On Graphs with Finite-Time Consensus and Their Use in Gradient Tracking

Edward Duc Hien Nguyen* Xin Jiang† Bicheng Ying‡ César A. Uribe*

October 30, 2023

Abstract

This paper studies sequences of graphs satisfying the finite-time consensus property and their use in Gradient Tracking. We provide an explicit weight matrix representation of the studied sequences and prove their finite-time consensus property (*i.e.*, iterating through such a finite sequence is equivalent to performing global or exact averaging). Moreover, we incorporate the studied finite-time consensus topologies into Gradient Tracking and present a new algorithmic scheme called Gradient Tracking for Finite-Time Consensus Topologies (GT-FT). We analyze the new scheme for nonconvex problems with stochastic gradient estimates. Our analysis shows that the convergence rate of GT-FT does not depend on the heterogeneity of the agents' functions or the connectivity of any individual graph in the topology sequence. Furthermore, owing to the sparsity of the graphs, GT-FT requires lower communication costs than Gradient Tracking using the static counterpart of the topology sequence.

1 Introduction

We study the decentralized solution of optimization problems of the form

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad f(x) \triangleq \frac{1}{n}\sum_{i=1}^{n} f_i(x), \quad \text{where } f_i(x) \triangleq \mathbb{E}_{\xi_i}[F_i(x;\xi_i)], \tag{1}$$

and $f_i \colon \mathbb{R}^d \to \mathbb{R}$ is a smooth, possibly nonconvex function. The symbol $\mathbb{E}_{\xi_i}$ denotes the expected value of the random variable or data ξ_i associated with the probability space $\{\Omega_i, \mathcal{F}_i, \mathbb{P}_i\}$. Hence, f_i is defined as the expected value of some loss function $F_i(\cdot, \xi_i)$ over ξ_i. Solving Problem (1) using traditional gradient descent methods can incur high computational costs as the gradient oracle must be accessed at each iteration, and the cost of gradient computation increases with the amount of data (Woodworth et al., 2018). Decentralized stochastic gradient methods are an alternative approach in which each function f_i and probability space $\{\Omega_i, \mathcal{F}_i, \mathbb{P}_i\}$ are assigned to be held exclusively and privately by an agent (node), and agents use stochastic gradient estimates of f_i. This parallelizes the cost of accessing the gradient oracle across the agents. Consequently, agents must cooperate and communicate with one another according to a certain network topology.

Previously, a considerable amount of work in decentralized optimization is tailored to applications such as wireless communications, power systems, and sensor networks (Cui et al., 2007; Kar and Moura, 2009; 2010; Zhang et al., 2015). Network topologies for these applications are either static, unknown, or defined beforehand. Moreover, communication might be fragile, and conservative worst-case scenarios must be considered. We instead focus on the high-performance computing scenario in which agents are abstractions

*Department of Electrical and Computer Engineering, Rice University. Email: en18@rice.edu, cauribe@rice.edu.

†Department of Industrial and Systems Engineering, Lehigh University. Email: xjiang@lehigh.edu.

‡Google Inc. Email: ybc@google.com.

of computing resources. Under this scenario, the communication links between agents are robust, and the network topology can be flexibly and cheaply rearranged (Jouppi et al., 2023).

In view of the flexibility in the design of network structure, careful selection of sparse topologies can reduce communication costs (Assran et al., 2019; Ding et al., 2023; Lan et al., 2018; Ying et al., 2021). A simple choice of network structure for communication is to allow global coordination across all agents through Parameter–Server (Li et al., 2014) or Ring–Allreduce (Patarasuk and Yuan, 2009). Yet, both strategies are not scalable. Global coordination following the Parameter–Server framework incurs significant bandwidth costs, while following the Ring–Allreduce protocol incurs high latency. An alternative approach to reduce communication costs is to use a static, sparse topology such as rings or static exponential graphs. Under this setup, communication costs are reduced as agents only communicate with their direct neighbors, and only local information is needed. However, the benefits of reduced communication costs come at a price of slower convergence when considering their implementation into decentralized algorithms (Nedić et al., 2018).

