. Modifying DeFacto results in Algorithm 2, which we call DeTAG. DeTAG works as follows: it divides the total number of

---

7. The number of stochastic gradients computed per iteration.
8. Note that neither of these techniques is our original design, and we do not take credit for them. Our main contribution here is to prove their combination leads to optimal complexity.

iterations T into several phases where each phase contains R iterations. In each iteration, the communication process calls Accelerated Gossip to update a model replica $\tilde{x}$ and the gradient tracker (line 4) while the computation process constantly computes gradients at the same point (line 5). At the end of each phase, model x, its replica $\tilde{x}$ and gradient tracker y are updated in line 7-10 and then DeTAG steps into the next phase. Aside from the two additional techniques, the main difference between DeTAG and DeFacto is that the communication matrix in DeTAG is a fixed gossip matrix W, which allows DeTAG to benefit from decentralization in the protocol layer as well as to adopt arbitrary $R \geq 1$ in practice (allowing R to be tuned independently of G).

Improvement on design compared to baselines. Comparing with other baselines in Table 2, the design of DeTAG improves in the sense that (1) It removes the dependency on the outer variance ς. (2) It drops the requirement[9] on the gossip matrix assumed in Tang et al. (2018b). (3) The baseline DSGT Zhang and You (2019c) and GT-DSGD Xin et al. (2021b) can be seen as special cases of taking $R = 1$ and $\eta = 0$ in DeTAG. That implies in practice, a well tuned DeTAG can never perform worse than the baseline DSGT or GT-DSGD.

The convergence rate of DeTAG is given in the following theorem.

Theorem 5 *For $\mathcal{F}_{\Delta,L}$, $\mathcal{O}_{\sigma^2}$ and $\mathcal{G}_{n,D}$ defined with any $\Delta > 0$, $L > 0$, $n \in \mathbb{N}^+$, $\lambda \in [0, 1)$, $\sigma > 0$, and $B \in \mathbb{N}^+$, under the assumption of Equation (6), if we adopt a non-increasing step size scheme $\{\alpha_k\}_{k \geq 0}$ that fulfills*

$$\alpha_0 \leq \frac{(1-\rho)^2}{32L}$$

the convergence rate of Algorithm 2 running on any loss function $f \in \mathcal{F}_{\Delta,L}$, any graph $G \in \mathcal{G}_{n,D}$, and any oracles $O \in \mathcal{O}_{\sigma^2}$ is bounded by

$$\sum_{k=0}^{T/R-1} \alpha_k \mathbb{E} \left\| \nabla f \left(\overline{\boldsymbol{X}}_k \right) \right\|^2$$

$$\leq O \left(\Delta + \frac{\sigma^2 L}{nBR} \sum_{k=0}^{T/R-1} \alpha_k^2 + \frac{\rho^2 \alpha_0^3 L^2 T \varsigma_0^2}{(1-\rho)^3 R} + \frac{\rho^4 \sigma^2 L^2}{(1-\rho)^3 BR} \sum_{k=0}^{T/R-1} \alpha_k^3 + \frac{\sigma^2 L^2}{nBR} \sum_{k=0}^{T/R-1} \alpha_k^3 \right),$$

where $\rho = \left(1 - \sqrt{1-\lambda} \right)^R$.

Based on Theorem 5, the optimality statement is then given in the following corollary.

Corollary 6 *Let A_2 denote Algorithm 2. Based on Theorem 5, if we set the phase length R to be*

$$R = \frac{\max \left(\frac{1}{2} \log(n), \frac{1}{2} \log \left(\frac{\varsigma_0^2 T}{\Delta L} \right) \right)}{\sqrt{1-\lambda}},$$

and set step size to be a constant α with

$$\alpha = \frac{1}{\sqrt{\frac{\sigma^2 TL}{BR^2 n \Delta}} + \frac{\rho^{\frac{2}{3}} L^{\frac{2}{3}} \varsigma_0^{\frac{2}{3}} T^{\frac{1}{3}}}{R^{\frac{1}{3}} \Delta^{\frac{1}{3}} (1-\rho)} + \frac{32L}{(1-\rho)^2}},$$

---

9. Tang et al. (2018b) requires the gossip matrix to be symmetric and its smallest eigenvalue is lower bounded by $-\frac{1}{3}$.

where $\rho = \left(1 - \sqrt{1 - \lambda}\right)^R$, the convergence rate of A_2 running on any loss function $f \in \mathcal{F}_{\Delta,L}$, any graph $G \in \mathcal{G}_{n,D}$, and any oracles $O \in \mathcal{O}_{\sigma^2}$ is bounded by

$$T_\epsilon(A_2, f, O, G) \leq O\left(\frac{\Delta L \sigma^2}{nB\epsilon^4} + \frac{\Delta L \log\left(n + \frac{\varsigma_0 n}{\epsilon\sqrt{\Delta L}}\right)}{\epsilon^2 \sqrt{1 - \lambda}}\right).$$

Comparing Theorem 1 and Corollary 6, DeTAG achieves the optimal complexity with only a logarithm gap.

Improvement on complexity. Revisiting Table 2, we can see the main improvement of DeTAG's complexity is in the two terms on communication complexity: (1) DeTAG only depends on the outer variance term ς_0 inside a log, and (2) It reduces the dependency on the spectral gap $1 - \lambda$ to the lower bound of square root, as shown in Corollary 2.

Understanding the phase length R. In DeTAG, the phase length R is a tunable parameter. Theorem 5 provides a suggested value for R. Intuitively, the value of R captures the level of consensus of workers should reach before they step into the next phase. Theoretically, we observe R is closely correlated to the mixing time of $\boldsymbol{W}$: if we do not use acceleration in Gossip, then R will become $\tilde{O}\left(\frac{1}{1-\lambda}\right)$, which is exactly the upper bound on the mixing time of the Markov Chain $\boldsymbol{W}$ defines Levin and Peres (2017).

6.2 DeTAG with Momentum

In previous subsection, we show a plain version of DeTAG, where only SGD and constant step size are used, is sufficient to achieve the theoretical lower bound. In this section, we discuss a variant of DeTAG as specified in Algorithm 4 where momentum is used.

Comparing Algorithm 2 and 4, the main difference is in the update of model parameters at the end of phase – Algorithm 4 now used the momentum buffer instead of the stochastic gradient itself to update the gradient tracker. Comparing this design to previous decentralized algorithms with momentum such as (Yu et al., 2019), the gradient tracker eliminates the requirement on communicating the momentum buffer. The convergence rate of Algorithm 4 is given in the following theorem.

Theorem 7 *Let A_4 denote Algorithm 4. If in A_4, we set the phase length R to be*

$$R = \frac{\max\left(\frac{1}{2}\log(n), \frac{1}{2}\log\left(\frac{\varsigma_0^2 T}{\Delta L}\right)\right)}{\sqrt{1 - \lambda}},$$

set the momentum parameter $\frac{\beta^2}{1-\beta} \leq \tilde{\gamma}$ where $\tilde{\gamma}$ is a constant and set step size to be a constant α with

$$\alpha = \frac{1}{\sqrt{\frac{\sigma^2 TL}{(1-\beta)^2 BR^2 n\Delta}} + \frac{\rho^{\frac{2}{3}} L^{\frac{2}{3}} \varsigma_0^{\frac{2}{3}} T^{\frac{1}{3}}}{(1-\beta)^{\frac{2}{3}} R^{\frac{1}{3}} \Delta^{\frac{1}{3}} (1-\rho)} + \frac{32(1+\tilde{\gamma})L}{(1-\beta)^2 (1-\rho)^2}},$$

where $\rho = \left(1 - \sqrt{1 - \lambda}\right)^R$, the convergence rate of A_4 running on any loss function $f \in \mathcal{F}_{\Delta,L}$, any graph $G \in \mathcal{G}_{n,D}$, and any oracles $O \in \mathcal{O}_{\sigma^2}$ is bounded by

$$T_\epsilon(A_4, f, O, G) \leq O\left(\frac{\Delta L \sigma^2}{nB\epsilon^4} + \frac{\Delta L \log\left(n + \frac{\varsigma_0 n}{\epsilon\sqrt{\Delta L}}\right)}{(1 - \beta)\epsilon^2 \sqrt{1 - \lambda}}\right).$$

Algorithm 4 DeTAG with Momentum Acceleration (DeTAGM) on worker i

---

Input: initialized model $\boldsymbol{x}_{0,i}$, a copy of model $\tilde{\boldsymbol{x}}_{0,i} \leftarrow \boldsymbol{x}_{0,i}$, gradient tracker $\boldsymbol{y}_{0,i}$, gradient buffer $\boldsymbol{g}_{(0)} = \boldsymbol{0}$, momentum buffer $\boldsymbol{v}_{(0)} = \boldsymbol{v}_{(-1)} = \boldsymbol{0}$, momentum constant β, step size $\{\alpha_k\}_{k \geq 0}$, a gossip matrix $\boldsymbol{W}$, number of iterations T, neighbor list $\mathcal{N}_i$

1: **for** $t = 0, 1, \cdots, T-1$ **do**
2:     $k \leftarrow \lfloor t/R \rfloor$.
3:     $r \leftarrow t \bmod R$.
4:     Perform the r-th step in Accelerated Gossip:

$$\tilde{\boldsymbol{x}}_{t+1,i} \leftarrow AG(\tilde{\boldsymbol{x}}_{t,i}, \boldsymbol{W}, \mathcal{N}_i, i) \tag{18}$$

$$\boldsymbol{y}_{t+1,i} \leftarrow AG(\boldsymbol{y}_{t,i}, \boldsymbol{W}, \mathcal{N}_i, i) \tag{19}$$

5:     Spend all B oracle budgets to compute stochastic gradient $\tilde{\boldsymbol{g}}$ at point $\boldsymbol{x}_{k,i}$ and accumulate it to gradient buffer: $\boldsymbol{g}_{(k)} \leftarrow \boldsymbol{g}_{(k)} + \tilde{\boldsymbol{g}}/R$.
6:     **if** $r = R-1$ **then**
7:         Update gradient tracker, model and momentum buffer:

$$\boldsymbol{v}_{(k+1)} \leftarrow \beta \boldsymbol{v}_{(k)} + \boldsymbol{g}_{(k)} \tag{20}$$

$$\boldsymbol{x}_{t+1,i} \leftarrow \tilde{\boldsymbol{x}}_{t+1,i} - \alpha_k \boldsymbol{y}_i \tag{21}$$

$$\boldsymbol{y}_{t+1,i} \leftarrow \boldsymbol{y}_{t+1,i} + \boldsymbol{v}_{(k)} - \boldsymbol{v}_{(k-1)} \tag{22}$$

8:         Reinitialize gradient buffer: $\boldsymbol{g}_{(k-1)} \leftarrow \boldsymbol{g}_{(k)}$ and then $\boldsymbol{g}_{(k)} \leftarrow \boldsymbol{0}$.
9:         Copy the current model: $\tilde{\boldsymbol{x}}_{t+1,i} \leftarrow \boldsymbol{x}_{t+1,i}$.
10:     **end if**
11: **end for**
12: **return** $\hat{\boldsymbol{x}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_{T,i}$

Comparison with Related Algorithms. From Theorem 7 we observe that momentum-enabled DeTAG has an additional factor $(1-\beta)$ only on the non-dominant term $O(\epsilon^{-2})$. This dependency is better than the one shown in (Yu et al., 2019). On the other hand, Algorithm 4 (DeTAGM) improves upon the recent state-of-the-art decentralized momentum SGD (Lin et al., 2021) in three aspects: (1) DeTAGM obtains the same convergence guarantee without averaging the momentum buffer, this suggests the usage of gradient tracker implicitly makes all the workers reach consensus on the momentum buffer. (2) DeTAGM gives a better dependency on the spectral gap, improving it from $1 - \lambda$ (as shown in (Lin et al., 2021) Theorem 3.1) to $\sqrt{1-\lambda}$. (3) DeTAGM allows a more relaxed condition on β. More concretely, Lin et al. (2021) requires $\beta/(1-\beta) \leq (1-\lambda)/21$ which does not allow the β to be chosen freely with respect to the mixing matrix. In contrast, DeTAGM only requires $\beta^2/(1-\beta) \leq \tilde{\gamma}$ with a constant $\tilde{\gamma}$.

7. Experiments

In this section we empirically evaluate DeTAG and baseline algorithms[10]: D-PSGD Lian et al. (2017a), D^2 Tang et al. (2018b), DSGT Zhang and You (2019c) and DeTAG. Note that GT-DSGD Xin et al. (2021b) and DSGT Zhang and You (2019c) are essentially the same algorithm so we omit the comparison to GT-DSGD. Also note that SGP Assran et al.

---

10. Since DeFacto is a only a "motivation" algorithm and in practice we observe it performs bad, we do not include the discussion of that.

(a) 100% Shuffled CIFAR10

(b) 50% Shuffled CIFAR10
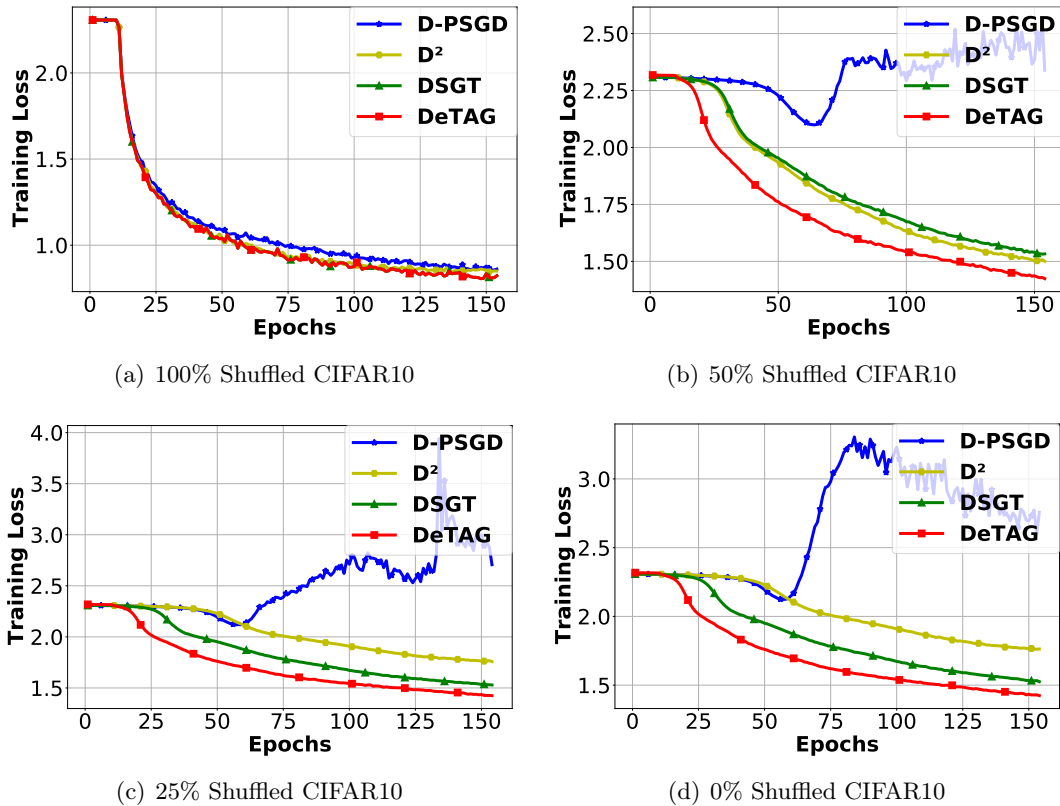
(c) 25% Shuffled CIFAR10

(d) 0% Shuffled CIFAR10

Figure 2: Fine tuned results of training LeNet on CIFAR10 with different shuffling strategies.

(2019) reduces to D-PSGD for symmetric mixing matrices in undirected graphs. As shown in Table 2, the main improvement of DeTAG are on the two parameters outer variance ς_0 and spectral gap $1 - \lambda$. In Subsection 7.1, we train LeNet and Resnet20 on CIFAR10/100 under different values of ς_0 and $1 - \lambda$. We use 8-GPU ring graph there and uses each GPU as an individual worker. Subsequently, in Subsection 7.2, we extend the system to a 32-GPU ring graph and train Resnet18 on ImageNet. We propose a new variant of DeTAG, DeTAG*, that allows larger-batch training in data centers. Hyperparameters can be found in the supplementary material.

7.1 Validation on Theory

Convergence over different outer variance. In the first experiments, we investigate the correlation between convergence speed and the outer variance $\varsigma(\varsigma_0)$. We train LeNet on CIFAR10 using 8 workers, which is a standard benchmark experiment in the decentralized data environment Tang et al. (2018b); Zhang and You (2019c). To create the decentralized data, we first sort all the data points based on its labels, shuffle the first $X\%$ data points and then evenly split to different workers. The X controls the degree of decentralization, we test $X = 0, 25, 50, 100$ and plot the results in Figure 2.

(a) $\kappa = 1$ $(1 - \lambda \approx 4e\text{-}2)$

(b) $\kappa = 0.1$ $(1 - \lambda \approx 4e\text{-}3)$

(c) $\kappa = 0.05$ $(1 - \lambda \approx 2e\text{-}3)$

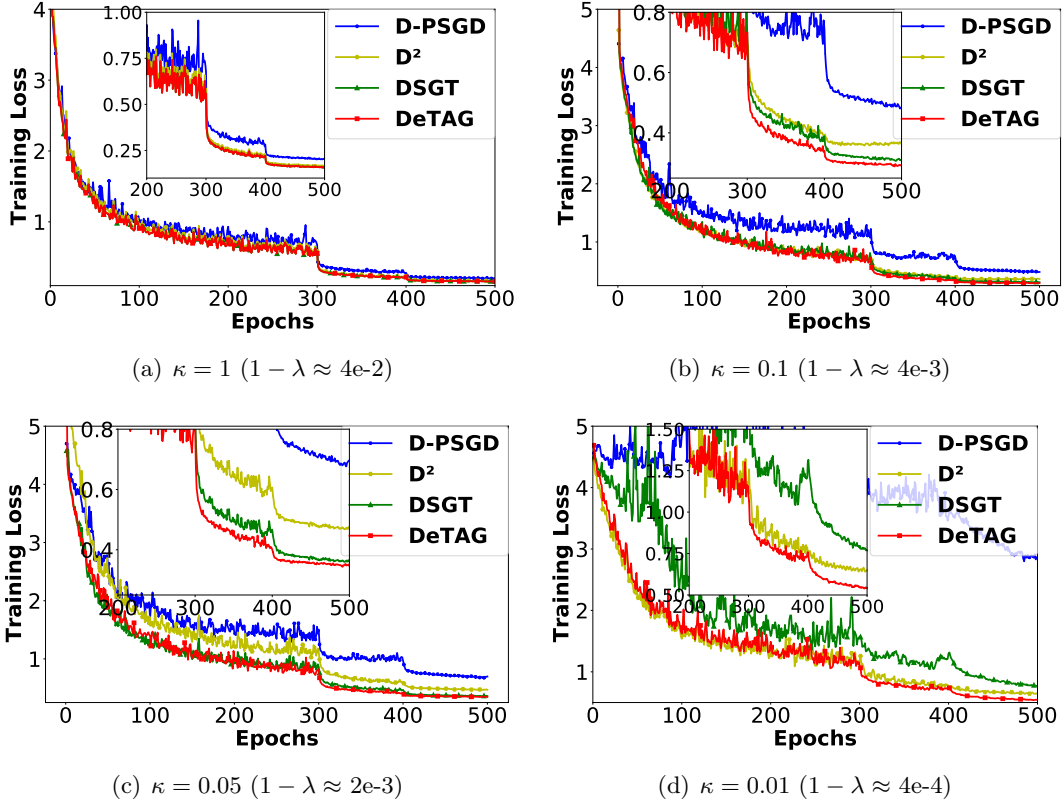(d) $\kappa = 0.01$ $(1 - \lambda \approx 4e\text{-}4)$

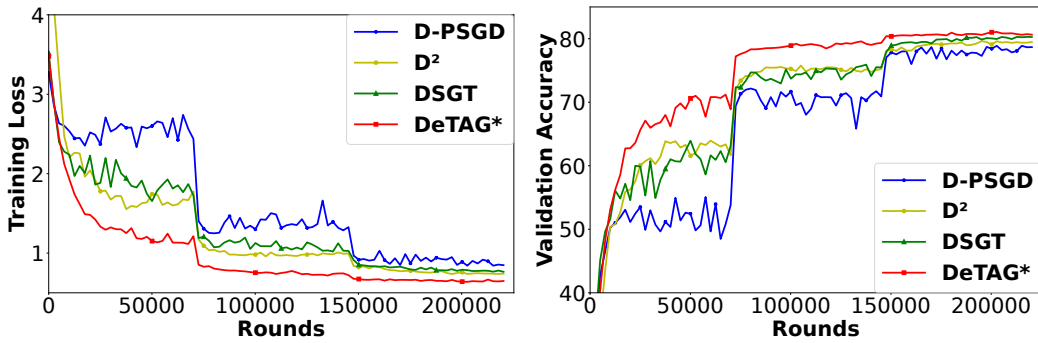Figure 3: Fine tuned results of training Resnet20 on CIFAR100 with different spectral gaps.

Figure 4: Results of Resnet18 on Imagenet among different algorithms.

We can see in Figure 2(a) when the dataset is fully shuffled, all the algorithms converge at similar speed while D-PSGD converges a little slower than other variance reduced algorithms. From Figure 2(b) to Figure 2(d) we can see when we shuffle less portion of the dataset, i.e., the dataset becomes more decentralized, D-PSGD fails to converge even with fine-tuned hyperparameter. Meanwhile, among D^2, DSGT and DeTAG, we can see DeTAG converges
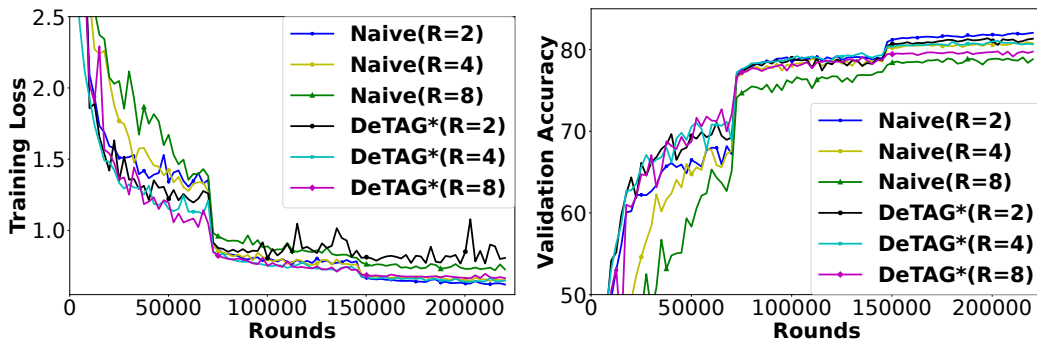
18

Figure 5: We compare DeTAG with hyperparameter R and its naive version – single round with R times large mini-batch size. The difference between them is that the former additionally includes $R-1$ Gossip rounds. We show from this experiment that multi-rounds Gossip helps enabling larger-batch training, which is critical in data-center-scale model training.

the fastest. When dataset becomes more decentralized, DSGT seems to receive more stable performance than D^2.