In this paper, we address the trade-off between communication costs and convergence rate in decentralized optimization algorithms and propose to use sequences of deterministic topologies that satisfy the *finite-time consensus* property. Topology sequences with the finite-time consensus property have the desirable feature that iterating through the entire graph sequence is equivalent to performing global or exact averaging. Moreover, each of the individual graphs in such a sequence is typically sparse (Shi et al., 2016; Takezawa et al., 2023; Ying et al., 2021) and, when used in a decentralized algorithm, requires limited communication costs at each iteration. We study several classes of topology sequences for which this seemingly restrictive requirement holds, including the static de Bruijn graph (de Bruijn, 1946; Delvenne et al., 2009), one-peer hyper-cubes (Shi et al., 2016), one-peer exponential graphs (Ying et al., 2021), and p-peer hyper-cuboids.

Despite the existence of various topology sequences with the finite-time consensus property, directly incorporating them into decentralized algorithms is not straightforward. Classical analyses of decentralized optimization algorithms assume, for example, symmetry and strong connectivity of the mixing matrices, which certain elements in a topology sequence with the finite-time consensus property might violate. The usefulness of finite-time consensus graphs in decentralized optimization was first examined in Ying et al. (2021), where the Decentralized Momentum Stochastic Gradient (DmSGD) method is applied with one-peer exponential graphs. The authors of Ying et al. (2021) show that the convergence rate of DmSGD using one-peer exponential graphs is the same as that using static exponential graphs. This equivalence is crucial as the communication cost of a one-peer exponential graph is significantly lower than that of a static exponential graph. Despite the simplicity of Decentralized Stochastic Gradient methods, their analysis requires making an assumption that bounds the heterogeneity between agents' local functions (see; *e.g.*, Ying et al. (2021)). An alternative algorithm class called exact or bias-corrected methods has been proposed to overcome this limitation so that convergence is achieved independent of the magnitude of the heterogeneity. Examples of these algorithms include EXTRA (Shi et al., 2015), Exact Diffusion (Yuan et al., 2019), and Gradient Tracking (GT) methods (Nedić et al., 2017). However, existing analysis for these methods breaks when finite-time consensus graphs are used in the algorithms. For example, recent tight analysis on GT methods (Alghunaim and Yuan, 2022; 2023; Koloskova et al., 2021) assumes static, strongly connected topologies described by a symmetric mixing matrix, which limits the direct extension of their analysis for the time-varying setting with possibly disconnected instantaneous communications.

Contributions. The contribution of this work is two-fold. First, we study several sequences of graphs that satisfy the finite-time consensus property. For one-peer exponential graphs presented in Ying et al. (2021), we present a simplified proof for its finite-time consensus property when the number of agents is a power of 2. For an arbitrary number of agents, we present the sequence of graphs called p-peer hyper-cuboids, and establish their finite-time consensus property. We also show that in certain cases, p-peer hyper-cuboids are permutation equivalent to the well-studied de Bruijn graphs.

Moreover, we incorporate the studied topology sequences into the Gradient Tracking (GT) algorithm and present a new algorithmic scheme called Gradient Tracking for Finite-Time Consensus Topologies (GT-FT). We present convergence analysis for GT-FT, with further stepsize tuning and a simple warm-up technique. The analysis relies on a new, Gauss–Seidel reformulation of the GT updates, and remarkably, the established convergence rate is independent of the connectivity of any of the individual graphs used in the algorithm. Furthermore, we show that GT-FT using one-peer exponential graphs has slightly lower iteration complexity than GT using the static exponential graph. Considering the decentralized manner of the algorithms, it suggests that GT-FT has significantly lower communication costs compared with the static counterpart.