Convergence over different spectral gaps. In the second experiments, we proceed to explore the relation between convergence speed and spectral gap $1-\lambda$ of the gossip matrix $\boldsymbol{W}$. We use 16 workers connected with a Ring graph to train Resnet20 on CIFAR100, and we generate a $\boldsymbol{W}_0$ on such graph using Metropolis method. Then we adopt the slack matrix method to modify the spectral gap Lu et al. (2020): $\boldsymbol{W}_\kappa = \kappa\boldsymbol{W}_0 + (1-\kappa)\boldsymbol{I}$, where κ is a control parameter. We test $\kappa = 1, 0.1, 0.05, 0.01$ and plot the results in Figure 3. We can see with different κ, DeTAG is able to achieve faster convergence compared to baselines. When the network becomes sparse, i.e., κ decreases, DeTAG enjoys more robust convergence.

7.2 DeTAG*: Practical Version of DeTAG in Data-Center-Scale Model Training

In previous sections, our theory indicates that under the same oracle and communication budget, simply rearranging the communication rounds allows us to achieve optimality on iteration complexity. Such a strategy not only benefits statistically, but also provides significant flexibility in system optimization. One obvious optimization is parallelism: since now gradients are computed consecutively on the same model parameters, the back propagation across multiple oracle queries can be carried out in parallel; with the same rationale, messages sent in different communication rounds can now share the same connection – reducing fixed costs of initializing a new round.

Motivated by these insights, in this section we evaluate another variant of DeTAG – instead of overlapping the oracle queries and communication rounds, we now let them operate separately. Concretely, in Algorithm 2, the communication is performed on a stale version of model parameters $\tilde{\boldsymbol{x}}$; now we let it communicate over $\boldsymbol{x}$. Note that this does not require additional gradient oracles or communication rounds, only reordering them. We call this

variant DeTAG* and compare this simple variant with other baselines in a large-scale setting. We use 32 GPUs with a Ring graph to train Resnet18 on Imagenet[11]. We plot the results in Figure 4 where we use $R = 4$ and $\eta = 0$ for DeTAG*. We can see DeTAG* consistently converges faster compared to other baselines in terms of oracle queries and communication rounds.

On the other hand, the main idea of DeTAG* seems to be increasing the mini-batch size via reordering the gradient queries. However, the design of multiple rounds of Gossip also plays a crucial role. We perform an ablation study in Figure 5, comparing DeTAG* with naively increasing mini-batch size but not communication rounds with same phase length. We can see that with multi-round Gossip, DeTAG* generally allows larger-batch training with less drop on the model accuracy.

8. Conclusion

In this paper, we investigate the tight lower bound on the iteration complexity of decentralized training. We propose two algorithms, DeFacto and DeTAG, that achieve the lower bound in terms of different decentralization in a learning system. DeTAG uses Gossip protocol, and is shown to be empirically competitive to many baseline algorithms, such as D-PSGD. We additionally study two DeTAG variants – DeTAG with momentum acceleration and DeTAG*. In the future, we plan to investigate the variants of the complexity bound with respect to communication that are compressed, asynchronous, etc.

Acknowledgement

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

Alekh Agarwal and Leon Bottou. A lower bound for the optimization of finite sums. *arXiv preprint arXiv:1410.0723*, 2014.

Dan Alistarh. A brief tutorial on distributed and concurrent machine learning. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, pages 487–488, 2018.

---

11. Note that the test crop we adopted in the experiment is 320x320, so that the error rate is around 20%. We notice some open source implementation on this task adopts test crop size of 224x224, and so the error rate is around 30%. More details can be found in `https://github.com/facebookarchive/fb.resnet.torch/blob/master/pretrained/README.md`.

Dan Alistarh, Bapi Chatterjee, and Vyacheslav Kungurtsev. Elastic consistency: A general consistency model for distributed stochastic gradient descent. *arXiv preprint arXiv:2001.05918*, 2020.

Zeyuan Allen-Zhu. How to make the gradients small stochastically: Even faster convex and nonconvex sgd. In *Advances in Neural Information Processing Systems*, pages 1157–1167, 2018.

Yossi Arjevani and Ohad Shamir. Communication complexity of distributed convex learning and optimization. In *Advances in neural information processing systems*, pages 1756–1764, 2015.

Yossi Arjevani and Ohad Shamir. Oracle complexity of second-order methods for finite-sum problems. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 205–213. JMLR. org, 2017.

Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Michael Rabbat. Stochastic gradient push for distributed deep learning. *arXiv preprint arXiv:1811.10792*, 2018.

Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Mike Rabbat. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, pages 344–353. PMLR, 2019.

Eric Balkanski and Yaron Singer. Parallelization does not accelerate convex optimization: Adaptivity lower bounds for non-smooth convex minimization. *arXiv preprint arXiv:1808.03880*, 2018.

Raphaël Berthier, Francis Bach, and Pierre Gaillard. Accelerated gossip in networks of given dimension using jacobi polynomial iterations. *SIAM Journal on Mathematics of Data Science*, 2(1):24–47, 2020.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečný, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Gossip algorithms: Design, analysis and applications. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 3, pages 1653–1664. IEEE, 2005.

Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE transactions on information theory*, 52(6):2508–2530, 2006.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points ii: First-order methods. *arXiv preprint arXiv:1711.00841*, 2017.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, pages 1–50, 2019.

Igor Colin, Ludovic Dos Santos, and Kevin Scaman. Theoretical limits of pipeline parallel optimization and application to distributed deep learning. In *Advances in Neural Information Processing Systems*, pages 12350–12359, 2019.

Jelena Diakonikolas and Cristóbal Guzmán. Lower bounds for parallel and randomized convex optimization. *arXiv preprint arXiv:1811.01903*, 2018.

John C Duchi, Alekh Agarwal, and Martin J Wainwright. Distributed dual averaging in networks. In *NIPS*, pages 550–558. Citeseer, 2010.

Darina Dvinskikh and Alexander Gasnikov. Decentralized and parallelized primal and dual accelerated methods for stochastic convex programming problems. *arXiv preprint arXiv:1904.09015*, 2019.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal nonconvex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pages 689–699, 2018.

Dylan Foster, Ayush Sekhari, Ohad Shamir, Nathan Srebro, Karthik Sridharan, and Blake Woodworth. The complexity of making the gradient small in stochastic convex optimization. *arXiv preprint arXiv:1902.04686*, 2019.

Leonidas Georgopoulos. Definitive consensus for distributed data inference. Technical report, EPFL, 2011.

Balázs Gerencsér. Markov chain mixing time on cycles. *Stochastic processes and their applications*, 121(11):2553–2570, 2011.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

William Gropp, Rajeev Thakur, and Ewing Lusk. *Using MPI-2: Advanced features of the message passing interface*. MIT press, 1999.

Lie He, An Bian, and Martin Jaggi. Cola: Decentralized linear learning. In *Advances in Neural Information Processing Systems*, pages 4536–4546, 2018.

Julien M Hendrickx, Raphaël M Jungers, Alexander Olshevsky, and Guillaume Vankeerberghen. Graph diameter, eigenvalues, and minimum-time consensus. *Automatica*, 50(2): 635–640, 2014.

Hadrien Hendrikx, Francis Bach, and Laurent Massoulié. Asynchronous accelerated proximal stochastic gradient for strongly convex distributed finite sums. *arXiv preprint arXiv:1901.09865*, 2019.

Qirong Ho, James Cipar, Henggang Cui, Seunghak Lee, Jin Kyu Kim, Phillip B Gibbons, Garth A Gibson, Greg Ganger, and Eric P Xing. More effective distributed ml via a stale synchronous parallel parameter server. In *Advances in neural information processing systems*, pages 1223–1231, 2013.

Ameeth Kanawaday and Aditya Sane. Machine learning for predictive maintenance of industrial machines using iot sensor data. In *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 87–90. IEEE, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Chih-Kai Ko. *On matrix factorization and scheduling for finite-time average-consensus*. PhD thesis, California Institute of Technology, 2010.

Anastasia Koloskova, Tao Lin, Sebastian U Stich, and Martin Jaggi. Decentralized deep learning with arbitrary communication compression. *arXiv preprint arXiv:1907.09356*, 2019a.

Anastasia Koloskova, Sebastian U Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. *arXiv preprint arXiv:1902.00340*, 2019b.

Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U Stich. A unified theory of decentralized sgd with changing topology and local updates. *arXiv preprint arXiv:2003.10422*, 2020.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Guanghui Lan and Yi Zhou. An optimal randomized incremental gradient method. *Mathematical programming*, 171(1-2):167–215, 2018.

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

Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14)*, pages 583–598, 2014a.

Mu Li, David G Andersen, Alexander J Smola, and Kai Yu. Communication efficient distributed machine learning with the parameter server. In *Advances in Neural Information Processing Systems*, pages 19–27, 2014b.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 5330–5340, 2017a.

Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. *arXiv preprint arXiv:1710.06952*, 2017b.

Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local sgd. *arXiv preprint arXiv:1808.07217*, 2018.

Tao Lin, Sai Praneeth Karimireddy, Sebastian U Stich, and Martin Jaggi. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. *arXiv preprint arXiv:2102.04761*, 2021.

Ji Liu and A Stephen Morse. Accelerated linear iterations for distributed averaging. *Annual Reviews in Control*, 35(2):160–165, 2011.

Ji Liu and Ce Zhang. Distributed learning systems with first-order methods. *arXiv preprint arXiv:2104.05245*, 2021.

Yucheng Lu and Christopher De Sa. Moniqua: Modulo quantized communication in decentralized sgd. *arXiv preprint arXiv:2002.11787*, 2020.

Yucheng Lu and Christopher De Sa. Optimal complexity in decentralized training. In *International Conference on Machine Learning*, pages 7111–7123. PMLR, 2021.

Yucheng Lu, Jack Nash, and Christopher De Sa. Mixml: A unified analysis of weakly consistent parallel learning. *arXiv preprint arXiv:2005.06706*, 2020.

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

Parvin Nazari, Davoud Ataee Tarzanagh, and George Michailidis. Dadam: A consensus-based distributed adaptive gradient method for online optimization. *arXiv preprint arXiv:1901.09109*, 2019.

Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o $(1/k^2)$. In *Doklady an ussr*, volume 269, pages 543–547, 1983.

Pitch Patarasuk and Xin Yuan. Bandwidth optimal all-reduce algorithms for clusters of workstations. *Journal of Parallel and Distributed Computing*, 69(2):117–124, 2009.

Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *international conference on machine learning*, pages 3027–3036. PMLR, 2017.

Kevin Scaman, Francis Bach, Sébastien Bubeck, Laurent Massoulié, and Yin Tat Lee. Optimal algorithms for non-smooth distributed optimization in networks. In *Advances in Neural Information Processing Systems*, pages 2740–2749, 2018.

Kevin Seaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3027–3036. JMLR. org, 2017.

Uday Shankar Shanthamallu, Andreas Spanias, Cihan Tepedelenlioglu, and Mike Stanley. A brief survey of machine learning methods and their sensor and iot applications. In *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pages 1–8. IEEE, 2017.

Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

Haoran Sun and Mingyi Hong. Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms. *IEEE Transactions on Signal processing*, 67(22):5912–5928, 2019.

Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. In *Advances in Neural Information Processing Systems*, pages 7652–7662, 2018a.

Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D2: Decentralized training over decentralized data. *arXiv preprint arXiv:1803.07068*, 2018b.

Hanlin Tang, Xiangru Lian, Shuang Qiu, Lei Yuan, Ce Zhang, Tong Zhang, and Ji Liu. Deepsqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. *arXiv preprint arXiv:1907.07346*, 2019a.

Hanlin Tang, Xiangru Lian, Chen Yu, Tong Zhang, and Ji Liu. Doublesqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. *arXiv preprint arXiv:1905.05957*, 2019b.

Ye Tian, Ying Sun, and Gesualdo Scutari. Achieving linear convergence in distributed asynchronous multiagent optimization. *IEEE Transactions on Automatic Control*, 65(12): 5264–5279, 2020.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

Nguyen H Tran, Wei Bao, Albert Zomaya, Minh NH Nguyen, and Choong Seon Hong. Federated learning over wireless networks: Optimization model design and analysis. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 1387–1395. IEEE, 2019.

Quoc Tran-Dinh, Ahmet Alacaoglu, Olivier Fercoq, and Volkan Cevher. An adaptive primal-dual framework for nonsmooth convex minimization. *Mathematical Programming Computation*, pages 1–41, 2019.

Xinchen Wan, Hong Zhang, Hao Wang, Shuihai Hu, Junxue Zhang, and Kai Chen. Rat-resilient allreduce tree for distributed machine learning. In *4th Asia-Pacific Workshop on Networking*, pages 52–57, 2020.

Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. *arXiv preprint arXiv:1808.07576*, 2018.

Jianyu Wang, Anit Kumar Sahu, Zhouyi Yang, Gauri Joshi, and Soummya Kar. Matcha: Speeding up decentralized sgd via matching decomposition sampling. *arXiv preprint arXiv:1905.09435*, 2019.

Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes, from any initialization. *arXiv preprint arXiv:1806.01811*, 2018.

Blake E Woodworth, Jialei Wang, Adam Smith, Brendan McMahan, and Nati Srebro. Graph oracle models, lower bounds, and gaps for parallel stochastic optimization. In *Advances in neural information processing systems*, pages 8496–8506, 2018.

Ran Xin, Usman A Khan, and Soummya Kar. Variance-reduced decentralized stochastic optimization with gradient tracking. *arXiv preprint arXiv:1909.11774*, 2019.

Ran Xin, Usman A Khan, and Soummya Kar. A hybrid variance-reduced method for decentralized stochastic non-convex optimization. *arXiv preprint arXiv:2102.06752*, 2021a.

Ran Xin, Usman A Khan, and Soummya Kar. An improved convergence analysis for decentralized online stochastic non-convex optimization. *IEEE Transactions on Signal Processing*, 69:1842–1858, 2021b.

Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(3):1–207, 2019.

Haishan Ye, Luo Luo, Ziang Zhou, and Tong Zhang. Multi-consensus decentralized accelerated gradient descent. *arXiv preprint arXiv:2005.00797*, 2020.

Chen Yu, Hanlin Tang, Cedric Renggli, Simon Kassing, Ankit Singla, Dan Alistarh, Ce Zhang, and Ji Liu. Distributed learning over unreliable networks. *arXiv preprint arXiv:1810.07766*, 2018.

Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. *arXiv preprint arXiv:1905.03817*, 2019.

Kun Yuan, Xinmeng Huang, Yiming Chen, Xiaohan Zhang, Yingya Zhang, and Pan Pan. Revisiting optimal convergence rate for smooth and non-convex stochastic decentralized optimization. *arXiv preprint arXiv:2210.07863*, 2022.

Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Jiaqi Zhang and Keyou You. Asynchronous decentralized optimization in directed networks. *arXiv preprint arXiv:1901.08215*, 2019a.

Jiaqi Zhang and Keyou You. Asyspa: An exact asynchronous algorithm for convex optimization over digraphs. *IEEE Transactions on Automatic Control*, 65(6):2494–2509, 2019b.

Jiaqi Zhang and Keyou You. Decentralized stochastic gradient tracking for empirical risk minimization. *arXiv preprint arXiv:1909.02712*, 2019c.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Dongruo Zhou and Quanquan Gu. Lower bounds for smooth nonconvex finite-sum optimization. *arXiv preprint arXiv:1901.11224*, 2019.

Supplementary Material

Appendix A. Experimental Details

A.1 Hyperparameter Tuning

In the experiment of training LeNet on CIFAR10, we tune the step size using grid search inside the following range: {5e-3, 1e-3, 5e-4, 2.5e-4, 1e-4, 5e-5}. Note that this range is in general smaller than the one chosen in Zhang and You (2019c), since here we are working with unshuffled data, and we found original range in baselines causes algorithms to diverge easily. Following Tang et al. (2018b), we let each run warm up for 10 epochs with step size 1e-5. For DeTAG, we further tune the accelerated gossip parameter η within {0, 0.1, 0.2, 0.4} and phase length R within {1, 2, 3}. We fix the momentum term to be 0.9 and weight decay to be 1e-4.

In the experiment of training Resnet20 on CIFAR100, we tune the step size using grid search inside the following range: {0.5, 0.1, 0.05, 0.01, 0.005}. For DeTAG, we further tune the accelerated gossip parameter η within {0, 0.1, 0.2, 0.4} and phase length R within {1, 2, 3}. We fix the momentum term to be 0.9 and weight decay to be 5e-4.

The hyperparameters adopted for each runs are shown in Table 3 and Table 4.

A.2 Techniques of Running DeTAG

We can see in the main loop of DeTAG, several gradient queries are made at the same point. This essentially is equivalent to a large mini-batch size. In practice, however, we can modify this to use local-steps and get better empirical results Lin et al. (2018). Another technique is to use warm-up epochs when data is decentralized. We observe it ensures a smooth convergence in practice. Last but not least, since at first the noise in the algorithms is generally large, we can use a dynamic phase length to obtain better results. That is, we start from phase length 1 for the first few epochs, and let DeTAG follow the special case of DSGT. Then we can gradually increase the phase length following given policies. The intuition is that as algorithm converges, we would need less noise from communication, and thus a longer phase length can benefit.

Table 3: (Initial) Step size α used for each experiments.

Experiment	Setting	Algorithm			
		D-PSGD	D^2	DSGT	DeTAG
LeNet/CIFAR10	100% Shuffled	5e-3	5e-3	5e-3	5e-3
	50% Shuffled	5e-5	2.5e-4	2.5e-4	5e-4
	25% Shuffled	5e-5	1e-4	2.5e-4	5e-4
	0% Shuffled	5e-5	1e-4	2.5e-4	5e-4
Resnet20/CIFAR100	$\kappa = 1$	0.5	0.5	0.5	0.5
	$\kappa = 0.1$	0.5	0.5	0.5	0.5
	$\kappa = 0.05$	0.5	0.5	0.5	0.5
	$\kappa = 0.01$	0.5	0.5	0.5	0.5

Table 4: DeTAG-specific hyperparameters used for each experiments.

Experiment	Setting	Accelerate Factor η	Phase Length R
LeNet/CIFAR10	100% Shuffled	0	1
	50% Shuffled	0.2	2
	25% Shuffled	0.2	2
	0% Shuffled	0.2	2
Resnet20/CIFAR100	$\kappa = 1$	0	1
	$\kappa = 0.1$	0.2	2
	$\kappa = 0.05$	0.2	2
	$\kappa = 0.01$	0.4	2

Appendix B. Technical Proof

B.1 Proof to Theorem 1

Proof To prove this theorem, it suffices for us to provide two examples, each has a (set of) loss function $f \in \mathcal{F}_{\Delta,L}$, a set of underlying oracles $O \in \mathcal{O}_{\sigma^2}$, a graph $G \in \mathcal{G}_{n,D}$, such that $\inf_{A \in \mathcal{A}_B} T_\epsilon(A, f, O, G)$ is lower bounded by $\Omega\left(\frac{\Delta L \sigma^2}{nB\epsilon^4}\right)$ and $\Omega\left(\frac{\Delta L D}{\epsilon^2}\right)$ iterations on these two examples, respectively. Then we will obtain the final bound as $\max\left\{\Omega\left(\frac{\Delta L \sigma^2}{nB\epsilon^4}\right), \Omega\left(\frac{\Delta L D}{\epsilon^2}\right)\right\}$, i.e., $\Omega\left(\frac{\Delta L \sigma^2}{nB\epsilon^4} + \frac{\Delta L D}{\epsilon^2}\right)$ as desired. For simplicity, we denote $\boldsymbol{z}^{(i)}$ as the i-th coordinate of vector $\boldsymbol{z} \in \mathbb{R}^d$.

For each setting, our constructions contain three main steps.

(1) The first step is to follow the construction of a zero chain function model Carmon et al. (2017, 2019). Following Arjevani et al. (2019) and define

$$\text{prog}(\boldsymbol{z}) = \max\{i \geq 0 | \boldsymbol{z}^{(i)} \neq 0\}, \forall \boldsymbol{z} \in \mathbb{R}^d.$$

A zero chain function f has the following property:

$$\text{prog}(\nabla f(\boldsymbol{x})) \leq \text{prog}(\boldsymbol{x}) + 1,$$

that means, for a model start from $\boldsymbol{x} = \boldsymbol{0}$, a single gradient evaluation can only make at most one more coordinate to be non-zero. The name of "chain" comes from the fact that the adjacent coordinates are linked like a chain and only if the previous coordinate becomes non-zero that the current coordinate can become non-zero via a gradient update. Consider a model with d dimension, if we show that $\|\nabla f(\boldsymbol{x})\| \geq \epsilon$ for any $\boldsymbol{x} \in \mathbb{R}^d$ with $\boldsymbol{x}^{(d)} = 0$, we will obtain d as a lower bound on the gradient calls to obtain the ϵ-stationary point. We refer such sequential lower bound as T_0.

(2) Step two is to construct a graph $G \in \mathcal{G}_{n,D}$ and a set of oracle $O \in \mathcal{O}_{\sigma^2}$. To do this, our basic idea is to follow Arjevani et al. (2019) and introduce randomness on the $\text{prog}(\boldsymbol{x})$, and thus the whole chain only make progress with probability p. As will be shown later, this requires $\Omega(T_0/p)$ iterations in total.

(3) The third and last step is to rescale the function and distribution so as to make it belong to the function and oracle classes we consider. In other words, this step is to guarantee the result is shown in terms of Δ, L, σ, n and D.

We start from a smooth and (potentially) non-convex zero chain function $\hat{f}$ Carmon et al. (2019) as defined below:

$$\hat{f}(\boldsymbol{x}) = -\Psi(1)\Phi(\boldsymbol{x}^{(1)}) + \sum_{i=1}^{T-1}[\Psi(-\boldsymbol{x}^{(i)})\Phi(-\boldsymbol{x}^{(i+1)}) - \Psi(\boldsymbol{x}^{(i)}\Phi(\boldsymbol{x}^{(i+1)}))], \qquad (23)$$

where for $\forall z \in \mathbb{R}$

$$\Psi(z) = \begin{cases} 0 & z \leq 1/2 \\ \exp\left(1 - \frac{1}{(2z-1)^2}\right) & z > 1/2 \end{cases}, \qquad \Phi(z) = \sqrt{e} \int_{-\infty}^{z} e^{\frac{1}{2}t^2} dt.$$

This function, as shown in previous works Carmon et al. (2019); Arjevani et al. (2019), is a zero-chain function and thus is generally "hard" to optimize: it costs at least T gradient

evaluations to find a stationary point. We summarize some properties of Equation (23) as the following (Proof can be found in Lemma 2 in Arjevani et al. (2019)):

1. $\hat{f}(\boldsymbol{x}) - \inf_x \hat{f}(\boldsymbol{x}) \leq \Delta_0 T$, $\forall \boldsymbol{x} \in \mathbb{R}^d$, where $\Delta_0 = 12$.

2. $\hat{f}$ is l_1-smooth, where $l_1 = 152$.

3. $\forall \boldsymbol{x} \in \mathbb{R}^T$, $\|\nabla \hat{f}(\boldsymbol{x})\|_\infty \leq G_\infty$, where $G_\infty = 23$.

4. $\forall \boldsymbol{x} \in \mathbb{R}^T$, if $\text{prog}(\boldsymbol{x}) < T$, then $\|\hat{f}(\boldsymbol{x})\|_\infty \geq 1$.

(Setting 1) Next we discuss the first setting with lower bound $\Omega\left(\frac{\Delta L \sigma^2}{n B \epsilon^4}\right)$. (Setting 1, Step 1) The loss functions are defined as

$$\hat{f}_i(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}),$$

note that $1/n \sum_{i=1}^n \hat{f}_i = \hat{f}$. It can be seen from Property 2 that all the $\hat{f}_i$ are l_1-smooth. (Setting 1, Step 2) For this setting we consider complete graph. We construct the oracle on worker i as the following:

$$[\hat{g}_i(\boldsymbol{x})]_j = \nabla_j \hat{f}_i(\boldsymbol{x}) \cdot \left(1 + \mathbb{1}\{j > \text{prog}(\boldsymbol{x})\}\left(\frac{z}{p} - 1\right)\right),$$

where $z \sim \text{Bernoulli}(p)$. It can be seen that

$$\mathbb{E}[\hat{g}_i(\boldsymbol{x})] = \nabla \hat{f}_i(\boldsymbol{x}),$$

and from Property 3 we know

$$\begin{aligned}
\mathbb{E}\|\hat{g}_i(\boldsymbol{x}) - \nabla \hat{f}_i(\boldsymbol{x})\|^2 &= |\nabla_{\text{prog}(\boldsymbol{x})+1} \hat{f}(\boldsymbol{x})|^2 \mathbb{E}\left(\frac{z}{p} - 1\right)^2 \\
&\leq \frac{\|\nabla \hat{f}_i(\boldsymbol{x})\|_\infty^2 (1-p)}{p} \\
&\leq \frac{\|\nabla \hat{f}(\boldsymbol{x})\|_\infty^2 (1-p)}{p} \\
&\leq \frac{G_\infty^2 (1-p)}{p}.
\end{aligned}$$

(Setting 1, Step 3) Finally we rescale each function as $f_i = L\lambda^2/l_1 \hat{f}_i(\boldsymbol{x}/\lambda)$ where λ is a parameter subject to change. For L: note that all f_i are $\frac{L}{l_1} \cdot l_1 = L$-smooth. For the Δ,

$$f - f^* = \frac{L\lambda^2}{l_1}(\hat{f} - \hat{f}^*) = \frac{L\lambda^2 \Delta_0 T}{l_1} \leq \Delta. \tag{24}$$

For the oracle, to be consistent with f_i, we rescale it as $g_i(\boldsymbol{x}) = L\lambda/l_1 \hat{g}_i(\boldsymbol{x}/\lambda)$, and we have

$$\mathbb{E}\|g_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x})\|^2 \leq \frac{L^2 \lambda^2}{l_1^2} \mathbb{E}\left\|g_i\left(\frac{\boldsymbol{x}}{\lambda}\right) - \nabla f_i\left(\frac{\boldsymbol{x}}{\lambda}\right)\right\|^2 \leq \frac{L^2 \lambda^2 G_\infty^2 (1-p)}{l_1^2 p} \leq \sigma^2. \tag{25}$$
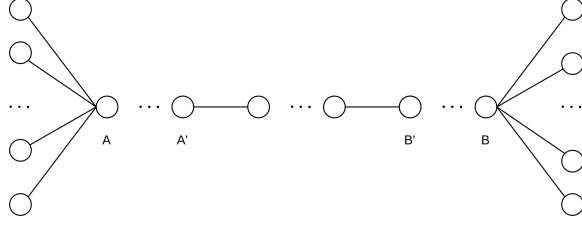
Figure 6: Illustration graph for setting 2 to in the proof of Theorem 1.

We assign $\lambda = 2l_1\epsilon/L$, then Equation (24) and (25) are fulfilled with

$$T = \left\lfloor \frac{\Delta}{\Delta_0 l_1 (2\epsilon)^2} \right\rfloor,$$
$$p = \min\{(2G_\infty \epsilon)^2/\sigma^2, 1\}.$$

Take $\delta = 1/2$ in Lemma 8, we have for probability at least $1/2$, $\|\nabla f(\hat{\boldsymbol{x}}^{(t)})\| \geq \epsilon$ for all $t \leq \frac{T+\log(\delta)}{\min\{nBp,1\}(e-1)}$. Use Property 4, for any $\boldsymbol{x} \in \mathbb{R}^T$ such that $\text{prog}(\boldsymbol{x}) < T$ it holds that $\|\nabla f(\boldsymbol{x})\| \geq 2\epsilon$, therefore,

$$\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_T)\| > \epsilon.$$

Then with small ϵ it follows that

$$T_\epsilon(A, f, O, G) \geq \frac{T-1}{nBp(e-1)} \geq \Omega\left(\frac{\Delta L \sigma^2}{nB\epsilon^4}\right),$$

and that completes the proof for setting 1.

(Setting 2) We proceed to the prove second bound $\Omega\left(\frac{\Delta L D}{\epsilon^2}\right)$.

(Setting 2 Step 1 & Step 2) We assign all the workers with index from 1 to n, we first define two indices set

$$I_0 = \{1, \cdots, |I_0|\},$$
$$I_1 = \{n, n-1, \cdots, n-|I_1|+1\}.$$

where $|\cdot|$ denotes a cardinality of a set. Consider the construction of G in Figure 6:

If $D \geq n - 2\lceil n/3 \rceil + 2$, then it implies the number of nodes between A and B is larger than $\lceil n/3 \rceil$. In this case, denote A' and B' as a sub linear graph where its number of nodes is exactly $\lceil n/3 \rceil$. Let all the nodes on the left of A' be in I_0 and all the nodes on the right of B' be I_1 We define all the local functions $\hat{f}_i(\boldsymbol{x})$ on such graph as following: when $i \in I_0$,

$$\hat{f}_i(\boldsymbol{x}) = -\frac{2n}{n - \lceil n/3 \rceil} \Psi(1)\Phi(\boldsymbol{x}^{(1)})$$
$$+ \sum_{i=2k, k\in\{1,2,\cdots\}, i<T} \frac{2n}{n - \lceil n/3 \rceil} [\Psi(-\boldsymbol{x}^{(i)})\Phi(-\boldsymbol{x}^{(i+1)}) - \Psi(\boldsymbol{x}^{(i)})\Phi(\boldsymbol{x}^{(i+1)})];$$

when $i \in I_1$,

$$\hat{f}_i(\boldsymbol{x}) = \sum_{i=2k-1, k\in\{1,2,\cdots\}, i<T} \frac{2n}{n-\lceil n/3\rceil}[\Psi(-\boldsymbol{x}^{(i)})\Phi(-\boldsymbol{x}^{(i+1)}) - \Psi(\boldsymbol{x}^{(i)})\Phi(\boldsymbol{x}^{(i+1)})];$$

when $i \notin I_0, I_1$,

$$\hat{f}_i(\boldsymbol{x}) = 0.$$

If $D < n - 2\lceil n/3\rceil + 2$, the distance between node A and node B is $D - 2$ and the sub linear graph whose end points are A and B contains $D - 1$ nodes. We let the number of nodes on the left of A be $\lceil \frac{n-D+1}{2}\rceil$, we denote the set of indices of all such nodes as I_0; and then we let the number of nodes on the right of B be $\lfloor \frac{n-D+1}{2}\rfloor$, we denote the set of indices of all such nodes as I_I. Since $D < n - 2\lceil n/3\rceil + 2$, this implies $|I_0|, |I_1| > n/3$. We define all the local functions on such graph as following:
when $i \in I_0$,

$$\hat{f}_i(\boldsymbol{x}) = -\frac{n}{|I_0|}\Psi(1)\Phi(\boldsymbol{x}^{(1)})$$
$$+ \sum_{i=2k, k\in\{1,2,\cdots\}, i<T} \frac{n}{|I_0|}[\Psi(-\boldsymbol{x}^{(i)})\Phi(-\boldsymbol{x}^{(i+1)}) - \Psi(\boldsymbol{x}^{(i)})\Phi(\boldsymbol{x}^{(i+1)})];$$

when $i \in I_1$,

$$\hat{f}_i(\boldsymbol{x}) = \sum_{i=2k-1, k\in\{1,2,\cdots\}, i<T} \frac{n}{|I_1|}[\Psi(-\boldsymbol{x}^{(i)})\Phi(-\boldsymbol{x}^{(i+1)}) - \Psi(\boldsymbol{x}^{(i)})\Phi(\boldsymbol{x}^{(i+1)})];$$

when $i \notin I_0, I_1$,

$$\hat{f}_i(\boldsymbol{x}) = 0.$$

In both cases discussed based on D, we can see that $\hat{f}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^n \hat{f}_i(\boldsymbol{x})$, and we are splitting hard zero-chain function into two main different part: the even components of the chain and the odd components of the chain. It is easy to see that for the zero chain function to make progress, it takes at least $\lceil n/3\rceil$, i.e., $\Omega(D)$ number of iterations in the first case (since here $D = \tilde{\gamma}n$ for some $\tilde{\gamma} > 1/3$) and D number of iterations in the seconds case. Then the total number of iterations is lower bounded by $\Omega(TD)$.

For the oracle, we let oracle on worker i as

$$[\hat{g}_i(\boldsymbol{x})]_j = \nabla_j \hat{f}_i(\boldsymbol{x}).$$

(Setting 2, Step 3) The last step is to rescale the parameters. Compared to setting 1, we know here all the $\hat{f}_i$ are $3l_1$-smooth, as before we let

$$f_i(\boldsymbol{x}) = \frac{L\lambda^2}{3l_1}\hat{f}_i\left(\frac{\boldsymbol{x}}{\lambda}\right), \qquad \lambda = \frac{6l_1\epsilon}{L}.$$

For the Δ bound we have

$$L\lambda^2\Delta_0 T/3l_1 \leq \Delta$$

33

to fulfill this it suffices to set

$$T = \left\lfloor \frac{\Delta L}{\Delta_0 l_1 (12\epsilon)^2} \right\rfloor.$$

It also can be seen that f is L-smooth. So in this setting,

$$T_\epsilon(A, f, O, G) \geq \Omega(TD) = \Omega\left(\frac{\Delta L D}{\epsilon^2}\right).$$

Combining Setting 1 and 2 we complete the proof. ∎

Lemma 8 *In setting 1 in the proof of Theorem 1, with probability at least $1-\delta$, $\|\nabla f(\boldsymbol{x}_t)\| \geq \epsilon$ for all $t \leq \frac{T+\log(\delta)}{\min\{nBp,1\}(e-1)}$.*

Proof Define a filtration at iteration t as the sigma field of all the previous events happened before iteration t. Let $i_j^{(t)} = \text{prog}(\boldsymbol{x}_{t,j}), \forall j \in [n]$ and $i^{(t)} = \max_j i_j^{(t)}$. And we denote $\mathcal{E}^{(t,m,j)}$ as the event of the $i_m^{(t)} + 1$-th coordinate of output of j-th query on worker m at iteration t is non-zero. Based on the independent sampling, these events are independent. Thus we know:

$$\mathbb{P}[i^{(t+1)} - i^{(t)} = 1 | \mathcal{U}^{(t)}] = \mathbb{P}\left[\bigcup_{\substack{i \in [n] \\ j \leq B}} \mathcal{E}^{(t,i,j)} | \mathcal{U}^{(t)}\right] \leq \sum_{i \in [n], j \leq B} \mathbb{P}\left[\mathcal{E}^{(t,i,j)} | \mathcal{U}^{(t)}\right] \leq \min\{nBp, 1\}.$$

Let $q^{(t)} = i^{(t+1)} - i^{(t)}$, with Chernoff bound, we obtain

$$\mathbb{P}[i^{(t)} \geq T] = \mathbb{P}[e^{\sum_{j=0}^{t-1} q^{(j)}} \geq e^T] \leq e^{-T} \mathbb{E}[e^{\sum_{j=0}^{t-1} q^{(j)}}].$$

For the expectation term we know that

$$\mathbb{E}[e^{\sum_{j=0}^{t-1} q^{(j)}}] = \mathbb{E}\left[\prod_{j=0}^{t-1} \mathbb{E}\left[e^{q^{(j)}} | \mathcal{U}^{(j)}\right]\right] \leq (1 - \min\{nBp, 1\} + \min\{nBp, 1\}e)^t \leq e^{\min\{nBp,1\}t(e-1)}.$$

Thus we know

$$\mathbb{P}[i^{(t)} \geq T] \leq e^{(e-1)\min\{nBp,1\}t - T} \leq \delta,$$

for every $t \leq \frac{T+\log(\delta)}{\min\{nBp,1\}(e-1)}$. ∎

B.2 Proof to Corollary 2

Proof Different from Theorem 5, in this corollary we do not choose D and n separately, so that our construction can just use the linear graph as follows:

(Linear graph, Step 1) We first let $|I_0| = |I_1| = \lceil n/3 \rceil$ in the proof of Theorem 1, meaning I_0 denotes the first $\lceil n/3 \rceil$ workers and I_1 denotes the last $\lceil n/3 \rceil$ workers. We define all the local functions $\hat{f}_i(\boldsymbol{x})$ as following:

when $i \in I_0$,

$$\hat{f}_i(\boldsymbol{x}) = -\frac{n}{\lceil n/3 \rceil} \Psi(1) \Phi(\boldsymbol{x}^{(1)})$$
$$+ \sum_{i=2k, k \in \{1,2,\cdots\}, i<T} \frac{n}{\lceil n/3 \rceil} [\Psi(-\boldsymbol{x}^{(i)}) \Phi(-\boldsymbol{x}^{(i+1)}) - \Psi(\boldsymbol{x}^{(i)}) \Phi(\boldsymbol{x}^{(i+1)})];$$

when $i \in I_1$,

$$\hat{f}_i(\boldsymbol{x}) = \sum_{i=2k-1, k \in \{1,2,\cdots\}, i<T} \frac{n}{\lceil n/3 \rceil} [\Psi(-\boldsymbol{x}^{(i)}) \Phi(-\boldsymbol{x}^{(i+1)}) - \Psi(\boldsymbol{x}^{(i)}) \Phi(\boldsymbol{x}^{(i+1)})];$$

when $i \notin I_0, I_1$,

$$\hat{f}_i(\boldsymbol{x}) = 0.$$

We can see that $\hat{f}(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} \hat{f}_i(\boldsymbol{x})$. (Linear graph, Step 2) We consider linear graph in this setting and from one end to the other, the worker's index is 1 to n, without the loss of generality. It is easy to see that for the zero chain function to make progress, it takes at least $n - 2\lceil n/3 \rceil + 1$ number of iterations. Note that in linear graph $n - 1 = D$, the total number of iterations is at least

$$\Omega\left(TD\right).$$

For the oracle, we let oracle on worker i as

$$[\hat{g}_i(\boldsymbol{x})]_j = \nabla_j \hat{f}_i(\boldsymbol{x})$$

(Linear graph, Step 3) The last step is to rescale the parameters. Compared to setting 1, we know here all the $\hat{f}_i$ are $3l_1$-smooth, as before we let

$$f_i(\boldsymbol{x}) = \frac{L\lambda^2}{3l_1} \hat{f}_i\left(\frac{\boldsymbol{x}}{\lambda}\right), \qquad \lambda = \frac{6l_1\epsilon}{L}.$$

For the Δ bound we have

$$L\lambda^2 \Delta_0 T / 3l_1 \leq \Delta,$$

to fulfill this it suffices to set

$$T = \left\lfloor \frac{\Delta L}{\Delta_0 l_1 (12\epsilon)^2} \right\rfloor.$$

It also can be seen that f is L-smooth. So in this setting,

$$T_\epsilon(A, f, O, G) \geq \Omega(TD) \geq \Omega\left(\frac{\Delta LD}{\epsilon^2}\right).$$

Given the bound, we use two additional results on linear graph as Berthier et al. (2020): the random walk matrix $\boldsymbol{W}_{rw}$ on linear graph with λ fulfilling

$$\frac{1}{\sqrt{1-\lambda}} = O(D).$$

Then we can rewrite the lower bound in the form of λ as shown in Corollary 2.

Finally, using the conclusion of $\lambda = \cos(\pi/n)$ for $n \in \{2, 3, \cdots, \}$ on linear graph we complete the proof.

∎

B.3 Proof to Theorem 4

Proof As (partially) discussed in the paper, DeFacto is statistically equivalent to centralized SGD. Specifically, it conduct $K = T/2R$ gradient steps where each step contains a mini-batch of R at the point of $\boldsymbol{x}_{k,i}, \forall i \in [n]$. Take the well-known convergence rate for centralized SGD:

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla f(\hat{\boldsymbol{x}})\|^2 \leq O\left(\frac{\Delta L\sigma}{\sqrt{nBT}} + \frac{\Delta L}{T}\right).$$

The convergence rate of DeFacto can be expressed as:

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla f(\hat{\boldsymbol{x}})\|^2 \leq O\left(\frac{\Delta L\sigma/\sqrt{R}}{\sqrt{nBK}} + \frac{\Delta L}{K}\right) = O\left(\frac{\Delta L\sigma}{\sqrt{nBT}} + \frac{\Delta LR}{T}\right) = O\left(\frac{\Delta L\sigma}{\sqrt{nBT}} + \frac{\Delta LD}{T}\right),$$

then we obtain for DeFacto, when $T = O(\Delta L\sigma^2(nB\epsilon^4)^{-1} + \Delta LD\epsilon^{-2})$,

$$\min_{t=0,1,\cdots,T-1}\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}})\| \leq \sqrt{\min_{t=0,1,\cdots,T-1}\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}})\|^2} \leq \sqrt{O\left(\frac{\Delta L\sigma}{\sqrt{nBT}} + \frac{\Delta LD}{T}\right)} \leq \epsilon,$$

that completes the proof.

∎

B.4 Proof to Theorem 5

Proof In this proof, we adopt an updated version of notation: we denote at the beginning of phase k ($k \geq 0$), the three quantities of interests are $\boldsymbol{X}_k$, $\boldsymbol{Y}_k$ and $\tilde{\boldsymbol{G}}_k$, and the update rule becomes (for any $k \geq 0$ with $\tilde{\boldsymbol{G}}_{-1} = \boldsymbol{0}$ by convention and $\boldsymbol{X}_0 = \boldsymbol{0}$, $\boldsymbol{Y}_0 = \boldsymbol{0}$ and $\tilde{\boldsymbol{G}}_0 = \boldsymbol{0}$ following initialization):

$$\boldsymbol{Y}_{k+1} = \mathcal{M}(\boldsymbol{Y}_k + \tilde{\boldsymbol{G}}_k - \tilde{\boldsymbol{G}}_{k-1}),$$

$$\boldsymbol{X}_{k+1} = \mathcal{M}(\boldsymbol{X}_k - \alpha_k \boldsymbol{Y}_k),$$

with

$$\tilde{\boldsymbol{G}}_{k+1} = \left[\nabla \tilde{f}_1(\boldsymbol{x}_{k,1}), \cdots, \nabla \tilde{f}_n(\boldsymbol{x}_{k,n})\right] \in \mathbb{R}^{d \times n},$$

$$\boldsymbol{G}_{k+1} = [\nabla f_1(\boldsymbol{x}_{k,1}), \cdots, \nabla f_n(\boldsymbol{x}_{k,n})] \in \mathbb{R}^{d \times n},$$

$$\boldsymbol{X}_k = [\boldsymbol{x}_{k,1}, \cdots, \boldsymbol{x}_{k,n}] \in \mathbb{R}^{d \times n},$$

$$\boldsymbol{Y}_k = \left[\boldsymbol{y}_{k,1}, \cdots, \boldsymbol{y}_{k,n}\right] \in \mathbb{R}^{d \times n},$$

where $\nabla \tilde{f}_i$ denotes the stochastic gradient oracle on worker i, and ∇f_i denotes the full gradient oracle on worker i. We use $\overline{\boldsymbol{X}}$ denote $\boldsymbol{X}\frac{\boldsymbol{1}}{n}$ for any matrix $\boldsymbol{X}$ with appropriate shape. We use $\lambda_i(\boldsymbol{W})$ to denote the i-th general largest eigenvalue of matrix $\boldsymbol{W}$. Under such notation, λ in the main paper is equavilent to $\lambda_2(\boldsymbol{W})$. We use $\mathcal{M}(\cdot)$ to denote the R-step accelerated gossip which has the following property Liu and Morse (2011):

$$\|\mathcal{M}(\boldsymbol{X}) - \overline{\boldsymbol{X}}\| \le \rho \|\boldsymbol{X} - \overline{\boldsymbol{X}}\|; \quad \mathcal{M}(\boldsymbol{X})\frac{\boldsymbol{1}}{n} = \boldsymbol{X}\frac{\boldsymbol{1}}{n}, \tag{26}$$

where $\rho = \left(1 - \sqrt{1 - \lambda_2(\boldsymbol{W})}\right)^R$. The proof to the statement of Equation (26) can be found in Ye et al. (2020).

For the stochastic oracle, based on the oracle class assumption, we have

$$\mathbb{E}\|\nabla \tilde{f}_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x})\|^2 \le \sigma^2,$$

and we denote $\tilde{\sigma}^2 = \frac{\sigma^2}{BR}$ as the variance of mini-batch of $B \cdot R$.