Outline. The rest of the paper is organized as follows. Section 2 discusses prior work on decentralized optimization algorithms and existing sequences of topologies satisfying the finite-time consensus property. Section 3 formally defines the finite-time consensus property and presents various topology sequences with this property. Section 4 includes a description of the Time-Varying Gradient Tracking (TV-GT) algorithm and our modified version: Gradient Tracking for Finite-Time Consensus Graphs (GT-FT). Section 5 presents convergence analysis of GT-FT under the nonconvex setting where agents only have access to stochastic gradient estimates. Numerical experiments in Section 6 verify the finite-time consensus property of the topology sequences studied in Section 3 and the algorithm analysis in Section 5.

2 Related Work

We begin reviewing various decentralized optimization algorithms and describe the scope and settings for their analysis. We highlight various works analyzing Gradient Tracking methods and detail why their analysis cannot be trivially extended to sequences of topologies that satisfy the finite-time consensus property. We then discuss prior work that studied sequences of topologies that satisfy the finite-time consensus property.

Many algorithms have been proposed to solve Problem (1) in a decentralized manner. We focus on *stochastic* decentralized algorithms in which agents cannot compute the full gradient of their local objective function. Among these algorithms, the most famous ones include Decentralized Stochastic Gradient (DSGD) methods (Cattivelli and Sayed, 2010; Lopes and Sayed, 2007; Sundhar Ram et al., 2010), EXTRA (Shi et al., 2015), Exact Diffusion/D^2/NIDS (Li et al., 2019; Tang et al., 2018; Yuan et al., 2019), and Gradient Tracking (Di Lorenzo and Scutari, 2016; Nedić et al., 2017; Qu and Li, 2018; Xu et al., 2015). (Accelerated variants of decentralized algorithms are beyond the scope of this work and left for future work.) Below, we briefly discuss current analytical findings on these algorithms that motivate our investigation into Gradient Tracking methods.

DSGD is arguably the most popular and widely studied decentralized algorithm due to its simplicity and communication efficiency. However, analysis of DSGD has revealed several of the algorithm's limitations. For example, DSGD requires a bounded heterogeneity assumption, which bounds the allowed difference between local functions/data. One typical formulation of the heterogeneity assumption for nonconvex problems used in recent analyses, such as the analysis done in Koloskova et al. (2020), reads as

$$\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x)\|_2^2 \le \hat{\zeta}^2 + P\|\nabla f(x)\|^2, \quad \text{for all } x \in \mathbb{R}^d,$$

where $(P, \hat{\zeta}) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0}$ are two constants.

The effect of heterogeneity is further magnified by large and sparse topologies, which further degrades the performance of DSGD. Both phenomena are captured in the work by Koloskova et al. (2020) where for nonconvex problems in which agents use stochastic gradients, they can, as one special scenario, derive the iteration complexity

$$O\left(\frac{L\hat{\sigma}^2}{n\epsilon} + \frac{L(\hat{\zeta}\sqrt{M+1} + \hat{\sigma}\sqrt{p})}{p\epsilon^{3/2}} + \frac{L\sqrt{(P+1)(M+1)}}{p\epsilon}\right) \tag{2}$$

to find an ϵ-first-order stationary point. The constant $p < 1$ is one minus the spectral gap of a static, doubly-stochastic, and symmetric mixing matrix $W \in \mathbb{R}^{n\times n}$, which describes the network topology. The constants $\hat{\sigma}^2, M$ are derived from their assumption on the gradient noise:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\xi_i}\|\nabla F_i(x_i,\xi_i) - \nabla f_i(x_i)\|_2^2 \le \hat{\sigma}^2 + \frac{M}{n}\sum_{i=1}^{n}\|\nabla f_i(x_i)\|_2^2,$$

for all $\{x_i\}_{i=1}^n \subset \mathbb{R}^d$. The constant L is the Lipschitz constant. The notation $x^{(0)}$ indicates the initial parameter of all agents, and f^* is the optimal value of Problem (1). It follows from (2) that larger heterogeneity or a smaller spectral gap will increase the number of iterations required to obtain the desired accuracy. The iteration complexity of DSGD increases as the amount of heterogeneity $\hat{\zeta}, P$ increases or as the connectivity of the network decreases, *i.e.*, $p \to 0$.