First, from the update rule of DeTAG,

$$\overline{\boldsymbol{Y}}_k = \mathcal{M}(\boldsymbol{Y}_{k-1} + \tilde{\boldsymbol{G}}_{k-1} - \tilde{\boldsymbol{G}}_{k-2})\frac{\boldsymbol{1}}{n} = \overline{\boldsymbol{Y}}_{k-1} + \overline{\tilde{\boldsymbol{G}}}_{k-1} - \overline{\tilde{\boldsymbol{G}}}_{k-2} = \overline{\boldsymbol{Y}}_0 + \sum_{j=0}^{k-1}(\overline{\tilde{\boldsymbol{G}}}_j - \overline{\tilde{\boldsymbol{G}}}_{j-1}) = \overline{\tilde{\boldsymbol{G}}}_{k-1} \tag{27}$$

and

$$\overline{\boldsymbol{X}}_{k+1} = \mathcal{M}(\boldsymbol{X}_k - \alpha_k \boldsymbol{Y}_k)\frac{\boldsymbol{1}}{n} = \overline{\boldsymbol{X}}_k - \alpha_k \overline{\boldsymbol{Y}}_k. \tag{28}$$

By Taylor Theorem, we obtain for any $k \ge 0$

$$\begin{aligned}
\mathbb{E}f\left(\overline{\boldsymbol{X}}_{k+1}\right) =& \mathbb{E}f\left(\overline{\boldsymbol{X}}_k - \alpha_k \overline{\boldsymbol{Y}}_k\right) \\
\le& \mathbb{E}f\left(\overline{\boldsymbol{X}}_k\right) - \alpha_k \mathbb{E}\left\langle \nabla f\left(\overline{\boldsymbol{X}}_k\right), \overline{\boldsymbol{Y}}_k \right\rangle + \frac{\alpha_k^2 L}{2}\mathbb{E}\left\|\overline{\boldsymbol{Y}}_k\right\|^2 \\
\overset{(27)}{=}& \mathbb{E}f\left(\overline{\boldsymbol{X}}_k\right) - \alpha_k \mathbb{E}\left\langle \nabla f\left(\overline{\boldsymbol{X}}_k\right), \overline{\boldsymbol{G}}_{k-1} \right\rangle + \frac{\alpha_k^2 L}{2}\mathbb{E}\left\|\overline{\tilde{\boldsymbol{G}}}_{k-1}\right\|^2.
\end{aligned}$$

37

For the last term, we have

$$
\begin{aligned}
\mathbb{E}\left\|\overline{\tilde{\boldsymbol{G}}}_{k-1}\right\|^2 =& \mathbb{E}\left\|\overline{\boldsymbol{G}}_{k-1}\right\|^2 + \mathbb{E}\left\|\overline{\boldsymbol{G}}_{k-1} - \overline{\tilde{\boldsymbol{G}}}_{k-1}\right\|^2 + 2\mathbb{E}\left\langle\overline{\boldsymbol{G}}_{k-1}, \overline{\boldsymbol{G}}_{k-1} - \overline{\tilde{\boldsymbol{G}}}_{k-1}\right\rangle \\
=& \mathbb{E}\left\|\overline{\boldsymbol{G}}_{k-1}\right\|^2 + \mathbb{E}\left\|\overline{\boldsymbol{G}}_{k-1} - \overline{\tilde{\boldsymbol{G}}}_{k-1}\right\|^2 \\
=& \mathbb{E}\left\|\overline{\boldsymbol{G}}_{k-1}\right\|^2 + \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left\|\boldsymbol{G}_{k-1}\boldsymbol{e}_i - \tilde{\boldsymbol{G}}_{k-1}\boldsymbol{e}_i\right\|^2 \\
\leq& \mathbb{E}\left\|\overline{\boldsymbol{G}}_{k-1}\right\|^2 + \frac{\tilde{\sigma}^2}{n},
\end{aligned}
\tag{29}
$$

where in the second step, we use the fact that the sampling noise is independent of the gradient itself. Note that this inequality still holds when $k = 0, -1$ as long as we slightly abuse the notations and define $\boldsymbol{G}_0 = \boldsymbol{G}_{-1} = \boldsymbol{0}$. Putting it back we obtain

$$
\begin{aligned}
&\mathbb{E}f\left(\overline{\boldsymbol{X}}_{k+1}\right) \\
\leq& \mathbb{E}f\left(\overline{\boldsymbol{X}}_k\right) - \alpha_k\mathbb{E}\left\langle\nabla f\left(\overline{\boldsymbol{X}}_k\right), \overline{\boldsymbol{G}}_{k-1}\right\rangle + \frac{\alpha_k^2 L}{2}\mathbb{E}\left\|\overline{\boldsymbol{G}}_{k-1}\right\|^2 + \frac{\alpha_k^2\tilde{\sigma}^2 L}{2n} \\
=& \mathbb{E}f\left(\overline{\boldsymbol{X}}_k\right) - \frac{\alpha_k}{2}\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{X}}_k\right)\right\|^2 - \frac{\alpha_k - \alpha_k^2 L}{2}\left\|\overline{\boldsymbol{G}}_{k-1}\right\|^2 + \frac{\alpha_k^2\tilde{\sigma}^2 L}{2n} + \frac{\alpha_k}{2}\mathbb{E}\left\|\overline{\boldsymbol{G}}_{k-1} - \nabla f\left(\overline{\boldsymbol{X}}_k\right)\right\|^2,
\end{aligned}
$$

where the last step we use the fact that for any $\boldsymbol{a}$ and $\boldsymbol{b}$, $2\langle\boldsymbol{a}, \boldsymbol{b}\rangle = \|\boldsymbol{a}\|^2 + \|\boldsymbol{b}\|^2 - \|\boldsymbol{a} - \boldsymbol{b}\|^2$. Expand the last term,

$$
\begin{aligned}
&\mathbb{E}\left\|\overline{\boldsymbol{G}}_{k-1} - \nabla f\left(\overline{\boldsymbol{X}}_k\right)\right\|^2 \\
\leq& 2\mathbb{E}\left\|\overline{\boldsymbol{G}}_{k-1} - \overline{\boldsymbol{G}}_{k+1}\right\|^2 + 2\mathbb{E}\left\|\overline{\boldsymbol{G}}_{k+1} - \nabla f\left(\overline{\boldsymbol{X}}_k\right)\right\|^2 \\
=& 2\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{x}_{k,i}) - \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{x}_{k-2,i})\right\|^2 + 2\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{x}_{k,i}) - \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\overline{\boldsymbol{X}}_k)\right\|^2 \\
\leq& \frac{2}{n}\sum_{i=1}^{n}\mathbb{E}\left\|\nabla f_i(\boldsymbol{x}_{k,i}) - \nabla f_i(\boldsymbol{x}_{k-2,i})\right\|^2 + \frac{2}{n}\sum_{i=1}^{n}\mathbb{E}\left\|\nabla f_i(\boldsymbol{x}_{k,i}) - \nabla f_i(\overline{\boldsymbol{X}}_k)\right\|^2 \\
\leq& \frac{2L^2}{n}\mathbb{E}\left\|\boldsymbol{X}_k - \boldsymbol{X}_{k-2}\right\|_F^2 + \frac{2L^2}{n}\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\boldsymbol{1}_n^\top\right\|_F^2,
\end{aligned}
$$

where in the third step we use the Jensen's Inequality and in the fourth step we apply the smoothness assumption.

With the assumption of $f(\boldsymbol{0}) - f^* \leq \Delta$, we obtain

$$
\begin{aligned}
&\sum_{k=0}^{K-1}\alpha_k(1 - \alpha_k L)\left\|\overline{\boldsymbol{G}}_k\right\|^2 + \sum_{k=0}^{K-1}\alpha_k\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{X}}_k\right)\right\|^2 \\
\leq& 2\Delta + \frac{\tilde{\sigma}^2 L}{n}\sum_{k=0}^{K-1}\alpha_k^2 + \frac{2L^2}{n}\sum_{k=0}^{K-1}\alpha_k\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\boldsymbol{1}_n^\top\right\|_F^2 + \frac{2L^2}{n}\sum_{k=0}^{K-1}\alpha_k\mathbb{E}\left\|\boldsymbol{X}_k - \boldsymbol{X}_{k-2}\right\|_F^2 \\
\leq& 2\Delta + \frac{\tilde{\sigma}^2 L}{n}\sum_{k=0}^{K-1}\alpha_k^2 + \frac{16L^2}{n}\sum_{k=0}^{K-1}\alpha_k\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\boldsymbol{1}_n^\top\right\|_F^2 + \frac{6L^2}{n}\sum_{k=0}^{K-1}\alpha_k\mathbb{E}\left\|\overline{\boldsymbol{X}}_k\boldsymbol{1}_n^\top - \overline{\boldsymbol{X}}_{k-2}\boldsymbol{1}_n^\top\right\|_F^2,
\end{aligned}
$$

where in the last step we use

$$\mathbb{E}\left\|\boldsymbol{X}_k - \boldsymbol{X}_{k-2}\right\|_F^2 \leq 3\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2 + 3\mathbb{E}\left\|\boldsymbol{X}_{k-2} - \overline{\boldsymbol{X}}_{k-2}\mathbf{1}_n^\top\right\|_F^2 + 3\mathbb{E}\left\|\overline{\boldsymbol{X}}_k\mathbf{1}_n^\top - \overline{\boldsymbol{X}}_{k-2}\mathbf{1}_n^\top\right\|_F^2.$$

In addition, for the last term we have

$$
\begin{aligned}
&\frac{6L^2}{n}\sum_{k=0}^{K-1}\alpha_k\mathbb{E}\left\|\overline{\boldsymbol{X}}_k\mathbf{1}_n^\top - \overline{\boldsymbol{X}}_{k-2}\mathbf{1}_n^\top\right\|_F^2 \\
=&\frac{6L^2n}{n}\sum_{k=0}^{K-1}\alpha_k\mathbb{E}\left\|\overline{\boldsymbol{X}}_k - \overline{\boldsymbol{X}}_{k-2}\right\|^2 \\
\leq&\frac{6L^2n}{n}\sum_{k=0}^{K-1}\alpha_k\left(2\mathbb{E}\left\|\overline{\boldsymbol{X}}_k - \overline{\boldsymbol{X}}_{k-1}\right\|^2 + 2\mathbb{E}\left\|\overline{\boldsymbol{X}}_{k-1} - \overline{\boldsymbol{X}}_{k-2}\right\|^2\right) \\
\overset{(28)}{=}&\frac{24L^2n}{n}\sum_{k=0}^{K-1}\alpha_k^3\mathbb{E}\left\|\overline{\overline{\boldsymbol{G}}}_k\right\|^2 \\
\overset{(29)}{\leq}&24L^2\sum_{k=0}^{K-1}\alpha_k^3\mathbb{E}\left\|\overline{\boldsymbol{G}}_k\right\|^2 + \frac{24\tilde{\sigma}^2L^2}{n}\sum_{k=0}^{K-1}\alpha_k^3,
\end{aligned}
$$

where in the third step we use the fact that $\{\alpha_k\}$ is non-increasing with respect to k. Push it back we have

$$
\begin{aligned}
&\sum_{k=0}^{K-1}\alpha_k(1 - \alpha_kL - 24\alpha_k^2L^2)\left\|\overline{\boldsymbol{G}}_k\right\|^2 + \sum_{k=0}^{K-1}\alpha_k\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{X}}_k\right)\right\|^2 \\
\leq&2\Delta + \frac{\tilde{\sigma}^2L}{n}\sum_{k=0}^{K-1}\alpha_k^2 + \frac{16L^2}{n}\sum_{k=0}^{K-1}\alpha_k\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2 + \frac{24\tilde{\sigma}^2L^2}{n}\sum_{k=0}^{K-1}\alpha_k^3.
\end{aligned}
\tag{30}
$$

The rest of the proof is to bound $\frac{16L^2}{n}\sum_{k=0}^{K-1}\alpha_k\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2$.

First from Lemma 9 we obtain the dynamics of $\boldsymbol{X}_k$ and $\boldsymbol{Y}_k$ are as follows:

$$
\begin{aligned}
&\mathbb{E}\left\|\boldsymbol{X}_{k+1} - \overline{\boldsymbol{X}}_{k+1}\mathbf{1}_n^\top\right\|_F^2 \\
\leq&\frac{2\rho^2}{(1+\rho^2)}\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2 + \frac{2\rho^2\alpha_k^2}{1-\rho^2}\mathbb{E}\left\|\boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k\mathbf{1}_n^\top\right\|_F^2 \\
&\mathbb{E}\left\|\boldsymbol{Y}_{k+1} - \overline{\boldsymbol{Y}}_{k+1}\mathbf{1}_n^\top\right\|_F^2 \\
\leq&\left(\frac{2\rho^2}{1+\rho^2} + \frac{16\alpha_k^2\rho^4L^2}{1-\rho^2}\right)\mathbb{E}\left\|\boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k\mathbf{1}_n^\top\right\|_F^2 + \frac{16\rho^2(1+\rho^2)L^2}{1-\rho^2}\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2 \\
&+\frac{16\alpha_k^2\rho^2nL^2}{1-\rho^2}\mathbb{E}\left\|\overline{\boldsymbol{Y}}_k\right\|^2 + \frac{4\rho^2}{1-\rho^2}\mathbb{E}\left\|\boldsymbol{G}_{k+2} - \boldsymbol{G}_{k+1} - \boldsymbol{G}_k + \boldsymbol{G}_{k-1}\right\|_F^2 + 8n\rho^2\tilde{\sigma}^2.
\end{aligned}
\tag{31}
$$

39

The dynamics of $\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2$ and $\left\|\boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k\mathbf{1}_n^\top\right\|_F^2$ follow a linear system. By solving this linear system (details in Lemma 10), we obtain the following:

$$
\sum_{k=0}^{K-1} \alpha_k \mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2
$$

$$
\leq \frac{2\rho^2\alpha_0^3(1+\rho)nK\varsigma_0^2}{(1-\rho)(1-\sqrt{\rho})^2} + \frac{32\rho^4 nL^2}{(1-\rho)^2(1-\sqrt{\rho})^2}\sum_{k=0}^{K-1}\alpha_k^5\mathbb{E}\left\|\overline{\boldsymbol{G}}_k\right\|^2 + \frac{8\rho^4}{(1-\rho)^2(1-\sqrt{\rho})^2}\sum_{k=0}^{K-1}\alpha_k^3\mathbb{E}\|\boldsymbol{U}_k\|_F^2
$$

$$
+ \frac{16\rho^4 n\tilde{\sigma}^2(1+\rho)}{(1-\rho)(1-\sqrt{\rho})^2}\sum_{k=0}^{K-1}\alpha_k^3,
$$

$$(32)$$

where $\boldsymbol{U}_k = \boldsymbol{G}_{k+2} - \boldsymbol{G}_{k+1} - \boldsymbol{G}_k + \boldsymbol{G}_{k-1}$. From here we need to additionally bound $\sum_{k=0}^{K-1}\alpha_k^3\mathbb{E}\|\boldsymbol{U}_k\|_F^2$.

Start by its definition,

$$
\sum_{k=0}^{K-1}\alpha_k^3\mathbb{E}\left\|\boldsymbol{U}_k\right\|_F^2
$$

$$
= \sum_{k=0}^{K-1}\alpha_k^3\mathbb{E}\left\|\boldsymbol{G}_{k+2} - \boldsymbol{G}_{k+1} - \boldsymbol{G}_k + \boldsymbol{G}_{k-1}\right\|_F^2
$$

$$
\leq 2\sum_{k=0}^{K-1}\alpha_k^3\mathbb{E}\left\|\boldsymbol{G}_{k+2} - \boldsymbol{G}_{k+1}\right\|_F^2 + 2\sum_{k=0}^{K-1}\alpha_k^3\mathbb{E}\left\|\boldsymbol{G}_k - \boldsymbol{G}_{k-1}\right\|_F^2
$$

$$
\leq 4\sum_{k=0}^{K-1}\alpha_k^3\mathbb{E}\left\|\boldsymbol{G}_{k+2} - \boldsymbol{G}_{k+1}\right\|_F^2
$$

$$
= 4\sum_{k=0}^{K-1}\alpha_k^3\sum_{i=1}^{n}\mathbb{E}\left\|\nabla f(\boldsymbol{x}_{k+1,i}) - \nabla f(\boldsymbol{x}_{k,i})\right\|^2
$$

$$
\leq 4L^2\sum_{k=0}^{K-1}\alpha_k^3\sum_{i=1}^{n}\mathbb{E}\left\|\boldsymbol{x}_{k+1,i} - \boldsymbol{x}_{k,i}\right\|^2
$$

$$
= 4L^2\sum_{k=0}^{K-1}\alpha_k^3\mathbb{E}\left\|\boldsymbol{X}_{k+1} - \boldsymbol{X}_k\right\|_F^2
$$

$$
\overset{(28)}{=} 4L^2\sum_{k=0}^{K-1}\alpha_k^3\mathbb{E}\left\|\mathcal{M}(\boldsymbol{X}_k) - \boldsymbol{X}_k - \alpha_k\mathcal{M}(\boldsymbol{Y}_k)\right\|_F^2
$$

$$
= 4L^2\sum_{k=0}^{K-1}\alpha_k^3\mathbb{E}\left\|\mathcal{M}(\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top) - (\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top) - \alpha_k\mathcal{M}(\boldsymbol{Y}_k)\right\|_F^2
$$

$$
\leq 4L^2\sum_{k=0}^{K-1}\alpha_k^3\cdot
$$

$$
\left(4\mathbb{E}\left\|\mathcal{M}(\boldsymbol{X}_k) - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2 + 4\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2 + 4\alpha_k^2\mathbb{E}\left\|\mathcal{M}(\boldsymbol{Y}_k) - \overline{\boldsymbol{Y}}_k\mathbf{1}_n^\top\right\|_F^2 + 4\alpha_k^2 n\mathbb{E}\left\|\overline{\boldsymbol{Y}}_k\right\|^2\right)
$$

$$\leq 16(1+\rho^2)L^2 \sum_{k=0}^{K-1} \alpha_k^3 \mathbb{E} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2 + 16\rho^2 L^2 \sum_{k=0}^{K-1} \alpha_k^5 \mathbb{E} \left\| \boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \right\|_F^2$$

$$+ 16nL^2 \sum_{k=0}^{K-1} \alpha_k^5 \mathbb{E} \left\| \overline{\boldsymbol{Y}}_k \right\|^2.$$

The bound depends both on $\boldsymbol{X}_k$ and $\boldsymbol{Y}_k$. Revisit Lemma 10, we obtain the following two bounds:

$$\sum_{k=0}^{K-1} \alpha_k^3 \mathbb{E} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2$$

$$\leq \frac{2\rho^2 \alpha_0^5 (1+\rho)nK\varsigma_0^2}{(1-\rho)(1-\sqrt{\rho})^2} + \frac{32\rho^4 nL^2}{(1-\rho)^2(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^7 \mathbb{E} \left\| \overline{\boldsymbol{G}}_k \right\|^2$$

$$+ \frac{8\rho^4}{(1-\rho)^2(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^5 \mathbb{E} \|\boldsymbol{U}_k\|_F^2 + \frac{16\rho^4 n\tilde{\sigma}^2(1+\rho)}{(1-\rho)(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^5,$$

and

$$\sum_{k=0}^{K-1} \alpha_k^5 \mathbb{E} \| \boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \|^2$$

$$\leq \frac{(1+\rho)\alpha_0^5 nK\varsigma_0^2}{1-\sqrt{\rho}} + \frac{8\alpha_0^7 \rho^4(1+\rho)L^2 nK\varsigma_0^2}{(1-\rho)(1-\sqrt{\rho})^2} + \sum_{k=0}^{K-1} \alpha_k^5 \left(\frac{1+\rho}{1-\sqrt{\rho}} + \frac{8\alpha_k^2 \rho^4(1+\rho)L^2}{(1-\rho)(1-\sqrt{\rho})^2} \right) \cdot$$

$$\left(\frac{4\rho^2}{1-\rho^2} \mathbb{E} \|\boldsymbol{U}_k\|_F^2 + \frac{16\alpha_k^2 \rho^2 nL^2}{1-\rho^2} \mathbb{E} \left\| \overline{\boldsymbol{G}}_k \right\|^2 + \frac{16\alpha_k^2 \rho^2 \tilde{\sigma}^2 L^2}{1-\rho^2} + 8n\rho^2\tilde{\sigma}^2 \right).$$

Put both bounds back,

$$\sum_{k=0}^{K-1} \alpha_k^3 \mathbb{E} \|\boldsymbol{U}_k\|_F^2 \leq \frac{32\rho^2 \alpha_0^5 (1+\rho)^2 nK\varsigma_0^2 L^2}{(1-\rho)(1-\sqrt{\rho})^2} + \frac{512\rho^4(1+\rho)nL^4}{(1-\rho)^2(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^7 \mathbb{E} \left\| \overline{\boldsymbol{G}}_k \right\|^2$$

$$+ \frac{256\rho^4(1+\rho)L^2}{(1-\rho)^2(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^5 \mathbb{E} \|\boldsymbol{U}_k\|_F^2 + \frac{256\rho^4 n\tilde{\sigma}^2(1+\rho)^2 L^2}{(1-\rho)(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^5$$

$$+ \frac{16\alpha_0^5 \rho^2(1+\rho)nK\varsigma_0^2 L^2}{1-\sqrt{\rho}} + \frac{128\alpha_0^7 \rho^6(1+\rho)L^4 nK\varsigma_0^2}{(1-\rho)(1-\sqrt{\rho})^2}$$

$$+ \sum_{k=0}^{K-1} \frac{4\rho^2}{1-\rho^2} \left(\frac{16\alpha_k^5 \rho^2(1+\rho)L^2}{1-\sqrt{\rho}} + \frac{128\alpha_k^8 \rho^6(1+\rho)L^4}{(1-\rho)(1-\sqrt{\rho})^2} \right) \mathbb{E} \|\boldsymbol{U}_k\|_F^2$$

$$+ \sum_{k=0}^{K-1} \left(\frac{16\alpha_k^5 \rho^2(1+\rho)L^2}{1-\sqrt{\rho}} + \frac{128\alpha_k^8 \rho^6(1+\rho)L^4}{(1-\rho)(1-\sqrt{\rho})^2} \right) \cdot$$

$$\left(\frac{16\alpha_k^4 \rho^2 nL^2}{1-\rho^2} \mathbb{E} \left\| \overline{\boldsymbol{G}}_k \right\|^2 + \frac{16\alpha_k^4 \rho^2 \tilde{\sigma}^2 L^2}{1-\rho^2} + 8n\rho^2\tilde{\sigma}^2 \right)$$

$$+ 16nL^2 \sum_{k=0}^{K-1} \alpha_k^5 \mathbb{E} \left\| \overline{\boldsymbol{Y}}_k \right\|^2$$

$$\leq \frac{64\rho^2 \alpha_0^5 (1+\rho)^2 nK\varsigma_0^2 L^2}{(1-\rho)(1-\sqrt{\rho})^2} + \frac{512\rho^4 (1+\rho)L^2}{(1-\rho)^2(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^5 \mathbb{E} \|\boldsymbol{U}_k\|_F^2$$

$$+ \frac{512\rho^4 n\tilde{\sigma}^2 (1+\rho)^2 L^2}{(1-\rho)(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^5 + 32nL^2 \sum_{k=0}^{K-1} \alpha_k^5 \mathbb{E} \left\| \overline{\boldsymbol{G}}_k \right\|^2,$$

where we repeatedly use Equation (37) and Equation (29), solve it we obtain

$$\sum_{k=0}^{K-1} \alpha_k^3 \mathbb{E} \|\boldsymbol{U}_k\|_F^2 \leq \frac{128\rho^2 \alpha_0^5 (1+\rho)^2 nK\varsigma_0^2 L^2}{(1-\rho)(1-\sqrt{\rho})^2} + \frac{1024\rho^4 n\tilde{\sigma}^2 (1+\rho)^2 L^2}{(1-\rho)(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^5$$

$$+ 64nL^2 \sum_{k=0}^{K-1} \alpha_k^5 \mathbb{E} \left\| \overline{\boldsymbol{G}}_k \right\|^2,$$

where again we use Equation (37), combine it with Equation (32) we obtain

$$\sum_{k=0}^{K-1} \alpha_k \mathbb{E} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2$$

$$\leq \frac{4\rho^2 \alpha_0^3 (1+\rho)nK\varsigma_0^2}{(1-\rho)(1-\sqrt{\rho})^2} + \frac{544\rho^4 nL^2}{(1-\rho)^2(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^5 \mathbb{E} \left\| \overline{\boldsymbol{G}}_k \right\|^2 + \frac{32\rho^4 n\tilde{\sigma}^2 (1+\rho)}{(1-\rho)(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^3,$$

where we use Equation (37). Put this result back to Equation (30), rearrange the terms and eliminate the numerical constants, we obtain

$$\sum_{k=0}^{K-1} \alpha_k \mathbb{E} \left\| \nabla f \left(\overline{\boldsymbol{X}}_k \right) \right\|^2$$

$$\leq O \left(\Delta + \frac{\tilde{\sigma}^2 L}{n} \sum_{k=0}^{K-1} \alpha_k^2 + \frac{\rho^2 \alpha_0^3 L^2 K\varsigma_0^2}{(1-\rho)^3} + \frac{\rho^4 \tilde{\sigma}^2 L^2}{(1-\rho)^3} \sum_{k=0}^{K-1} \alpha_k^3 + \frac{\tilde{\sigma}^2 L^2}{n} \sum_{k=0}^{K-1} \alpha_k^3 \right).$$

Fit in $T = KR$ and $\tilde{\sigma}^2 = \sigma^2/BR$, we obtain

$$\sum_{k=0}^{T/R-1} \alpha_k \mathbb{E} \left\| \nabla f \left(\overline{\boldsymbol{X}}_k \right) \right\|^2$$

$$\leq O \left(\Delta + \frac{\sigma^2 L}{nBR} \sum_{k=0}^{T/R-1} \alpha_k^2 + \frac{\rho^2 \alpha_0^3 L^2 T\varsigma_0^2}{(1-\rho)^3 R} + \frac{\rho^4 \sigma^2 L^2}{(1-\rho)^3 BR} \sum_{k=0}^{T/R-1} \alpha_k^3 + \frac{\sigma^2 L^2}{nBR} \sum_{k=0}^{T/R-1} \alpha_k^3 \right)$$

That completes the proof.