Exact or bias-corrected decentralized algorithms have been proposed to circumvent the negative effects that heterogeneity has on the convergence of DSGD. Examples of these algorithms include EXTRA (Shi et al., 2015), Exact Diffusion (Yuan et al., 2019), and Gradient Tracking methods (Nedić et al., 2017). The aforementioned three algorithms have been studied in a unified framework in Alghunaim et al. (2021) and Alghunaim and Yuan (2022). The analysis in Alghunaim et al. (2021) and Alghunaim and Yuan (2022) on EXTRA, Exact Diffusion, and Gradient Tracking reveals that the bounded heterogeneity assumption is not necessary. Furthermore, the convergence rate of these algorithms is dependent on the heterogeneity at the initial point.

The convergence of EXTRA and Exact Diffusion has only been established for symmetric and static mixing matrices (Alghunaim and Yuan, 2022). We specifically choose to focus on Gradient Tracking methods because GT methods have been shown to converge for various scenarios such as directed graphs (Pu et al., 2020; Xi et al., 2018) and time-varying graphs (Scutari and Sun, 2019). We note that the recent analyses on Gradient Tracking methods (Alghunaim and Yuan, 2022; 2023; Koloskova et al., 2021) have comprehensively covered the static, connected topology case. Their analyses cannot readily be extended to the time-varying case as they all rely on assuming the symmetry of the mixing matrix. Even if one assumes that every mixing matrix of a sequence of network topologies is symmetric, it does not necessarily hold that the product of these matrices is also symmetric.

Gradient tracking with time-varying topologies (TV-GT) has been studied in other literature under various assumptions. Nedić et al. (2017) analyze TV-GT, which they call DIGing, for the strongly-convex deterministic scenario and with τ-connected graphs. The assumption of τ-connected graphs means that the union of a sequence of τ-length graphs is connected, and is weaker than assuming that the topology at every iteration is connected. Another example of TV-GT is the algorithm NEXT (Di Lorenzo and Scutari, 2016) (and its extension SONATA (Scutari and Sun, 2019)), which are analyzed for the nonconvex multi-agent composite optimization problems in the deterministic scenario and with τ-connected graphs. We note that DIGing (Nedić et al., 2017), NEXT (Di Lorenzo and Scutari, 2016), and SONATA (Scutari and Sun, 2019) do not cover the case in which agents can only have access to stochastic gradient estimates of their local objective functions. Moreover, Song et al. (2022) provide convergence analysis for TV-GT under the nonconvex and stochastic setting and make no assumption on the symmetry of the mixing matrices. Nevertheless, they only consider the graph with the smallest connectivity (in expectation) of the set of all time-varying topologies used in the algorithm. This does not cover deterministic sequences of topologies that satisfy the finite-time consensus property where certain elements of the sequence are always disconnected.

Table 1: Summary of analyses for Time-Varying Gradient Tracking for the nonconvex, stochastic case.

Reference	Nonconvex	Stochastic Gradients	Network Class	Iteration Complexity[a]
Song et al. (2022)	✓	✓	Strongly Connected	$O\left(\frac{\sigma^2}{n\epsilon^2}\right) + O\left(\frac{\sigma}{(1-\lambda)^{3/2}\epsilon^{3/2}}\right) + O\left(\frac{1}{(1-\lambda)^2\epsilon}\right)$[b]
Our Work	✓	✓	Finite-Time Consensus	$O\left(\frac{L\sigma}{n\epsilon^2}\right) + O\left(\frac{\tau^{3/2}L\sigma}{\epsilon^{3/2}}\right) + O\left(\frac{\tau^2 L}{\epsilon}\right)$[c]

[a] These algorithms follow the Adapt-then-Combine update scheme so the computational and communication iteration complexities are the same.

[b] The convergence rate in this paper omits the Lipschitz constant from their rate. The constant λ is the mixing rate of the mixing matrix associated with the graph with the smallest connectivity (in expectation) of the set of all time-varying topologies used in the algorithm. The constant σ is from the bounded variance assumption imposed on the gradient noise.