∎

B.5 Proof to Corollary 6

Proof Let $\alpha_k = \alpha$ in Theorem 5, we obtain,

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f\left(\overline{X}_k\right)\right\|^2 \le O\left(\frac{\Delta}{\alpha K} + \frac{\alpha \tilde{\sigma}^2 L}{n} + \frac{\rho^2 \alpha^2 L^2 \varsigma_0^2}{(1-\rho)^3} + \frac{\rho^4 \alpha^2 \tilde{\sigma}^2 L^2}{(1-\rho)^3} + \frac{\alpha^2 \tilde{\sigma}^2 L^2}{n}\right),$$

where we omit the numerical constants. Set

$$\alpha = \cfrac{1}{\tilde{\sigma}\sqrt{KL/n\Delta} + \cfrac{\rho^{\frac{2}{3}}L^{\frac{2}{3}}\varsigma_0^{\frac{2}{3}}K^{\frac{1}{3}}}{\Delta^{\frac{1}{3}}(1-\rho)} + \cfrac{32L}{(1-\rho)^2}}$$

$$= \cfrac{1}{\sqrt{\cfrac{\sigma^2 TL}{BR^2 n\Delta}} + \cfrac{\rho^{\frac{2}{3}}L^{\frac{2}{3}}\varsigma_0^{\frac{2}{3}}T^{\frac{1}{3}}}{R^{\frac{1}{3}}\Delta^{\frac{1}{3}}(1-\rho)} + \cfrac{32L}{(1-\rho)^2}},$$

we obtain

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f\left(\overline{X}_k\right)\right\|^2 \le O\left(\frac{\sqrt{\Delta L}\tilde{\sigma}}{\sqrt{nK}} + \frac{(\rho\Delta L\varsigma_0)^{\frac{2}{3}}}{(1-\rho)K^{\frac{2}{3}}} + \frac{\rho^2 n\Delta L}{(1-\rho)^3 K} + \frac{\Delta L}{(1-\rho)^2 K}\right).$$

Fit in $T = KR$ and $\tilde{\sigma}^2 = \sigma^2/BR$, we obtain

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f\left(\overline{X}_k\right)\right\|^2 \le O\left(\frac{\sqrt{\Delta L}\sigma}{\sqrt{nBT}} + \frac{(\rho\Delta L\varsigma_0 R)^{\frac{2}{3}}}{(1-\rho)T^{\frac{2}{3}}} + \frac{\rho^2 nR\Delta L}{(1-\rho)^3 T} + \frac{R\Delta L}{(1-\rho)^2 T}\right),$$

set

$$R = \frac{1}{\sqrt{1-\lambda_2(\boldsymbol{W})}}\max\left(\frac{1}{2}\log(n), \frac{1}{2}\log\left(\frac{\varsigma_0^2 T}{\Delta L}\right)\right),$$

we first have $\rho \le 1/\sqrt{2}$ since

$$R \ge \frac{\log(n)}{2\sqrt{1-\lambda_2(\boldsymbol{W})}} \ge \frac{-\log(n)}{2\log(1-\sqrt{1-\lambda_2(\boldsymbol{W})})} \Rightarrow \left(1-\sqrt{1-\lambda_2(\boldsymbol{W})}\right)^R \le \frac{1}{\sqrt{n}} \le \frac{1}{\sqrt{2}},$$

this implies

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f\left(\overline{X}_k\right)\right\|^2 \le O\left(\frac{\sqrt{\Delta L}\sigma}{\sqrt{nBT}} + \frac{(\rho\Delta L\varsigma_0 R)^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{\rho^2 nR\Delta L}{T} + \frac{R\Delta L}{T}\right)$$

$$\le O\left(\frac{\sqrt{\Delta L}\sigma}{\sqrt{nBT}} + \frac{(\rho\varsigma_0\sqrt{T}R/\sqrt{\Delta L})^{\frac{2}{3}}\Delta L}{T} + \frac{\rho^2 nR\Delta L}{T} + \frac{R\Delta L}{T}\right),$$

with the assignment of R, $\rho^2 n < 1$ and $\rho\varsigma_0\sqrt{T}/\sqrt{\Delta L} < 1$, so since it also holds that $R \ge 1$ (and so $R^{2/3} \le R$),

$$\min_t \left\|\nabla f\left(\overline{X}_t\right)\right\|^2 \le \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f\left(\overline{X}_k\right)\right\|^2$$

$$\leq O\left(\frac{\sqrt{\Delta L}\sigma}{\sqrt{nBT}} + \frac{(R)^{\frac{2}{3}}\Delta L}{T} + \frac{R\Delta L}{T} + \frac{R\Delta L}{T}\right)$$

$$\leq O\left(\frac{\sqrt{\Delta L}\sigma}{\sqrt{nBT}} + \frac{R\Delta L}{T}\right)$$

$$= O\left(\frac{\sqrt{\Delta L}\sigma}{\sqrt{nBT}} + \frac{\Delta L}{T\sqrt{1-\lambda_2(\boldsymbol{W})}} \cdot \max\left(\log(n), \log\left(\frac{\varsigma_0^2 T}{\Delta L}\right)\right)\right),$$

when

$$T \leq O\left(\frac{\Delta L\sigma^2}{nB\epsilon^4}\right),$$

we have

$$\frac{\sqrt{\Delta L}\sigma}{\sqrt{nBT}} \leq O(\epsilon^2).$$

On the other hand, when

$$T \leq O\left(\max\left(\frac{\log(n)\Delta L}{\epsilon^2\sqrt{1-\lambda_2(\boldsymbol{W})}}, \frac{\Delta L}{\epsilon^2\sqrt{1-\lambda_2(\boldsymbol{W})}}\log\left(\frac{\varsigma_0^2}{\epsilon^2\Delta L}\right)\right)\right),$$

we have

$$\frac{\Delta L}{T\sqrt{1-\lambda_2(\boldsymbol{W})}} \cdot \max\left(\log(n), \log\left(\frac{\varsigma_0^2 T}{\Delta L}\right)\right) \leq O(\epsilon^2),$$

to see this, note that

$$\frac{\Delta L}{T\sqrt{1-\lambda_2(\boldsymbol{W})}}\log\left(\frac{\varsigma_0^2 T}{\Delta L}\right) = \epsilon^2 \frac{\log\left(\frac{\varsigma_0^2}{\epsilon^2\Delta L}\log\left(\frac{\varsigma_0^2}{\epsilon^2\Delta L}\right)\right)}{\log\left(\frac{\varsigma_0^2}{\epsilon^2\Delta L}\right)} \leq O(\epsilon^2).$$

Finally, we can obtain the upper bound

$$T \leq O\left(\frac{\Delta L\sigma^2}{nB\epsilon^4} + \max\left(\frac{\log(n)\Delta L}{\epsilon^2\sqrt{1-\lambda_2(\boldsymbol{W})}}, \frac{\Delta L}{\epsilon^2\sqrt{1-\lambda_2(\boldsymbol{W})}}\log\left(\frac{\varsigma_0^2}{\epsilon^2\Delta L}\right)\right)\right)$$

$$= O\left(\frac{\Delta L\sigma^2}{nB\epsilon^4} + \frac{\Delta L}{\epsilon^2\sqrt{1-\lambda_2(\boldsymbol{W})}}\log\left(n + \frac{\varsigma_0 n}{\epsilon\sqrt{\Delta L}}\right)\right),$$

as desired. That completes the proof. ∎

B.6 Proof to Theorem 7

Proof In this proof, we reuse the definition for $\boldsymbol{X}_k$, $\boldsymbol{Y}_k$ and $\tilde{\boldsymbol{G}}_k$ as adopted in the proof to Theorem 5, and additionally define $\boldsymbol{V}_k$ as the momentum matrix, which is

$$\boldsymbol{V}_k = [\boldsymbol{v}_{k,1}, \cdots, \boldsymbol{v}_{k,n}] \in \mathbb{R}^{d \times n},$$

and the update of DeTAG with momentum becomes

$$\begin{aligned}
\boldsymbol{V}_{k+1} &= \beta \boldsymbol{V}_k + \tilde{\boldsymbol{G}}_k \\
\boldsymbol{Y}_{k+1} &= \mathcal{M}(\boldsymbol{Y}_k + \boldsymbol{V}_k - \boldsymbol{V}_{k-1}), \\
\boldsymbol{X}_{k+1} &= \mathcal{M}(\boldsymbol{X}_k - \alpha \boldsymbol{Y}_k).
\end{aligned}$$

From this update rule, we first know that

$$\overline{\boldsymbol{Y}}_k = \mathcal{M}(\boldsymbol{Y}_{k-1} + \boldsymbol{V}_{k-1} - \boldsymbol{V}_{k-1})\frac{\mathbf{1}}{n} = \overline{\boldsymbol{Y}}_{k-1} + \overline{\boldsymbol{V}}_{k-1} - \overline{\boldsymbol{V}}_{k-2} = \overline{\boldsymbol{V}}_0 + \sum_{j=0}^{k-1}(\overline{\boldsymbol{V}}_j - \overline{\boldsymbol{V}}_{j-1}) = \overline{\boldsymbol{V}}_{k-1} \tag{33}$$

and

$$\overline{\boldsymbol{X}}_{k+1} = \mathcal{M}(\boldsymbol{X}_k - \alpha \boldsymbol{Y}_k)\frac{\mathbf{1}}{n} = \overline{\boldsymbol{X}}_k - \alpha \overline{\boldsymbol{Y}}_k. \tag{34}$$

From Lemma 12, we obtain

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{X}}_k\right)\right\|^2 + \frac{1}{K}\sum_{k=0}^{K-1}\left(1 - \frac{4\alpha(1+\tilde{\gamma})L}{1-\beta} - \frac{54\alpha^3 L^2}{(1-\beta)^3}\right)\mathbb{E}\left\|\overline{\boldsymbol{G}}_k\right\|^2$$

$$\leq \frac{2(1-\beta)\Delta}{\alpha K} + \frac{4\alpha\tilde{\gamma}\tilde{\sigma}^2 L}{(1-\beta)n} + \left(\frac{2L^2}{nK} + \frac{12\alpha L^2}{(1-\beta)nK}\right)\sum_{k=0}^{K-1}\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2 + \frac{54\alpha^3\tilde{\sigma}^2 L^2}{(1-\beta)^3 n}. \tag{35}$$

Similar to the proof of Theorem 5, we bound $\sum_{k=0}^{K-1}\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2$ via solving a linear system. With minor modification[12] on Lemma 9 we obtain

$$\left\|\boldsymbol{X}_{k+1} - \overline{\boldsymbol{X}}_{k+1}\mathbf{1}_n^\top\right\|_F^2 \leq \frac{2\rho^2}{(1+\rho^2)}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2 + \frac{2\rho^2\alpha^2}{1-\rho^2}\left\|\boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k\mathbf{1}_n^\top\right\|_F^2.$$

and

$$\begin{aligned}
&\mathbb{E}\left\|\boldsymbol{Y}_{k+1} - \overline{\boldsymbol{Y}}_{k+1}\mathbf{1}_n^\top\right\|_F^2 \\
&\leq \left(\frac{2\rho^2}{1+\rho^2} + \frac{16\alpha^2\rho^4 L^2}{1-\rho^2}\right)\mathbb{E}\left\|\boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k\mathbf{1}_n^\top\right\|_F^2 + \frac{16\rho^2(1+\rho^2)L^2}{1-\rho^2}\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2 \\
&\quad + \frac{16\alpha^2\rho^2 n L^2}{1-\rho^2}\mathbb{E}\left\|\overline{\boldsymbol{Y}}_k\right\|^2 + \frac{4\rho^2}{1-\rho^2}\mathbb{E}\left\|\boldsymbol{G}_{k+2} - \boldsymbol{G}_{k+1} - \boldsymbol{V}_k + \boldsymbol{V}_{k-1}\right\|_F^2.
\end{aligned}$$

---

12. Since the main updates to DeTAG and DeTAGM only differs in using $\boldsymbol{V}$ and $\tilde{\boldsymbol{G}}$.

We now obtain the linear system with momentum version and can perform a similar analysis as in Lemma 10 with

$$\boldsymbol{U}_k = \boldsymbol{G}_{k+2} - \boldsymbol{G}_{k+1} - \boldsymbol{V}_k + \boldsymbol{V}_{k-1}$$

$$\boldsymbol{u}_k = \begin{bmatrix} 0 \\ \frac{4\rho^2}{1-\rho^2} \mathbb{E}\left\| \boldsymbol{U}_k \right\|_F^2 + \frac{16\alpha^2\rho^2 nL^2}{1-\rho^2} \mathbb{E}\left\| \overline{\boldsymbol{Y}}_k \right\|^2 \end{bmatrix}.$$

And we can get the following two bounds:

$$\sum_{k=0}^{K-1} \mathbb{E}\left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2$$

$$\leq \frac{2\rho^2\alpha^2(1+\rho)nK\varsigma_0^2}{(1-\rho)(1-\sqrt{\rho})^2} + \frac{32\rho^4\alpha^4 nL^2}{(1-\rho)^2(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \mathbb{E}\left\| \overline{\boldsymbol{G}}_k \right\|^2 + \frac{8\rho^4\alpha^2}{(1-\rho)^2(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \mathbb{E}\|\boldsymbol{U}_k\|_F^2, \tag{36}$$

and

$$\sum_{k=0}^{K-1} \mathbb{E}\|\boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top\|^2$$

$$\leq \frac{(1+\rho)nK\varsigma_0^2}{1-\sqrt{\rho}} + \frac{8\alpha^2\rho^4(1+\rho)L^2 nK\varsigma_0^2}{(1-\rho)(1-\sqrt{\rho})^2}$$

$$+ \left(\frac{1+\rho}{1-\sqrt{\rho}} + \frac{8\alpha^2\rho^4(1+\rho)L^2}{(1-\rho)(1-\sqrt{\rho})^2} \right) \sum_{k=0}^{K-1} \left(\frac{4\rho^2}{1-\rho^2} \mathbb{E}\left\| \boldsymbol{U}_k \right\|_F^2 + \frac{16\alpha^2\rho^2 nL^2}{1-\rho^2} \mathbb{E}\left\| \overline{\boldsymbol{G}}_k \right\|^2 \right).$$

We next solve $\|\boldsymbol{U}_k\|_F$, from the definition of $\boldsymbol{U}_k$:

$$\sum_{k=0}^{K-1} \mathbb{E}\left\| \boldsymbol{U}_k \right\|_F^2 = \sum_{k=0}^{K-1} \mathbb{E}\left\| \boldsymbol{G}_{k+2} - \boldsymbol{G}_{k+1} - \boldsymbol{V}_k + \boldsymbol{V}_{k-1} \right\|_F^2$$

$$\leq 2\sum_{k=0}^{K-1} \mathbb{E}\left\| \boldsymbol{G}_{k+2} - \boldsymbol{G}_{k+1} \right\|_F^2 + 2\sum_{k=0}^{K-1} \mathbb{E}\left\| \boldsymbol{V}_k - \boldsymbol{V}_{k-1} \right\|_F^2.$$

For the second term,

$$\sum_{k=0}^{K-1} \mathbb{E}\left\| \boldsymbol{V}_k - \boldsymbol{V}_{k-1} \right\|_F^2$$

$$= \sum_{k=0}^{K-1} \mathbb{E}\left\| \sum_{j=0}^{k-1} \beta^{k-j-1}\tilde{\boldsymbol{G}}_j - \sum_{j=0}^{k-2} \beta^{k-j-2}\tilde{\boldsymbol{G}}_j \right\|_F^2$$

$$= \sum_{k=0}^{K-1} \mathbb{E}\left\| \sum_{j=0}^{k-2} \beta^{k-j-2}\tilde{\boldsymbol{G}}_{j+1} - \sum_{j=0}^{k-2} \beta^{k-j-2}\tilde{\boldsymbol{G}}_j \right\|_F^2$$

$$= \frac{1}{(1-\beta)^2} \sum_{k=0}^{K-1} \mathbb{E} \left\| \sum_{j=0}^{k-2} (1-\beta)\beta^{k-j-2} \tilde{\boldsymbol{G}}_{j+1} - \sum_{j=0}^{k-2} (1-\beta)\beta^{k-j-2} \tilde{\boldsymbol{G}}_j \right\|_F^2$$

$$\leq \frac{1}{1-\beta} \sum_{k=0}^{K-1} \sum_{j=0}^{k-2} \beta^{k-j-2} \mathbb{E} \left\| \tilde{\boldsymbol{G}}_{j+1} - \tilde{\boldsymbol{G}}_j \right\|_F^2$$

$$\leq \frac{1}{(1-\beta)^2} \sum_{k=0}^{K-1} \mathbb{E} \left\| \tilde{\boldsymbol{G}}_{k+1} - \tilde{\boldsymbol{G}}_k \right\|_F^2$$

$$= \frac{1}{(1-\beta)^2} \sum_{k=0}^{K-1} \left(\mathbb{E} \left\| \boldsymbol{G}_{k+1} - \boldsymbol{G}_k \right\|_F^2 + \mathbb{E} \left\| \tilde{\boldsymbol{G}}_{k+1} - \boldsymbol{G}_{k+1} \right\|_F^2 + \mathbb{E} \left\| \tilde{\boldsymbol{G}}_k - \boldsymbol{G}_k \right\|_F^2 \right)$$

$$\leq \frac{1}{(1-\beta)^2} \sum_{k=0}^{K-1} \mathbb{E} \left\| \boldsymbol{G}_{k+1} - \boldsymbol{G}_k \right\|_F^2 + \frac{2\tilde{\sigma}^2 nK}{(1-\beta)^2}.$$

Put it back to the previous inequality,

$$\sum_{k=0}^{K-1} \mathbb{E} \left\| \boldsymbol{U}_k \right\|_F^2 \leq 2 \sum_{k=0}^{K-1} \mathbb{E} \left\| \boldsymbol{G}_{k+2} - \boldsymbol{G}_{k+1} \right\|_F^2 + 2 \sum_{k=0}^{K-1} \mathbb{E} \left\| \boldsymbol{V}_k - \boldsymbol{V}_{k-1} \right\|_F^2$$

$$\leq \left(2 + \frac{2}{(1-\beta)^2} \right) L^2 \sum_{k=0}^{K-1} \mathbb{E} \left\| \boldsymbol{X}_{k+1} - \boldsymbol{X}_k \right\|_F^2 + \frac{4\tilde{\sigma}^2 nK}{(1-\beta)^2}$$

$$\leq \frac{4L^2}{(1-\beta)^2} \sum_{k=0}^{K-1} \mathbb{E} \left\| \boldsymbol{X}_{k+1} - \boldsymbol{X}_k \right\|_F^2 + \frac{4\tilde{\sigma}^2 nK}{(1-\beta)^2}$$