[c] The constant τ is the length of the sequence of topologies with the finite-time consensus property, σ is from the bounded variance assumption imposed on the gradient noise, and L is the Lipschitz constant.

Topologies with the so-called *finite-time consensus property* have been studied across various works but have only recently revealed their usefulness for decentralized optimization. Delvenne et al. (2009) prove that a sequence of static de Bruijn graphs can achieve finite-time convergence. Shi et al. (2016) study (symmetric and asymmetric) gossip algorithms with finite-time convergence, and they provide hyper-cubes as an example of finite-time consensus graphs. However, both papers do not study the use of finite-time consensus property in decentralized optimization algorithms. Assran et al. (2019) first propose to decompose a static exponential graph into a sequence of directed one-peer exponential graphs, and then the "Push-Sum" communication protocol (proposed by Nedić et al. (2018)) is leveraged to enable the use of directed graphs in decentralized algorithms. Yet, Assran et al. (2019) do not prove the finite-time consensus property for the proposed graphs, and their analysis requires the bounded heterogeneity assumption. Further investigation into the one-peer exponential graphs is conducted by Ying et al. (2021), where the authors prove that sequences of one-peer exponential graphs with 2^τ agents ($\tau \in \mathbb{N}_{\geq 1}$) satisfy the finite-time consensus property. In addition, Ying et al. (2021) incorporate the one-peer exponential graphs into the Decentralized Momentum Stochastic Gradient (DmSGD) method. The authors show that the convergence rate of DmSGD using one-peer exponential graphs is the same as DmSGD using static exponential graphs. The equivalence of the convergence rate is significant because the communication cost of a one-peer exponential graph is much lower than that of a static exponential graph. We note that the final convergence rate derived by Ying et al. (2021) fails to consider the requirement on stepsizes imposed by their analysis, and thus, their result is incomparable to ours. Recent work by Takezawa et al. (2023) proposes the k-peer hyper-hypercubes and base-$(k+1)$ graphs and claims that both satisfy the finite-time consensus property for an arbitrary number of nodes. However, Takezawa et al. (2023) only present a constructive approach to building the graphs and do not give an explicit matrix representation of the corresponding weight/mixing matrices. Although numerical evidence provided in Takezawa et al. (2023) demonstrates the finite-time consensus property of their proposed base-$(k+1)$ graphs, theoretical justification is still lacking and requires further investigation. Last but not least, we remark that the scenarios studied in Assran et al. (2019); Takezawa et al. (2023); Ying et al. (2021) all require the bounded heterogeneity assumption due to their choice to study DSGD methods.

3 Finite-Time Consensus

In this section, we formally define the finite-time consensus property, and establish this property for several sequences of graphs, including the one-peer exponential graphs and the p-peer hyper-cuboids.

3.1 Definition of Finite-Time Consensus Property

We formally present the conditions for a sequence of graphs (or topologies) to have the finite-time consensus (or exact averaging) property.

Definition 3.1 (Graph sequence with Finite-Time Consensus)**.** *The sequence of graphs $\mathcal{G}^{(l)} = (\mathcal{V}, W^{(l)}, \mathcal{E}^{(l)})$, $l = 0, \ldots, \tau - 1$, has the finite-time consensus property with parameter $\tau \in \mathbb{N}_{\geq 1}$ if and only if the weight matrices $\{W^{(l)}\}_{l=0}^{\tau-1}$ are doubly stochastic and satisfy*

$$W^{(\tau-1)} W^{(\tau-2)} \cdots W^{(1)} W^{(0)} = \frac{1}{n} \mathbb{1}\mathbb{1}^\intercal. \tag{3}$$

For a sequence of doubly stochastic matrices $\{W^{(l)}\}_{l=0}^{\tau-1}$, the condition (3) is equivalent to

$$\big(W^{(\tau-1)} - \tfrac{1}{n}\mathbb{1}\mathbb{1}^\intercal\big)\big(W^{(\tau-2)} - \tfrac{1}{n}\mathbb{1}\mathbb{1}^\intercal\big) \cdots \big(W^{(1)} - \tfrac{1}{n}\mathbb{1}\mathbb{1}^\intercal\big)\big(W^{(0)} - \tfrac{1}{n}\mathbb{1}\mathbb{1}^\intercal\big) = 0. \tag{4}$$

The parameter τ in Definition 3.1 is not arbitrary and might depend on the matrix size n and graph structure. Examples of graphs with finite-time consensus are provided in Sections 3.2–3.4, and the parameter τ will be clear from the context. In Definition 3.1, each matrix $W^{(l)}$ in the sequence is required to be doubly stochastic but needs not be symmetric or connected. The potential disconnectivity might be a desirable property in the context of decentralized optimization, because such graphs tend to be sparser and using them in decentralized

Table 2: Classes of graph sequences that satisfy Definition 3.1.

Topology	Orientation	Size n	Maximum Degree	# of Iterations (τ) for Finite-Time Consensus
One-Peer Exponential (Ying et al., 2021)	Directed	Power of 2 [a]	1	$\log_2(n)$
One-Peer Hyper-Cube (Shi et al., 2016)	Undirected	Power of 2	1	$\log_2(n)$
p-Peer Hyper-Cuboid	Undirected	Any $n \in \mathbb{N}$	Largest Prime Factor p [b]	# of Prime Factors
de Bruijn (Delvenne et al., 2009)	Directed	Power of $p \in \mathbb{N}_{\geq 2}$	p	$\log_p(n)$

[a] When $n \neq 2^\tau$, one-peer exponential graphs are still well-defined, but the finite-time consensus property no longer holds.
[b] Consider $n = 20$ and its prime factors $\{2, 2, 5\}$. The largest prime factor is $p = 5$ and the number of prime factors is $\tau = 3$.

algorithms helps reduce the communication overhead at each iteration. Nevertheless, when considered jointly, a sequence of graphs satisfying Definition 3.1 exhibits the same connectivity properties as the fully connected graph.

Graph sequences with finite-time consensus have been proposed and analyzed in various contexts. For example, Delvenne et al. (2009) establish the finite-time property for a sequence of static de Bruijn graphs. (In Delvenne et al. (2009), the term “deadbeat” consensus is used instead of finite-time consensus.) Later, Shi et al. (2016) justify the finite-time consensus property for a sequence of one-peer hyper-cubes (when $n = 2^\tau$ for some $\tau \in \mathbb{N}_{\geq 1}$). (As the name suggests, the hyper-cuboids discussed in Section 3.4 reduce to hyper-cubes when the matrix size n is a power of 2.) Assran et al. (2019) observe in numerical experiments that one-peer exponential graphs have the finite-time consensus property when n is a power of 2, which is later theoretically justified in Ying et al. (2021). More recently, Takezawa et al. (2023) claim to build graph sequences of any node size $n \in \mathbb{N}$ that satisfy (3), but they do not provide any theoretical justification. In another line of research, Ding et al. (2023) revisit the optimal message passing algorithm developed in Bar-Noy et al. (1993) and proposes a communication-optimal exact consensus algorithm. The decentralized algorithm proposed in Ding et al. (2023) needs an additional copy of the optimization variable at each agent, and “finite-time consensus” is achieved with the help of these auxiliary variables. The discussion of this approach is out of the scope of this paper, and further investigation is left as future work. Table 2 presents several of the existing graph sequences for which Definition 3.1 is proven to hold. In the remainder of this section, we detail the one-peer exponential graphs and the p-peer hyper-cuboids. For the well-studied de Bruijn graphs, we establish its connection to the p-peer hyper-cuboids and postpone the details to Appendix A.