Multiplying $(1-\beta)^2$ on both sides for simplicity,

$$(1-\beta)^2 \sum_{k=0}^{K-1} \mathbb{E} \left\| \boldsymbol{U}_k \right\|_F^2$$

$$\leq 4L^2 \sum_{k=0}^{K-1} \mathbb{E} \left\| \boldsymbol{X}_{k+1} - \boldsymbol{X}_k \right\|_F^2 + 4\tilde{\sigma}^2 nK$$

$$\leq 4\tilde{\sigma}^2 nK + 4L^2 \sum_{k=0}^{K-1} \left(4\mathbb{E} \left\| \mathcal{M}(\boldsymbol{X}_k) - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2 + 4\mathbb{E} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2 \right)$$

$$+ 4L^2 \sum_{k=0}^{K-1} \left(4\alpha^2 \mathbb{E} \left\| \mathcal{M}(\boldsymbol{Y}_k) - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \right\|_F^2 + 4\alpha^2 n \mathbb{E} \left\| \overline{\boldsymbol{Y}}_k \right\|^2 \right)$$

$$= 16(1+\rho^2)L^2 \sum_{k=0}^{K-1} \mathbb{E} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2 + 16\alpha^2 \rho^2 L^2 \sum_{k=0}^{K-1} \mathbb{E} \left\| \boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \right\|_F^2$$

$$+ 16\alpha^2 n L^2 \sum_{k=0}^{K-1} \mathbb{E} \left\| \overline{\boldsymbol{Y}}_k \right\|^2 + 4\tilde{\sigma}^2 nK$$

$$\leq \frac{32\rho^2\alpha^2(1+\rho)^2nK\varsigma_0^2L^2}{(1-\rho)(1-\sqrt{\rho})^2} + \frac{512\rho^4(1+\rho)\alpha^4nL^4}{(1-\rho)^2(1-\sqrt{\rho})^2}\sum_{k=0}^{K-1}\mathbb{E}\left\|\overline{\boldsymbol{G}}_k\right\|^2$$

$$+ \frac{256\rho^4(1+\rho)\alpha^2L^2}{(1-\rho)^2(1-\sqrt{\rho})^2}\sum_{k=0}^{K-1}\mathbb{E}\|\boldsymbol{U}_k\|_F^2$$

$$+ 16\alpha^2\rho^2L^2\left(\frac{(1+\rho)nK\varsigma_0^2}{1-\sqrt{\rho}} + \frac{8\alpha^2\rho^4(1+\rho)L^2nK\varsigma_0^2}{(1-\rho)(1-\sqrt{\rho})^2}\right)$$

$$+ \left(\frac{16\alpha^2\rho^2(1+\rho)L^2}{1-\sqrt{\rho}} + \frac{128\alpha^4\rho^6(1+\rho)L^4}{(1-\rho)(1-\sqrt{\rho})^2}\right)\sum_{k=0}^{K-1}\frac{4\rho^2}{1-\rho^2}\mathbb{E}\left\|\boldsymbol{U}_k\right\|_F^2$$

$$+ \left(\frac{16\alpha^2\rho^2(1+\rho)L^2}{1-\sqrt{\rho}} + \frac{128\alpha^4\rho^6(1+\rho)L^4}{(1-\rho)(1-\sqrt{\rho})^2}\right)\sum_{k=0}^{K-1}\frac{16\alpha^2\rho^2nL^2}{1-\rho^2}\mathbb{E}\left\|\overline{\boldsymbol{G}}_k\right\|^2$$

$$+ 16\alpha^2nL^2\sum_{k=0}^{K-1}\mathbb{E}\left\|\overline{\boldsymbol{Y}}_k\right\|^2 + 4\tilde{\sigma}^2nK$$

$$\leq \frac{64\rho^2\alpha^2(1+\rho)^2nK\varsigma_0^2L^2}{(1-\rho)(1-\sqrt{\rho})^2} + 4\tilde{\sigma}^2nK + \frac{512\rho^4(1+\rho)\alpha^2L^2}{(1-\rho)^2(1-\sqrt{\rho})^2}\sum_{k=0}^{K-1}\mathbb{E}\|\boldsymbol{U}_k\|_F^2$$

$$+ 32\alpha^2nL^2\sum_{k=0}^{K-1}\mathbb{E}\left\|\overline{\boldsymbol{G}}_k\right\|^2,$$

where we repeatedly use step size requirement and Equation (29), solve it we obtain

$$\sum_{k=0}^{K-1}\mathbb{E}\left\|\boldsymbol{U}_k\right\|_F^2 \leq \frac{128\rho^2\alpha^2(1+\rho)^2nK\varsigma_0^2L^2}{(1-\beta)^2(1-\rho)(1-\sqrt{\rho})^2} + \frac{8n\tilde{\sigma}^2K}{(1-\beta)^2} + \frac{64\alpha^2nL^2}{(1-\beta)^2}\sum_{k=0}^{K-1}\mathbb{E}\left\|\overline{\boldsymbol{G}}_k\right\|^2,$$

combine it with Equation (36) we obtain

$$\sum_{k=0}^{K-1}\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2$$

$$\leq \frac{4\rho^2\alpha^2(1+\rho)nK\varsigma_0^2}{(1-\rho)(1-\sqrt{\rho})^2} + \frac{544\rho^4\alpha^4nL^2}{(1-\rho)^2(1-\sqrt{\rho})^2}\sum_{k=0}^{K-1}\mathbb{E}\left\|\overline{\boldsymbol{G}}_k\right\|^2 + \frac{64\rho^4\alpha^2n\tilde{\sigma}^2K}{(1-\beta)^2(1-\rho)^2(1-\sqrt{\rho})^2}.$$

Recall from Equation (35) that

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{X}}_k\right)\right\|^2 + \frac{1}{K}\sum_{k=0}^{K-1}\left(1 - \frac{4\alpha(1+\tilde{\gamma})L}{1-\beta} - \frac{54\alpha^3L^2}{(1-\beta)^3}\right)\mathbb{E}\left\|\overline{\boldsymbol{G}}_k\right\|^2$$

$$\leq \frac{2(1-\beta)\Delta}{\alpha K} + \frac{4\alpha\tilde{\gamma}\tilde{\sigma}^2L}{(1-\beta)n} + \left(\frac{2L^2}{nK} + \frac{12\alpha L^2}{(1-\beta)nK}\right)\sum_{k=0}^{K-1}\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2 + \frac{54\alpha^3\tilde{\sigma}^2L^2}{(1-\beta)^3n}.$$

We obtain

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{X}}_k\right)\right\|^2$$

$$\leq O\left(\frac{(1-\beta)\Delta}{\alpha K} + \frac{\alpha\tilde{\sigma}^2 L}{(1-\beta)n} + \frac{\rho^2\alpha^2 L^2\varsigma_0^2}{(1-\beta)(1-\rho)^3} + \frac{\rho^4\alpha^2\tilde{\sigma}^2 L^2}{(1-\beta)^3(1-\rho)^4} + \frac{\alpha^2\tilde{\sigma}^2 L^2}{(1-\beta)^3 n}\right),$$

where we omit the numerical constants. Set

$$\alpha = \frac{1}{\tilde{\sigma}\sqrt{\frac{KL}{(1-\beta)^2 n\Delta}} + \frac{\rho^{\frac{2}{3}}L^{\frac{2}{3}}\varsigma_0^{\frac{2}{3}}K^{\frac{1}{3}}}{(1-\beta)^{\frac{2}{3}}\Delta^{\frac{1}{3}}(1-\rho)} + \frac{32(1+\tilde{\gamma})L}{(1-\beta)^2(1-\rho)^2}}$$

$$= \frac{1}{\sqrt{\frac{\sigma^2 TL}{(1-\beta)^2 BR^2 n\Delta}} + \frac{\rho^{\frac{2}{3}}L^{\frac{2}{3}}\varsigma_0^{\frac{2}{3}}T^{\frac{1}{3}}}{(1-\beta)^{\frac{2}{3}}R^{\frac{1}{3}}\Delta^{\frac{1}{3}}(1-\rho)} + \frac{32(1+\tilde{\gamma})L}{(1-\beta)^2(1-\rho)^2}},$$

we obtain

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{X}}_k\right)\right\|^2$$

$$\leq O\left(\frac{\sqrt{\Delta L}\tilde{\sigma}}{\sqrt{nK}} + \frac{(\rho\Delta L\varsigma_0)^{\frac{2}{3}}}{(1-\rho)K^{\frac{2}{3}}} + \frac{\rho^2 n\Delta L}{(1-\rho)^4 K} + \frac{\Delta L}{(1-\beta)(1-\rho)^2 K}\right).$$

By performing a similar analysis as in Corollary 6 (we omit it here for brevity), we obtain

$$T \leq O\left(\frac{\Delta L\sigma^2}{nB\epsilon^4} + \frac{\Delta L}{\epsilon^2(1-\beta)\sqrt{1-\lambda_2(\boldsymbol{W})}}\log\left(n + \frac{\varsigma_0 n}{\epsilon\sqrt{\Delta L}}\right)\right),$$

that completes the proof. ∎

B.7 Technical Lemmas

Lemma 9 *Given the update formula of DeTAG (and DeTAGM), we have the following two bounds:*

$$\left\|\boldsymbol{X}_{k+1} - \overline{\boldsymbol{X}}_{k+1}\mathbf{1}_n^\top\right\|_F^2 \leq \frac{2\rho^2}{(1+\rho^2)}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2 + \frac{2\rho^2\alpha_k^2}{1-\rho^2}\left\|\boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k\mathbf{1}_n^\top\right\|_F^2,$$

and

$$\mathbb{E}\left\|\boldsymbol{Y}_{k+1} - \overline{\boldsymbol{Y}}_{k+1}\mathbf{1}_n^\top\right\|_F^2$$

$$\leq \left(\frac{2\rho^2}{1+\rho^2} + \frac{16\alpha_k^2\rho^4 L^2}{1-\rho^2}\right)\mathbb{E}\left\|\boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k\mathbf{1}_n^\top\right\|_F^2 + \frac{16\rho^2(1+\rho^2)L^2}{1-\rho^2}\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2$$

$$+ \frac{16\alpha_k^2\rho^2 nL^2}{1-\rho^2}\mathbb{E}\left\|\overline{\boldsymbol{Y}}_k\right\|^2 + \frac{4\rho^2}{1-\rho^2}\mathbb{E}\left\|\boldsymbol{G}_{k+2} - \boldsymbol{G}_{k+1} - \boldsymbol{G}_k + \boldsymbol{G}_{k-1}\right\|_F^2 + 8n\rho^2\tilde{\sigma}^2.$$

Proof

$$\left\|\boldsymbol{X}_{k+1} - \overline{\boldsymbol{X}}_{k+1}\mathbf{1}_n^\top\right\|_F^2$$

$$\overset{(28)}{=} \left\| \mathcal{M}(\boldsymbol{X}_k - \alpha_k \boldsymbol{Y}_k) - (\overline{\boldsymbol{X}}_k - \alpha_k \overline{\boldsymbol{Y}}_k)\mathbf{1}_n^\top \right\|_F^2$$

$$= \left\| \mathcal{M}(\boldsymbol{X}_k) - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2 - 2\alpha_k \left\langle \mathcal{M}(\boldsymbol{X}_k) - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top, \mathcal{M}(\boldsymbol{Y}_k) - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \right\rangle$$

$$+ \alpha_k^2 \left\| \mathcal{M}(\boldsymbol{Y}_k) - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \right\|_F^2$$

$$\overset{(26)}{\leq} \rho^2 \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2 + \frac{\rho^2(1-\rho^2)}{1+\rho^2} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2 + \frac{\rho^2(1+\rho^2)\alpha_k^2}{1-\rho^2} \left\| \boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \right\|_F^2$$

$$+ \alpha_k^2 \rho^2 \left\| \boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \right\|_F^2$$

$$= \frac{2\rho^2}{(1+\rho^2)} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2 + \frac{2\rho^2 \alpha_k^2}{1-\rho^2} \left\| \boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \right\|_F^2,$$

where in the third step we use the fact that for any $\boldsymbol{a}$ and $\boldsymbol{b}$,

$$-2\langle \boldsymbol{a}, \boldsymbol{b} \rangle \leq \frac{1-\rho^2}{1+\rho^2} \|\boldsymbol{a}\|^2 + \frac{1+\rho^2}{1-\rho^2} \|\boldsymbol{b}\|^2.$$

For $\boldsymbol{Y}_k$, we have

$$\mathbb{E} \left\| \boldsymbol{Y}_{k+1} - \overline{\boldsymbol{Y}}_{k+1} \mathbf{1}_n^\top \right\|_F^2$$

$$\overset{(27)}{=} \mathbb{E} \left\| \mathcal{M}(\boldsymbol{Y}_k + \tilde{\boldsymbol{G}}_k - \tilde{\boldsymbol{G}}_{k-1}) - (\overline{\boldsymbol{Y}}_k + \overline{\tilde{\boldsymbol{G}}}_k - \overline{\tilde{\boldsymbol{G}}}_{k-1})\mathbf{1}_n^\top \right\|_F^2$$

$$= \mathbb{E} \left\| \mathcal{M}(\boldsymbol{Y}_k) - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \right\|_F^2 + \mathbb{E} \left\| \mathcal{M}(\tilde{\boldsymbol{G}}_k - \tilde{\boldsymbol{G}}_{k-1}) - (\overline{\tilde{\boldsymbol{G}}}_k - \overline{\tilde{\boldsymbol{G}}}_{k-1})\mathbf{1}_n^\top \right\|_F^2$$

$$+ 2\mathbb{E} \left\langle \mathcal{M}(\boldsymbol{Y}_k) - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top, \mathcal{M}(\tilde{\boldsymbol{G}}_k - \tilde{\boldsymbol{G}}_{k-1}) - (\overline{\tilde{\boldsymbol{G}}}_k - \overline{\tilde{\boldsymbol{G}}}_{k-1})\mathbf{1}_n^\top \right\rangle$$

$$\overset{(26)}{\leq} \rho^2 \mathbb{E} \left\| \boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \right\|_F^2 + \rho^2 \mathbb{E} \left\| \boldsymbol{G}_k - \boldsymbol{G}_{k-1} - (\overline{\boldsymbol{G}}_k - \overline{\boldsymbol{G}}_{k-1})\mathbf{1}_n^\top \right\|_F^2$$

$$+ \frac{(1-\rho^2)\rho^2}{1+\rho^2} \mathbb{E} \left\| \boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \right\|_F^2 + \frac{(1+\rho^2)\rho^2}{1-\rho^2} \mathbb{E} \left\| \boldsymbol{G}_k - \boldsymbol{G}_{k-1} - (\overline{\boldsymbol{G}}_k - \overline{\boldsymbol{G}}_{k-1})\mathbf{1}_n^\top \right\|_F^2$$

$$+ 2\rho^2 \mathbb{E}\|\boldsymbol{G}_k - \tilde{\boldsymbol{G}}_k\|_F^2 + 2\rho^2 \mathbb{E}\|\boldsymbol{G}_{k-1} - \tilde{\boldsymbol{G}}_{k-1}\|_F^2 + 2\rho^2 \mathbb{E}\|\overline{\boldsymbol{G}_k}\mathbf{1}_n^\top - \overline{\tilde{\boldsymbol{G}}}_k \mathbf{1}_n^\top\|_F^2$$

$$+ 2\rho^2 \mathbb{E}\|\overline{\boldsymbol{G}_{k-1}}\mathbf{1}_n^\top - \overline{\tilde{\boldsymbol{G}}}_{k-1} \mathbf{1}_n^\top\|_F^2$$

$$\leq \frac{2\rho^2}{1+\rho^2} \mathbb{E} \left\| \boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \right\|_F^2 + \frac{4\rho^2}{1-\rho^2} \mathbb{E} \left\| \boldsymbol{G}_{k+2} - \boldsymbol{G}_{k+1} \right\|_F^2$$

$$+ \frac{4\rho^2}{1-\rho^2} \mathbb{E} \left\| \boldsymbol{G}_{k+2} - \boldsymbol{G}_{k+1} - \boldsymbol{G}_k + \boldsymbol{G}_{k-1} \right\|_F^2 + 8n\rho^2 \tilde{\sigma}^2,$$

where in the last step we use $\|I - \frac{\mathbf{1}\mathbf{1}^\top}{n}\| \leq 1$ and $\|\boldsymbol{A}\boldsymbol{B}\|_F \leq \|\boldsymbol{A}\|_F \|\boldsymbol{B}\|$.

For the second term, we have

$$\mathbb{E} \left\| \boldsymbol{G}_{k+2} - \boldsymbol{G}_{k+1} \right\|_F^2$$

$$= \sum_{i=1}^n \mathbb{E} \left\| \nabla f(\boldsymbol{x}_{k+1,i}) - \nabla f(\boldsymbol{x}_{k,i}) \right\|^2$$

$$\leq L^2 \sum_{i=1}^{n} \mathbb{E} \left\| \boldsymbol{x}_{k+1,i} - \boldsymbol{x}_{k,i} \right\|^2$$

$$= L^2 \mathbb{E} \left\| \boldsymbol{X}_{k+1} - \boldsymbol{X}_k \right\|_F^2$$

$$\overset{(28)}{=} L^2 \mathbb{E} \left\| \mathcal{M}(\boldsymbol{X}_k) - \boldsymbol{X}_k - \alpha_k \mathcal{M}(\boldsymbol{Y}_k) \right\|_F^2$$

$$= L^2 \mathbb{E} \left\| \mathcal{M}(\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top) - (\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top) - \alpha_k \mathcal{M}(\boldsymbol{Y}_k) \right\|_F^2$$

$$\leq 4L^2 \mathbb{E} \left\| \mathcal{M}(\boldsymbol{X}_k) - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2 + 4L^2 \mathbb{E} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2 + 4\alpha_k^2 L^2 \mathbb{E} \left\| \mathcal{M}(\boldsymbol{Y}_k) - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \right\|_F^2$$

$$+ 4\alpha_k^2 n L^2 \mathbb{E} \left\| \overline{\boldsymbol{Y}}_k \right\|^2$$

$$\leq 4(1 + \rho^2) L^2 \mathbb{E} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2 + 4\alpha_k^2 \rho^2 L^2 \mathbb{E} \left\| \boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \right\|_F^2 + 4\alpha_k^2 n L^2 \mathbb{E} \left\| \overline{\boldsymbol{Y}}_k \right\|^2.$$

Putting it back we obtain

$$\mathbb{E} \left\| \boldsymbol{Y}_{k+1} - \overline{\boldsymbol{Y}}_{k+1} \mathbf{1}_n^\top \right\|_F^2$$

$$\leq \left(\frac{2\rho^2}{1 + \rho^2} + \frac{16\alpha_k^2 \rho^4 L^2}{1 - \rho^2} \right) \mathbb{E} \left\| \boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \right\|_F^2 + \frac{16\rho^2(1 + \rho^2)L^2}{1 - \rho^2} \mathbb{E} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2$$

$$+ \frac{16\alpha_k^2 \rho^2 n L^2}{1 - \rho^2} \mathbb{E} \left\| \overline{\boldsymbol{Y}}_k \right\|^2 + \frac{4\rho^2}{1 - \rho^2} \mathbb{E} \left\| \boldsymbol{G}_{k+2} - \boldsymbol{G}_{k+1} - \boldsymbol{G}_k + \boldsymbol{G}_{k-1} \right\|_F^2 + 8n\rho^2 \tilde{\sigma}^2.$$

That completes the proof. ∎

Lemma 10 *Given the update formula of DeTAG, if we use non-increasing step size with the constraint*

$$\alpha_0 \leq \frac{(1-\rho)^2}{32L},$$

then for any positive integer $p = 1, 2, \cdots$, the following two bounds hold

$$\sum_{k=0}^{K-1} \alpha_k^p \mathbb{E} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2$$

$$\leq \frac{2\rho^2 \alpha_0^{p+2}(1+\rho)nK\varsigma_0^2}{(1-\rho)(1-\sqrt{\rho})^2} + \frac{32\rho^4 nL^2}{(1-\rho)^2(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^{p+4} \mathbb{E} \left\| \overline{\boldsymbol{G}}_k \right\|^2$$

$$+ \frac{8\rho^4}{(1-\rho)^2(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^{p+2} \mathbb{E} \| \boldsymbol{U}_k \|_F^2 + \frac{16\rho^4 n\tilde{\sigma}^2(1+\rho)}{(1-\rho)(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^{p+2},$$

and

$$\sum_{k=0}^{K-1} \alpha_k^p \mathbb{E} \| \boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k \mathbf{1}_n^\top \|^2$$

$$\leq \frac{(1+\rho)\alpha_0^p nK\varsigma_0^2}{1-\sqrt{\rho}} + \frac{8\alpha_0^{p+2}\rho^4(1+\rho)L^2nK\varsigma_0^2}{(1-\rho)(1-\sqrt{\rho})^2} + \sum_{k=0}^{K-1}\alpha_k^p\left(\frac{1+\rho}{1-\sqrt{\rho}} + \frac{8\alpha_k^2\rho^4(1+\rho)L^2}{(1-\rho)(1-\sqrt{\rho})^2}\right).$$
$$\left(\frac{4\rho^2}{1-\rho^2}\mathbb{E}\left\|\boldsymbol{U}_k\right\|_F^2 + \frac{16\alpha_k^2\rho^2nL^2}{1-\rho^2}\mathbb{E}\left\|\overline{\boldsymbol{G}}_k\right\|^2 + \frac{16\alpha_k^2\rho^2\tilde{\sigma}^2L^2}{1-\rho^2} + 8n\rho^2\tilde{\sigma}^2\right).$$

Proof Given the update formula of DeTAG and Lemma 9, the algorithm induces the following linear system: for any $k \geq 0$,

$$\begin{bmatrix} \mathbb{E}\left\|\boldsymbol{X}_{k+1} - \overline{\boldsymbol{X}}_{k+1}\mathbf{1}_n^\top\right\|_F^2 \\ \mathbb{E}\left\|\boldsymbol{Y}_{k+1} - \overline{\boldsymbol{Y}}_{k+1}\mathbf{1}_n^\top\right\|_F^2 \end{bmatrix} \preceq \begin{bmatrix} \boldsymbol{P}_{k,11} & \boldsymbol{P}_{k,12} \\ \boldsymbol{P}_{k,21} & \boldsymbol{P}_{k,22} \end{bmatrix} \begin{bmatrix} \mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2 \\ \mathbb{E}\left\|\boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k\mathbf{1}_n^\top\right\|_F^2 \end{bmatrix}$$
$$+ \begin{bmatrix} 0 \\ \frac{4\rho^2}{1-\rho^2}\mathbb{E}\left\|\boldsymbol{U}_k\right\|_F^2 + \frac{16\alpha_k^2\rho^2nL^2}{1-\rho^2}\mathbb{E}\left\|\overline{\boldsymbol{Y}}_k\right\|^2 + 8n\rho^2\tilde{\sigma}^2 \end{bmatrix},$$

where $\preceq$ is element-wise and

$$\boldsymbol{P}_{k,11} = \frac{2\rho^2}{1+\rho^2}$$
$$\boldsymbol{P}_{k,12} = \frac{2\rho^2\alpha_k^2}{1-\rho^2}$$
$$\boldsymbol{P}_{k,21} = \frac{16\rho^2(1+\rho^2)L^2}{1-\rho^2}$$
$$\boldsymbol{P}_{k,22} = \frac{2\rho^2}{1+\rho^2} + \frac{16\alpha_k^2\rho^4L^2}{1-\rho^2}$$
$$\boldsymbol{U}_k = \boldsymbol{G}_{k+2} - \boldsymbol{G}_{k+1} - \boldsymbol{G}_k + \boldsymbol{G}_{k-1}.$$

For simplicity, define

$$\boldsymbol{z}_k = \begin{bmatrix} \mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2 \\ \mathbb{E}\left\|\boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k\mathbf{1}_n^\top\right\|_F^2 \end{bmatrix}$$
$$\boldsymbol{u}_k = \begin{bmatrix} 0 \\ \frac{4\rho^2}{1-\rho^2}\mathbb{E}\left\|\boldsymbol{U}_k\right\|_F^2 + \frac{16\alpha_k^2\rho^2nL^2}{1-\rho^2}\mathbb{E}\left\|\overline{\boldsymbol{Y}}_k\right\|^2 + 8n\rho^2\tilde{\sigma}^2 \end{bmatrix},$$

then we can write this linear system as

$$\boldsymbol{z}_k \preceq \boldsymbol{P}_{k-1}\boldsymbol{z}_{k-1} + \boldsymbol{u}_{k-1} \preceq \boldsymbol{P}_0^k\boldsymbol{z}_0 + \sum_{t=0}^{k-1}\boldsymbol{P}_t^{k-t}\boldsymbol{u}_t,$$

where we use the fact that $\boldsymbol{P}_k \preceq \boldsymbol{P}_t, \forall k \geq t \geq 0$ since step size is non-increasing. For any $t \geq 0$, let $\lambda_1(\boldsymbol{P}_t), \lambda_2(\boldsymbol{P}_t)$ denote the two eigenvalues of $\boldsymbol{P}_t$ (without the loss of generality, we denote $\lambda_1(\boldsymbol{P}_t) < \lambda_2(\boldsymbol{P}_t)$), define

$$\Psi = \sqrt{(\boldsymbol{P}_{t,11} - \boldsymbol{P}_{t,22})^2 + 4\boldsymbol{P}_{t,12}\boldsymbol{P}_{t,21}},$$

then with eigendecomposition, we obtain

$$\lambda_1(\boldsymbol{P}_t) = \frac{\boldsymbol{P}_{t,11} + \boldsymbol{P}_{t,22} - \Psi}{2}$$

$$\lambda_2(\boldsymbol{P}_t) = \frac{\boldsymbol{P}_{t,11} + \boldsymbol{P}_{t,22} + \Psi}{2} = \frac{2\rho^2}{1+\rho^2} + \frac{8\alpha_t^2\rho^4 L^2}{1-\rho^2} + \frac{16\alpha_t\rho^2 L\sqrt{\alpha_t^2\rho^4 L^2 + (1+\rho^2)}}{1-\rho^2}$$

$$\boldsymbol{P}_t^k \preceq \begin{bmatrix} \frac{\lambda_1^k(\boldsymbol{P}_t)+\lambda_2^k(\boldsymbol{P}_t)}{2} + \frac{(\boldsymbol{P}_{t,11}-\boldsymbol{P}_{t,22})(\lambda_2^k(\boldsymbol{P}_t)-\lambda_1^k(\boldsymbol{P}_t))}{2\Psi} & \frac{\boldsymbol{P}_{t,12}}{\Psi}(\lambda_2^k(\boldsymbol{P}_t) - \lambda_1^k(\boldsymbol{P}_t)) \\ \frac{\boldsymbol{P}_{21}}{\Psi}(\lambda_2^k(\boldsymbol{P}_t) - \lambda_1^k(\boldsymbol{P}_t)) & \frac{\lambda_1^k(\boldsymbol{P}_t)+\lambda_2^k(\boldsymbol{P})}{2} + \frac{(\boldsymbol{P}_{t,11}-\boldsymbol{P}_{t,22})(\lambda_1^k(\boldsymbol{P}_t)-\lambda_2^k(\boldsymbol{P}_t))}{2\Psi} \end{bmatrix},$$

Applying the step size requirement that

$$\alpha_0 L < \frac{(1-\rho)^2}{32}, \tag{37}$$

it can be verified that $\lambda_2(\boldsymbol{P}_t) \leq \lambda_2(\boldsymbol{P}_0) \leq \frac{\sqrt{\rho}+\rho}{1+\rho}$. Next we can compute the $\mathbb{E}\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\|_F^2$ and $\mathbb{E}\|\boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k\mathbf{1}_n^\top\|_F^2$: we use $\boldsymbol{X}[1:]$ to denote the first row of matrix $\boldsymbol{X}$. First for $\boldsymbol{X}_k$, we obtain:

$$\boldsymbol{P}_0^k\boldsymbol{z}_0[1:] \leq \boldsymbol{P}_{0,12}k\lambda_2^{k-1}(\boldsymbol{P}_0)\mathbb{E}\|\boldsymbol{Y}_0 - \overline{\boldsymbol{Y}}_0\mathbf{1}_n^\top\|_F^2 = \frac{2\rho^2\alpha_0^2 k}{1-\rho^2}\lambda_2^{k-1}(\boldsymbol{P}_0)\mathbb{E}\|\boldsymbol{Y}_0 - \overline{\boldsymbol{Y}}_0\mathbf{1}_n^\top\|_F^2$$

where we use the property that for any matrix $\boldsymbol{P}$,

$$\lambda_2^k(\boldsymbol{P}) - \lambda_1^k(\boldsymbol{P}) = (\lambda_2(\boldsymbol{P}) - \lambda_1(\boldsymbol{P}))\sum_{l=0}^{k-1}\lambda_2(\boldsymbol{P})^l\lambda_1(\boldsymbol{P})^{k-1-l} = \Psi k\lambda_2^{k-1}(\boldsymbol{P}).$$

And similarly,

$$\boldsymbol{P}_t^{k-t}\boldsymbol{u}_t[1:]$$

$$\leq \frac{2\rho^2\alpha_t^2(k-t)}{1-\rho^2}\lambda_2^{k-t-1}(\boldsymbol{P}_t)\left(\frac{4\rho^2}{1-\rho^2}\mathbb{E}\|\boldsymbol{U}_t\|_F^2 + \frac{16\alpha_t^2\rho^2 nL^2}{1-\rho^2}\mathbb{E}\left\|\overline{\boldsymbol{Y}}_t\right\|^2 + 8n\rho^2\tilde{\sigma}^2\right)$$

$$= \frac{2\rho^2\alpha_t^2(k-t)}{1-\rho^2}\lambda_2^{k-t-1}(\boldsymbol{P}_t)\left(\frac{4\rho^2}{1-\rho^2}\mathbb{E}\|\boldsymbol{U}_t\|_F^2 + \frac{16\alpha_t^2\rho^2 nL^2}{1-\rho^2}\mathbb{E}\left\|\overline{\tilde{\boldsymbol{G}}}_{t-1}\right\|^2 + 8n\rho^2\tilde{\sigma}^2\right)$$

$$\overset{(29)}{\leq} \frac{2\rho^2\alpha_t^2(k-t)}{1-\rho^2}\lambda_2^{k-t-1}(\boldsymbol{P}_t)\left(\frac{4\rho^2}{1-\rho^2}\mathbb{E}\|\boldsymbol{U}_t\|_F^2 + \frac{16\alpha_t^2\rho^2 nL^2}{1-\rho^2}\mathbb{E}\left\|\overline{\boldsymbol{G}}_{t-1}\right\|^2 + \frac{16\alpha_t^2\rho^2\tilde{\sigma}^2 L^2}{1-\rho^2} + 8n\rho^2\tilde{\sigma}^2\right),$$

then from the linear system, we obtain

$$\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2$$

$$\leq \frac{2\rho^2\alpha_0^2 k}{1-\rho^2}\lambda_2^{k-1}(\boldsymbol{P}_0)\mathbb{E}\|\boldsymbol{Y}_0 - \overline{\boldsymbol{Y}}_0\mathbf{1}_n^\top\|_F^2$$

$$+ \sum_{t=0}^{k-1}\frac{2\rho^2\alpha_t^2(k-t)}{1-\rho^2}\lambda_2^{k-t-1}(\boldsymbol{P}_t)\cdot$$

$$\left(\frac{4\rho^2}{1-\rho^2}\mathbb{E}\|\boldsymbol{U}_t\|_F^2 + \frac{16\alpha_t^2\rho^2 nL^2}{1-\rho^2}\mathbb{E}\left\|\overline{\boldsymbol{G}}_{t-1}\right\|^2 + \frac{16\alpha_t^2\rho^2\tilde{\sigma}^2 L^2}{1-\rho^2} + 8n\rho^2\tilde{\sigma}^2\right).$$

53

Summing over $k = 0$ to $K - 1$ we obtain for any positive integer $p = 1, 2, \cdots$,

$$\sum_{k=0}^{K-1} \alpha_k^p \mathbb{E} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \boldsymbol{1}_n^\top \right\|_F^2$$

$$\leq \frac{2\rho^2 \alpha_0^{p+2}}{(1-\rho^2)(1-\lambda_2(\boldsymbol{P}_0))^2} \sum_{k=0}^{K-1} \mathbb{E} \| \boldsymbol{Y}_0 - \overline{\boldsymbol{Y}}_0 \boldsymbol{1}_n^\top \|_F^2$$

$$+ \frac{2\rho^2}{(1-\rho^2)(1-\lambda_2(\boldsymbol{P}_0))^2} \sum_{k=0}^{K-1} \alpha_k^{p+2} \cdot$$

$$\left(\frac{4\rho^2}{1-\rho^2} \mathbb{E} \| \boldsymbol{U}_k \|_F^2 + \frac{16\alpha_k^2 \rho^2 nL^2}{1-\rho^2} \mathbb{E} \left\| \overline{\boldsymbol{G}}_k \right\|^2 + \frac{16\alpha_k^2 \rho^2 \tilde{\sigma}^2 L^2}{1-\rho^2} + 8n\rho^2\tilde{\sigma}^2 \right)$$

$$\leq \frac{2\rho^2 \alpha_0^{p+2}(1+\rho)nK\varsigma_0^2}{(1-\rho)(1-\sqrt{\rho})^2} + \frac{32\rho^4 nL^2}{(1-\rho)^2(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^{p+4} \mathbb{E} \left\| \overline{\boldsymbol{G}}_k \right\|^2$$

$$+ \frac{8\rho^4}{(1-\rho)^2(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^{p+2} \mathbb{E} \| \boldsymbol{U}_k \|_F^2 + \frac{32\rho^4 \tilde{\sigma}^2 L^2}{(1-\rho)^2(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^{p+4}$$

$$+ \frac{8\rho^4 n\tilde{\sigma}^2(1+\rho)}{(1-\rho)(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^{p+2}$$

$$\leq \frac{2\rho^2 \alpha_0^{p+2}(1+\rho)nK\varsigma_0^2}{(1-\rho)(1-\sqrt{\rho})^2} + \frac{32\rho^4 nL^2}{(1-\rho)^2(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^{p+4} \mathbb{E} \left\| \overline{\boldsymbol{G}}_k \right\|^2$$

$$+ \frac{8\rho^4}{(1-\rho)^2(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^{p+2} \mathbb{E} \| \boldsymbol{U}_k \|_F^2 + \frac{16\rho^4 n\tilde{\sigma}^2(1+\rho)}{(1-\rho)(1-\sqrt{\rho})^2} \sum_{k=0}^{K-1} \alpha_k^{p+2},$$

where in the first step we applied Lemma 11, in the second step we used $\frac{1}{1-\lambda_2(\boldsymbol{P}_0)} < \frac{1+\rho}{1-\sqrt{\rho}}$ since $\lambda_2(\boldsymbol{P}_0) \leq \frac{\sqrt{\rho}+\rho}{1+\rho}$. And the third step holds due to Equation (37). We proceed to analyze the case in $\boldsymbol{Y}_k$: we first have for any $t \geq 0$,

$$[\boldsymbol{P}_t^k]_{22} = \frac{\lambda_1^k(\boldsymbol{P}_t) + \lambda_2^k(\boldsymbol{P}_t)}{2} + \frac{(\boldsymbol{P}_{t,11} - \boldsymbol{P}_{t,22})(\lambda_1^k(\boldsymbol{P}_t) - \lambda_2^k(\boldsymbol{P}_t))}{2\Psi}$$

$$\leq \lambda_2^k(\boldsymbol{P}_t) + \frac{8\alpha_t^2 \rho^4 L^2 k \lambda_2^{k-1}(\boldsymbol{P}_t)}{1-\rho^2},$$

then we can have

$$\boldsymbol{P}_0^k \boldsymbol{z}_0[2:] \leq \left(\lambda_2^k(\boldsymbol{P}_0) + \frac{8\alpha_0^2 \rho^4 L^2 k \lambda_2^{k-1}(\boldsymbol{P}_0)}{1-\rho^2} \right) \mathbb{E} \| \boldsymbol{Y}_0 - \overline{\boldsymbol{Y}}_0 \boldsymbol{1}_n^\top \|_F^2,$$

and

$$\boldsymbol{P}_t^{k-t} \boldsymbol{u}_t[2:]$$

$$\leq \left(\lambda_2^{k-t}(\boldsymbol{P}_t) + \frac{8\alpha_t^2 \rho^4 L^2 (k-t) \lambda_2^{k-t-1}(\boldsymbol{P}_t)}{1-\rho^2} \right) \cdot$$

$$\left(\frac{4\rho^2}{1-\rho^2}\mathbb{E}\left\|\boldsymbol{U}_t\right\|_F^2 + \frac{16\alpha_t^2\rho^2 nL^2}{1-\rho^2}\mathbb{E}\left\|\overline{\boldsymbol{Y}}_t\right\|^2 + 8n\rho^2\tilde{\sigma}^2\right)$$

$$\leq\left(\lambda_2^{k-t}(\boldsymbol{P}_t) + \frac{8\alpha_t^2\rho^4 L^2(k-t)\lambda_2^{k-t-1}(\boldsymbol{P}_t)}{1-\rho^2}\right)\cdot$$

$$\left(\frac{4\rho^2}{1-\rho^2}\mathbb{E}\left\|\boldsymbol{U}_t\right\|_F^2 + \frac{16\alpha_t^2\rho^2 nL^2}{1-\rho^2}\mathbb{E}\left\|\overline{\boldsymbol{G}}_{t-1}\right\|^2 + \frac{16\alpha_t^2\rho^2\tilde{\sigma}^2 L^2}{1-\rho^2} + 8n\rho^2\tilde{\sigma}^2\right).$$

Using Lemma 11, and summing over $k = 0$ to $K - 1$, we obtain for any positive integer $p = 1, 2, \cdots$,

$$\sum_{k=0}^{K-1}\alpha_k^p\mathbb{E}\|\boldsymbol{Y}_k - \overline{\boldsymbol{Y}}_k\mathbf{1}_n^\top\|^2$$

$$\leq\frac{(1+\rho)\alpha_0^p nK\varsigma_0^2}{1-\sqrt{\rho}} + \frac{8\alpha_0^{p+2}\rho^4(1+\rho)L^2 nK\varsigma_0^2}{(1-\rho)(1-\sqrt{\rho})^2} + \sum_{k=0}^{K-1}\alpha_k^p\left(\frac{1+\rho}{1-\sqrt{\rho}} + \frac{8\alpha_k^2\rho^4(1+\rho)L^2}{(1-\rho)(1-\sqrt{\rho})^2}\right)\cdot$$

$$\left(\frac{4\rho^2}{1-\rho^2}\mathbb{E}\left\|\boldsymbol{U}_k\right\|_F^2 + \frac{16\alpha_k^2\rho^2 nL^2}{1-\rho^2}\mathbb{E}\left\|\overline{\boldsymbol{G}}_k\right\|^2 + \frac{16\alpha_k^2\rho^2\tilde{\sigma}^2 L^2}{1-\rho^2} + 8n\rho^2\tilde{\sigma}^2\right).$$

That completes the proof. ∎

Lemma 11 *(Zhang and You, 2019c) Let $\{b_k\}$ be a non-negative sequence, θ be a constant in $(0, 1)$, and $a_k = \sum_{t=1}^k s_t(k-t)\theta^{k-t-1}$, it holds that*

$$\sum_{t=1}^k a_t = \frac{1}{(1-\theta)^2}\sum_{t=1}^k s_t.$$

Lemma 12 *If $\frac{\beta^2}{1-\beta} < \tilde{\gamma}$ where $\tilde{\gamma}$ is the constant given in Theorem 7, then we have*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{X}}_k\right)\right\|^2 + \frac{1}{K}\sum_{k=0}^{K-1}\left(1 - \frac{4\alpha(1+\tilde{\gamma})L}{1-\beta} - \frac{54\alpha^3 L^2}{(1-\beta)^3}\right)\mathbb{E}\left\|\overline{\boldsymbol{G}}_k\right\|^2$$

$$\leq\frac{2(1-\beta)\Delta}{\alpha K} + \frac{4\alpha\tilde{\gamma}\tilde{\sigma}^2 L}{(1-\beta)n} + \left(\frac{2L^2}{nK} + \frac{12\alpha L^2}{(1-\beta)nK}\right)\sum_{k=0}^{K-1}\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2 + \frac{54\alpha^3\tilde{\sigma}^2 L^2}{(1-\beta)^3 n}.$$

Proof We follow (Yu et al., 2019) and define an auxiliary sequence

$$\overline{\boldsymbol{Z}}_k = \begin{cases} \overline{\boldsymbol{X}}_k & k = 0, \\ \frac{1}{1-\beta}\overline{\boldsymbol{X}}_k - \frac{\beta}{1-\beta}\overline{\boldsymbol{X}}_{k-1} & k \geq 1 \end{cases}$$

We start from applying Taylor Theorem to $f(\overline{\boldsymbol{Z}}_k)$. Before that, we transform two terms as below. In the following analysis, we first bound all the terms with $k \geq 4$ to prevent negative subscripts. We will give the bounds for $k = 0, 1, 2, 3$ in the end.

55

First, for any $k \geq 4$,

$$
\begin{aligned}
\overline{\boldsymbol{Z}}_{k+1} - \overline{\boldsymbol{Z}}_k &= \frac{1}{1-\beta} \left(\overline{\boldsymbol{X}}_{k+1} - \overline{\boldsymbol{X}}_k \right) - \frac{\beta}{1-\beta} \left(\overline{\boldsymbol{X}}_k - \overline{\boldsymbol{X}}_{k-1} \right) \\
&\stackrel{(34)}{=} - \left(\frac{\alpha}{1-\beta} \overline{\boldsymbol{Y}}_k - \frac{\alpha\beta}{1-\beta} \overline{\boldsymbol{Y}}_{k-1} \right) \\
&\stackrel{(33)}{=} - \left(\frac{\alpha}{1-\beta} \overline{\boldsymbol{V}}_{k-1} - \frac{\alpha\beta}{1-\beta} \overline{\boldsymbol{V}}_{k-2} \right) \\
&= - \left[\frac{\alpha}{1-\beta} \left(\beta \overline{\boldsymbol{V}}_{k-2} + \overline{\widetilde{\boldsymbol{G}}}_{k-2} \right) - \frac{\alpha\beta}{1-\beta} \overline{\boldsymbol{V}}_{k-2} \right] \\
&= - \frac{\alpha}{1-\beta} \overline{\widetilde{\boldsymbol{G}}}_{k-2}.
\end{aligned}
$$

Additionally, note that for any $k \geq 4$,

$$
\begin{aligned}
\sum_{k=4}^{K-1} \left\| \overline{\boldsymbol{Z}}_k - \overline{\boldsymbol{X}}_k \right\|^2 &= \sum_{k=4}^{K-1} \left\| \frac{\beta}{1-\beta} \left(\overline{\boldsymbol{X}}_k - \overline{\boldsymbol{X}}_{k-1} \right) \right\|^2 \\
&= \sum_{k=4}^{K-1} \left\| \frac{\alpha\beta}{1-\beta} \overline{\boldsymbol{V}}_{k-2} \right\|^2 \\
&= \frac{\alpha^2\beta^2}{(1-\beta)^2} \sum_{k=4}^{K-1} \left\| \overline{\widetilde{\boldsymbol{G}}}_{k-3} + \sum_{j=0}^{k-4} \beta^{k-3-j} \overline{\widetilde{\boldsymbol{G}}}_j \right\|^2 \\
&\leq \frac{2\alpha^2\beta^2}{(1-\beta)^2} \sum_{k=4}^{K-1} \left\| \overline{\widetilde{\boldsymbol{G}}}_{k-3} \right\|^2 + \frac{2\alpha^2\beta^2}{(1-\beta)^2} \sum_{k=4}^{K-1} \left\| \sum_{j=0}^{k-4} \beta^{k-3-j} \overline{\widetilde{\boldsymbol{G}}}_j \right\|^2 \\
&\leq \frac{2\alpha^2\beta^2}{(1-\beta)^2} \sum_{k=4}^{K-1} \left\| \overline{\widetilde{\boldsymbol{G}}}_k \right\|^2 + \frac{2\alpha^2\beta^2}{(1-\beta)^2} \sum_{k=4}^{K-1} \sum_{j=0}^{k-4} \beta^{k-3-j} \left\| \overline{\widetilde{\boldsymbol{G}}}_j \right\|^2 \\
&\leq \frac{2\alpha^2\beta^2}{(1-\beta)^2} \sum_{k=4}^{K-1} \left\| \overline{\widetilde{\boldsymbol{G}}}_k \right\|^2 + \frac{2\alpha^2\beta^2}{(1-\beta)^2} \sum_{k=4}^{K-1} \left\| \overline{\widetilde{\boldsymbol{G}}}_j \right\|^2 \left(\sum_{j=0}^{\infty} \beta^j \right) \\
&\leq \frac{4\alpha^2\beta^2}{(1-\beta)^3} \sum_{k=4}^{K-1} \left\| \overline{\widetilde{\boldsymbol{G}}}_k \right\|^2.
\end{aligned}
$$

Now we can apply Taylor Theorem, for any $k \geq 4$,

$$
\begin{aligned}
&\mathbb{E} f \left(\overline{\boldsymbol{Z}}_{k+1} \right) \\
\leq\ &\mathbb{E} f \left(\overline{\boldsymbol{Z}}_k \right) + \mathbb{E} \left\langle \nabla f \left(\overline{\boldsymbol{Z}}_k \right), \overline{\boldsymbol{Z}}_{k+1} - \overline{\boldsymbol{Z}}_k \right\rangle + \frac{L}{2} \mathbb{E} \left\| \overline{\boldsymbol{Z}}_{k+1} - \overline{\boldsymbol{Z}}_k \right\|^2 \\
=\ &\mathbb{E} f \left(\overline{\boldsymbol{Z}}_k \right) - \frac{\alpha}{1-\beta} \mathbb{E} \left\langle \nabla f \left(\overline{\boldsymbol{Z}}_k \right), \overline{\widetilde{\boldsymbol{G}}}_{k-2} \right\rangle + \frac{\alpha^2 L}{2(1-\beta)^2} \mathbb{E} \left\| \overline{\widetilde{\boldsymbol{G}}}_{k-2} \right\|^2 \\
=\ &\mathbb{E} f \left(\overline{\boldsymbol{Z}}_k \right) - \frac{\alpha}{1-\beta} \mathbb{E} \left\langle \nabla f \left(\overline{\boldsymbol{Z}}_k \right) - \nabla f \left(\overline{\boldsymbol{X}}_k \right), \overline{\boldsymbol{G}}_{k-2} \right\rangle - \frac{\alpha}{1-\beta} \mathbb{E} \left\langle \nabla f \left(\overline{\boldsymbol{X}}_k \right), \overline{\boldsymbol{G}}_{k-2} \right\rangle
\end{aligned}
$$

$$+ \frac{\alpha^2 L}{2(1-\beta)^2} \mathbb{E}\left\|\overline{\tilde{\boldsymbol{G}}}_{k-2}\right\|^2$$

$$\leq \mathbb{E}f\left(\overline{\boldsymbol{Z}}_k\right) + \frac{(1-\beta)L}{2\beta^2}\mathbb{E}\left\|\overline{\boldsymbol{Z}}_k - \overline{\boldsymbol{X}}_k\right\|^2 + \frac{\alpha^2\beta^2 L}{2(1-\beta)^3}\mathbb{E}\left\|\overline{\tilde{\boldsymbol{G}}}_k\right\|^2 - \frac{\alpha}{2(1-\beta)}\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{X}}_k\right)\right\|^2$$

$$- \frac{\alpha}{2(1-\beta)}\mathbb{E}\left\|\overline{\boldsymbol{G}}_{k-2}\right\|^2 + \frac{\alpha}{2(1-\beta)}\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{X}}_k\right) - \overline{\boldsymbol{G}}_{k-2}\right\|^2 + \frac{\alpha^2 L}{2(1-\beta)^2}\mathbb{E}\left\|\overline{\tilde{\boldsymbol{G}}}_{k-2}\right\|^2$$

$$\leq \mathbb{E}f\left(\overline{\boldsymbol{Z}}_k\right) + \frac{(1-\beta)L}{2\beta^2}\mathbb{E}\left\|\overline{\boldsymbol{Z}}_k - \overline{\boldsymbol{X}}_k\right\|^2 + \frac{\alpha^2\beta^2 L}{2(1-\beta)^3}\mathbb{E}\left\|\overline{\tilde{\boldsymbol{G}}}_k\right\|^2 - \frac{\alpha}{2(1-\beta)}\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{X}}_k\right)\right\|^2$$

$$- \frac{\alpha}{2(1-\beta)}\mathbb{E}\left\|\overline{\boldsymbol{G}}_{k-2}\right\|^2 + \frac{\alpha}{1-\beta}\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{X}}_k\right) - \overline{\boldsymbol{G}}_{k+1}\right\|^2 + \frac{\alpha}{1-\beta}\mathbb{E}\left\|\overline{\boldsymbol{G}}_{k+1} - \overline{\boldsymbol{G}}_{k-2}\right\|^2$$

$$+ \frac{\alpha^2 L}{2(1-\beta)^2}\mathbb{E}\left\|\overline{\tilde{\boldsymbol{G}}}_{k-2}\right\|^2,$$

where in the fourth step, we apply for any $\boldsymbol{a}, \boldsymbol{b}$, it holds that $\langle \boldsymbol{a}, \boldsymbol{b}\rangle \leq \frac{c}{2}\|\boldsymbol{a}\|^2 + \frac{1}{2c}\|\boldsymbol{b}\|^2$ (for any $c > 0$) and $-2\langle \boldsymbol{a}, \boldsymbol{b}\rangle = -\|\boldsymbol{a}\|^2 - \|\boldsymbol{b}\|^2 + \|\boldsymbol{a} - \boldsymbol{b}\|^2$.

Similar to the analysis of Equation (29), we obtain for any $k \geq 0$,

$$\mathbb{E}\left\|\overline{\tilde{\boldsymbol{G}}}_k\right\|^2 \leq \mathbb{E}\left\|\overline{\boldsymbol{G}}_k\right\|^2 + \frac{\tilde{\sigma}^2}{n}.$$

The gradient difference term in the Taylor Theorem above can be written as

$$\frac{\alpha}{1-\beta}\mathbb{E}\left\|\overline{\boldsymbol{G}}_{k+1} - \overline{\boldsymbol{G}}_{k-2}\right\|^2$$

$$= \frac{\alpha}{1-\beta}\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{x}_{k,i}) - \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{x}_{k-3,i})\right\|^2$$

$$\leq \frac{\alpha}{(1-\beta)n}\sum_{i=1}^{n}\mathbb{E}\left\|\nabla f_i(\boldsymbol{x}_{k,i}) - \nabla f_i(\boldsymbol{x}_{k-3,i})\right\|^2$$

$$\leq \frac{3\alpha L^2}{(1-\beta)n}\mathbb{E}\left\|\boldsymbol{X}_k - \overline{\boldsymbol{X}}_k\mathbf{1}_n^\top\right\|_F^2 + \frac{3\alpha L^2}{(1-\beta)n}\mathbb{E}\left\|\boldsymbol{X}_{k-3} - \overline{\boldsymbol{X}}_{k-3}\mathbf{1}_n^\top\right\|_F^2$$

$$+ \frac{3\alpha L^2}{(1-\beta)n}\mathbb{E}\left\|\overline{\boldsymbol{X}}_k\mathbf{1}_n^\top - \overline{\boldsymbol{X}}_{k-3}\mathbf{1}_n^\top\right\|_F^2.$$

Put them back, and sum over from $k = 4$ to $K - 1$,

$$\frac{\alpha}{2(1-\beta)}\sum_{k=4}^{K-1}\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{X}}_k\right)\right\|^2 + \frac{\alpha}{2(1-\beta)}\mathbb{E}\sum_{k=4}^{K-1}\left\|\overline{\boldsymbol{G}}_{k-2}\right\|^2$$

$$\leq \sum_{k=4}^{K-1}\left(\mathbb{E}f\left(\overline{\boldsymbol{Z}}_k\right) - \mathbb{E}f\left(\overline{\boldsymbol{Z}}_{k+1}\right)\right) + \frac{(1-\beta)L}{2\beta^2}\sum_{k=4}^{K-1}\mathbb{E}\left\|\overline{\boldsymbol{Z}}_k - \overline{\boldsymbol{X}}_k\right\|^2 + \frac{\alpha^2\beta^2 L}{(1-\beta)^3}\sum_{k=4}^{K-1}\mathbb{E}\left\|\overline{\tilde{\boldsymbol{G}}}_k\right\|^2$$

$$+ \frac{\alpha}{1-\beta}\sum_{k=4}^{K-1}\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{X}}_k\right) - \overline{\boldsymbol{G}}_{k+1}\right\|^2 + \frac{\alpha}{1-\beta}\sum_{k=4}^{K-1}\mathbb{E}\left\|\overline{\boldsymbol{G}}_{k+1} - \overline{\boldsymbol{G}}_{k-2}\right\|^2$$

$$\leq f(\overline{\boldsymbol{Z}}_4) - f^* + \frac{2\alpha^2(1+\tilde{\gamma})L}{(1-\beta)^2}\sum_{k=4}^{K-1}\mathbb{E}\left\|\overline{\boldsymbol{G}}_k\right\|^2 + \frac{\alpha^2\tilde{\gamma}\tilde{\sigma}^2 L(K-4)}{(1-\beta)^2 n}$$

$$
+ \frac{\alpha L^2}{(1-\beta)n} \sum_{k=4}^{K-1} \mathbb{E} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2 + \frac{6\alpha^2 L^2}{(1-\beta)^2 n} \sum_{k=4}^{K-1} \mathbb{E} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2
$$

$$
+ \frac{3\alpha^2 L^2}{(1-\beta)^2} \sum_{k=4}^{K-1} \mathbb{E} \left\| \overline{\boldsymbol{X}}_k - \overline{\boldsymbol{X}}_{k-3} \right\|^2
$$

$$
\leq f(\overline{\boldsymbol{Z}}_4) - f^* + \frac{2\alpha^2(1+\tilde{\gamma})L}{(1-\beta)^2} \sum_{k=4}^{K-1} \mathbb{E} \left\| \overline{\boldsymbol{G}}_k \right\|^2 + \frac{\alpha^2 \tilde{\gamma} \tilde{\sigma}^2 L(K-4)}{(1-\beta)^2 n}
$$

$$
+ \left(\frac{\alpha L^2}{(1-\beta)n} + \frac{6\alpha^2 L^2}{(1-\beta)^2 n} \right) \sum_{k=4}^{K-1} \mathbb{E} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2
$$

$$
+ \frac{3\alpha^4 L^2}{(1-\beta)^2} \sum_{k=4}^{K-1} \mathbb{E} \left\| \overline{\boldsymbol{V}}_{k-2} + \overline{\boldsymbol{V}}_{k-3} + \overline{\boldsymbol{V}}_{k-4} \right\|^2 .
$$

To deal with the last term, note that

$$
\mathbb{E} \left\| \overline{\boldsymbol{V}}_k \right\|^2 = \mathbb{E} \left\| \beta \overline{\boldsymbol{V}}_{k-1} + (1-\beta) \frac{\overline{\tilde{\boldsymbol{G}}}_{k-1}}{1-\beta} \right\|^2
$$

$$
\leq \beta \mathbb{E} \left\| \overline{\boldsymbol{V}}_{k-1} \right\|^2 + \frac{1}{1-\beta} \mathbb{E} \left\| \overline{\tilde{\boldsymbol{G}}}_{k-1} \right\|^2
$$

Summing from $k=0$ to $K-1$, we obtain

$$
\sum_{k=0}^{K-1} \mathbb{E} \left\| \overline{\boldsymbol{V}}_k \right\|^2 \leq \beta \sum_{k=0}^{K-1} \mathbb{E} \left\| \overline{\boldsymbol{V}}_{k-1} \right\|^2 + \frac{1}{1-\beta} \sum_{k=0}^{K-1} \mathbb{E} \left\| \overline{\tilde{\boldsymbol{G}}}_{k-1} \right\|^2
$$

$$
\leq \beta \sum_{k=0}^{K-1} \mathbb{E} \left\| \overline{\boldsymbol{V}}_k \right\|^2 + \frac{1}{1-\beta} \sum_{k=0}^{K-1} \mathbb{E} \left\| \overline{\tilde{\boldsymbol{G}}}_k \right\|^2 .
$$

Rearrange the terms, we obtain

$$
\sum_{k=0}^{K-1} \mathbb{E} \left\| \overline{\boldsymbol{V}}_k \right\|^2 \leq \frac{1}{(1-\beta)^2} \sum_{k=0}^{K-1} \mathbb{E} \left\| \overline{\tilde{\boldsymbol{G}}}_k \right\|^2 .
$$

Combining with Lemma 13 and rearrange the terms we obtain

$$
\frac{1}{K} \sum_{k=0}^{K-1} \mathbb{E} \left\| \nabla f\left(\overline{\boldsymbol{X}}_k \right) \right\|^2 + \frac{1}{K} \sum_{k=0}^{K-1} \left(1 - \frac{4\alpha(1+\tilde{\gamma})L}{1-\beta} - \frac{54\alpha^3 L^2}{(1-\beta)^3} \right) \mathbb{E} \left\| \overline{\boldsymbol{G}}_k \right\|^2
$$

$$
\leq \frac{2(1-\beta)\Delta}{\alpha K} + \frac{4\alpha \tilde{\gamma} \tilde{\sigma}^2 L}{(1-\beta)n} + \left(\frac{2L^2}{nK} + \frac{12\alpha L^2}{(1-\beta)nK} \right) \sum_{k=0}^{K-1} \mathbb{E} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \mathbf{1}_n^\top \right\|_F^2 + \frac{54\alpha^3 \tilde{\sigma}^2 L^2}{(1-\beta)^3 n} .
$$

That completes the proof.

∎

58

Lemma 13 *When $k = 0, 1, 2, 3$, in Lemma 12 it fulfills that:*

$$\frac{\alpha}{2(1-\beta)} \sum_{k=0}^{3} \mathbb{E} \left\| \nabla f \left(\overline{\boldsymbol{X}}_k \right) \right\|^2 \leq f(\boldsymbol{0}) - f^* + \frac{2\alpha^2(1+\tilde{\gamma})L}{(1-\beta)^2} \sum_{k=0}^{3} \mathbb{E} \left\| \overline{\boldsymbol{G}}_k \right\|^2 + \frac{4\alpha^2 \tilde{\gamma} \tilde{\sigma}^2 L}{(1-\beta)^2 n}$$

$$+ \left(\frac{\alpha L^2}{(1-\beta)n} + \frac{6\alpha^2 L^2}{(1-\beta)^2 n} \right) \sum_{k=0}^{3} \mathbb{E} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \boldsymbol{1}_n^\top \right\|_F^2 .$$

Proof For simplicity, it'd be helpful to see how sequences $\{\overline{\boldsymbol{X}}_k\}$, $\{\overline{\boldsymbol{Y}}_k\}$, $\{\overline{\boldsymbol{V}}_k\}$ and $\{\overline{\boldsymbol{Z}}_k\}$ change at $k = 0, 1, 2, 3, 4$.

Following their definitions, we can get when $k = 0$,

$$\overline{\boldsymbol{V}}_0 = \boldsymbol{0}$$
$$\overline{\boldsymbol{Y}}_0 = \boldsymbol{0}$$
$$\overline{\boldsymbol{X}}_0 = \boldsymbol{0}$$
$$\overline{\boldsymbol{Z}}_0 = \overline{\boldsymbol{X}}_0 = \boldsymbol{0}.$$

We additionally define $\tilde{\boldsymbol{G}}_0 = \boldsymbol{0}$ and $\boldsymbol{V}_{-1} = \boldsymbol{0}$ for simplicity.

Then when $k = 1$,

$$\overline{\boldsymbol{V}}_1 = \beta \overline{\boldsymbol{V}}_0 + \tilde{\boldsymbol{G}}_0 = \boldsymbol{0}$$
$$\overline{\boldsymbol{Y}}_1 = \overline{\boldsymbol{Y}}_0 + \overline{\boldsymbol{V}}_0 - \overline{\boldsymbol{V}}_{-1} = \boldsymbol{0}$$
$$\overline{\boldsymbol{X}}_1 = \overline{\boldsymbol{X}}_0 - \alpha \overline{\boldsymbol{Y}}_0 = \boldsymbol{0}$$
$$\overline{\boldsymbol{Z}}_1 = \frac{1}{1-\beta} \overline{\boldsymbol{X}}_1 - \frac{\beta}{1-\beta} \overline{\boldsymbol{X}}_0 = \boldsymbol{0}.$$

When $k = 2$,

$$\overline{\boldsymbol{V}}_2 = \beta \overline{\boldsymbol{V}}_1 + \tilde{\boldsymbol{G}}_1 = \tilde{\boldsymbol{G}}_1$$
$$\overline{\boldsymbol{Y}}_2 = \overline{\boldsymbol{Y}}_1 + \overline{\boldsymbol{V}}_1 - \overline{\boldsymbol{V}}_0 = \boldsymbol{0}$$
$$\overline{\boldsymbol{X}}_2 = \overline{\boldsymbol{X}}_1 - \alpha \overline{\boldsymbol{Y}}_1 = \boldsymbol{0}$$
$$\overline{\boldsymbol{Z}}_2 = \frac{1}{1-\beta} \overline{\boldsymbol{X}}_2 - \frac{\beta}{1-\beta} \overline{\boldsymbol{X}}_1 = \boldsymbol{0}.$$

When $k = 3$,

$$\overline{\boldsymbol{V}}_3 = \beta \overline{\boldsymbol{V}}_2 + \tilde{\boldsymbol{G}}_2 = \beta \tilde{\boldsymbol{G}}_1 + \tilde{\boldsymbol{G}}_2$$
$$\overline{\boldsymbol{Y}}_3 = \overline{\boldsymbol{Y}}_2 + \overline{\boldsymbol{V}}_2 - \overline{\boldsymbol{V}}_1 = \tilde{\boldsymbol{G}}_1$$
$$\overline{\boldsymbol{X}}_3 = \overline{\boldsymbol{X}}_2 - \alpha \overline{\boldsymbol{Y}}_2 = \boldsymbol{0}$$
$$\overline{\boldsymbol{Z}}_3 = \frac{1}{1-\beta} \overline{\boldsymbol{X}}_3 - \frac{\beta}{1-\beta} \overline{\boldsymbol{X}}_2 = \boldsymbol{0}.$$

Finally, when $k = 4$,

$$\overline{\boldsymbol{V}}_4 = \beta \overline{\boldsymbol{V}}_3 + \tilde{\boldsymbol{G}}_3 = \beta^2 \tilde{\boldsymbol{G}}_1 + \beta \tilde{\boldsymbol{G}}_2 + \tilde{\boldsymbol{G}}_3$$

59

$$\overline{\boldsymbol{Y}}_4 = \overline{\boldsymbol{Y}}_3 + \overline{\boldsymbol{V}}_3 - \overline{\boldsymbol{V}}_2 = \beta \tilde{\boldsymbol{G}}_1 + \tilde{\boldsymbol{G}}_2$$

$$\overline{\boldsymbol{X}}_4 = \overline{\boldsymbol{X}}_3 - \alpha \overline{\boldsymbol{Y}}_3 = -\alpha \tilde{\boldsymbol{G}}_1$$

$$\overline{\boldsymbol{Z}}_4 = \frac{1}{1-\beta} \overline{\boldsymbol{X}}_4 - \frac{\beta}{1-\beta} \overline{\boldsymbol{X}}_3 = -\frac{\alpha}{1-\beta} \tilde{\boldsymbol{G}}_1.$$

Fitting these values into the analysis of Lemma 12, it is not hard to verify that

$$\frac{\alpha}{2(1-\beta)} \sum_{k=0}^{3} \mathbb{E} \left\| \nabla f\left(\overline{\boldsymbol{X}}_k\right) \right\|^2 \le f(\boldsymbol{0}) - f^* + \frac{2\alpha^2(1+\tilde{\gamma})L}{(1-\beta)^2} \sum_{k=0}^{3} \mathbb{E} \left\| \overline{\boldsymbol{G}}_k \right\|^2 + \frac{4\alpha^2\tilde{\gamma}\tilde{\sigma}^2 L}{(1-\beta)^2 n}$$

$$+ \left(\frac{\alpha L^2}{(1-\beta)n} + \frac{6\alpha^2 L^2}{(1-\beta)^2 n} \right) \sum_{k=0}^{3} \mathbb{E} \left\| \boldsymbol{X}_k - \overline{\boldsymbol{X}}_k \boldsymbol{1}_n^\top \right\|_F^2.$$

We omit the derivation for brevity. That completes the proof. ■

Appendix C. Details to footnotes

C.1 Asynchronous Algorithm (Footnote 2)

In the full paper, we focus on the synchronous algorithms, i.e., we assume the existence of a synchronization process among workers between two adjacent iterations. We now extend our formulation to asynchronous algorithms. Since workers now update and communicate asynchronously, we define any gradient update that took place on a randomly chosen worker as one iteration. This randomness depends on system implementation, stochastic events, etc. This is a commonly adopted definition in the analysis of (decentralized) asynchronous algorithms Lian et al. (2017b). To obtain a lower bound in such case, consider the two settings as shown in the proof of Theorem 1. In setting 1, it can be easily verified that the lower bound for sample complexity is

$$\Omega\left(\frac{\Delta L \sigma^2}{B \epsilon^4}\right). \tag{38}$$

This holds because in the extreme case, only one worker is making contributions to the optimization. And since we have not made any assumption on how workers are sampled to conduct the next iteration, this is a valid bound for arbitrary distribution. On the other hand, considering setting 2, the lower bound is still $\Omega(T_0 D)$ where $T_0 = \Omega(\Delta L \epsilon^{-2})$ is the lower bound in the sequential case, since the systems need at least $\Omega(D)$ iterations for the workers in I_0 and I_2 to contact. The lower bound for communication complexity is then

$$\Omega\left(\frac{\Delta L D}{\epsilon^2}\right). \tag{39}$$

Combining them together, we can get the final lower bound as:

$$\Omega\left(\frac{\Delta L \sigma^2}{B \epsilon^4} + \frac{\Delta L D}{\epsilon^2}\right). \tag{40}$$

Note that this bound holds with probability 1. It is possible to propose finer-grained assumption on how workers are chosen (e.g. uniformly random) and use concentration inequalities (e.g. Hoeffding's inequality) to get tighter bounds, we leave this as future work.

C.2 Relax zero-respecting assumption (Footnote 3)

To relax the zero-respecting assumption, we can use the technique proposed by Carmon et al. (2019) (See their proofs to Proposition 1 and 2). The basic idea is that to adversarially construct the loss function and rotate the non-zero coordinates in t-th iterations, such that when the algorithm operates on the rotated function, the first t iterations match with that of the old function. However, the new rotated function is still zero-respecting to the algorithm after t-th iteration so is generally hard to optimize. The details can be found in Carmon et al. (2019).

C.3 Specific algorithm for Average Consensus (Footnote 8)

Many algorithms have been proposed on solving the Average Consensus problem, readers can find details in many previous works on graph theory such as Georgopoulos (2011); Hendrickx et al. (2014); Ko (2010). A straightforward algorithm is the Minimum Spanning Tree, that is, we first generate a spanning tree of the graph, and then the workers send and receive message using propagation on the tree. Specifically, starting from the leaves, all the children nodes of the tree send its accumulated value to the parents and the root compute the averaged value after gathering the information from the graph. And then reversely, the parent nodes send the value back to the child nodes and eventually all the nodes will get the averaged value. This algorithm is also known as the `GATHER-PROPAGATE` algorithm as discussed in Ko (2010), section 3. We include the detailed pseudo-code[13] in Algorithm 5.

---

13. This code is proposed by Ko (2010), we do not intend to take credit for this.

Algorithm 5 GATHER-PROPAGATE (Spanning Tree) for a single coordinate

---

Input: communication graph G, a single coordinate on workers (all the coordinates follow the same instructions) to be communicated $\boldsymbol{X} \in \mathbb{R}^n$.

1: $\boldsymbol{d} \leftarrow$ vector of 1's indexed by $V(G)$ (vertices set of graph G).
2: $\mathcal{I} \leftarrow$ a spnning tree of G with root r arbitrarily picked.
3: **for** $v \in V(\mathcal{I})$ **do**
4:     $l_v \leftarrow \bar{D}(r, v)$ (the distance between r and v)
5: **end for**
6: **for** $\alpha = \max_v l_v, \cdots, 1$ **do**
7:     **for** v with $l_v = \alpha$ **do**
8:         v gives all its value onto its parents u:

$$\begin{bmatrix} \boldsymbol{X}_u \\ \boldsymbol{X}_v \end{bmatrix} \leftarrow \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_u \\ \boldsymbol{X}_v \end{bmatrix}$$

9:         $\boldsymbol{d}_u \leftarrow \boldsymbol{d}_u + \boldsymbol{d}_v$
10:     **end for**
11: **end for**
12: **for** $\alpha = 0, \cdots, \max_v l_v - 1$ **do**
13:     **for** u with $l_u = \alpha$ **do**
14:         $\{v_1, \cdots, v_\beta\} \leftarrow$ set of children of u
15:         re-distribute the results:

$$\begin{bmatrix} \boldsymbol{X}_u \\ \boldsymbol{X}_{v_1} \\ \vdots \\ \boldsymbol{X}_{v_\beta} \end{bmatrix} \leftarrow \frac{1}{\boldsymbol{d}_u} \begin{bmatrix} \boldsymbol{d}_u - \boldsymbol{d}_u - \cdots - \boldsymbol{d}_{v_\beta} \\ \boldsymbol{d}_{v_1} \\ \vdots \\ \boldsymbol{d}_{v_\beta} \end{bmatrix} \boldsymbol{X}_u$$

16:     **end for**
17: **end for**
18: **return** $X \frac{\mathbf{1}\mathbf{1}^\top}{n}$