3.2 One-Peer Exponential Graphs

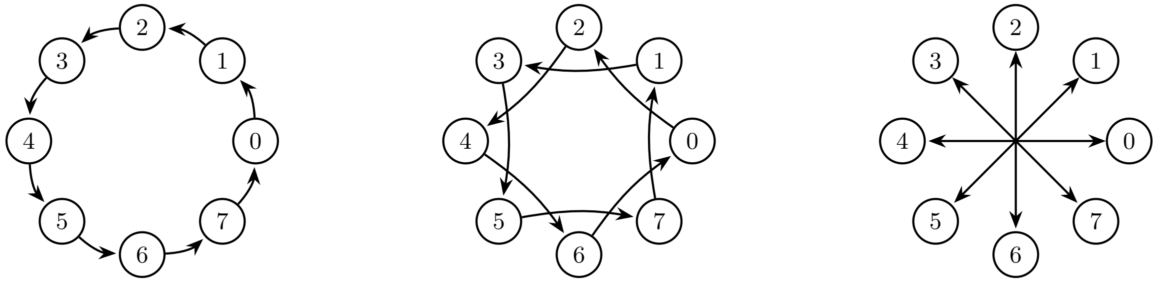
We present the weight matrices of one-peer exponential graphs (Ying et al., 2021), listing several properties we will leverage throughout this paper. In particular, the weight matrices representing one-peer exponential graphs are asymmetric, doubly stochastic, sparse (potentially disconnected), and when n is a power of 2, satisfy (3).

As a byproduct, we develop an alternative proof for the finite-time consensus property of one-peer exponential graphs (when $n = 2^\tau$). In comparison, Ying et al. (2021) provided the first theoretical justification, which we believe was inspired by the use of binary numbers in the proof for hyper-cubes (Shi et al., 2016).

For a given matrix size $n \in \mathbb{N}_{\geq 2}$, let $\tau = \lceil \log_2(n) \rceil$. Then, the weight matrices representing the one-peer exponential graphs $\{\mathcal{G}^{(l)}\}_{l \in \mathbb{N}}$ are defined as

$$w_{ij}^{(l)} = \begin{cases} \frac{1}{2} & \text{if } \operatorname{mod}(j-i, n) = 2^{\operatorname{mod}(l,\tau)} \\ \frac{1}{2} & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

and Figure 1 shows the three one-peer exponential graphs when $n = 8$ and $\tau = 3$. By definition, given $n \in \mathbb{N}_{\geq 2}$, there exists a number of $\tau = \lceil \log_2(n) \rceil$ distinct one-peer exponential graphs. All of them are asymmetric, doubly stochastic, circulant, and some of them are *not* strongly connected. When one-peer



(a) $\mathcal{G}^{(0)}$.

(b) $\mathcal{G}^{(1)}$.

(c) $\mathcal{G}^{(2)}$.

Figure 1: The three one-peer exponential graphs $\{\mathcal{G}^{(l)}\}_{l=0}^{2}$ with $n = 8$ and $\tau = \log_2(8) = 3$. Note that all nodes have self-loops, although not explicitly shown in the figure.

exponential graphs are used as network topology in decentralized algorithms, each agent only communicates with one neighbor at each iteration. Thus, the total communication cost per iteration is $\Omega(1)$. Finally, the crucial property that makes one-peer exponential graphs useful in decentralized optimization is presented as follows.

Proposition 3.1. *Given $n \in \mathbb{N}_{\geq 2}$, let $\tau = \lceil \log_2(n) \rceil$, and let $\{W^{(l)}\}_{l \in \mathbb{N}} \subset \mathbb{R}^{n \times n}$ be the weight matrices defined in (5). Each matrix $W^{(l)}$ is circulant and doubly stochastic, i.e., $W^{(l)}\mathbb{1} = \mathbb{1}$ and $\mathbb{1}^{\mathsf{T}} W^{(l)} = \mathbb{1}^{\mathsf{T}}$. In addition, if n is a power of 2 (i.e., $n = 2^{\tau}$) and the index sequence $\{l_i\}_{i=0}^{\tau-1}$ satisfies $\{\mathrm{mod}(l_0, \tau), \ldots, \mathrm{mod}(l_{\tau-1}, \tau)\} = \{0, \ldots, \tau - 1\}$, then the matrices $\{W^{(l_i)}\}_{i=0}^{\tau-1}$ satisfy the finite-time consensus property in Definition 3.1.*

From Proposition 3.1, given a sequence of one-peer exponential graphs with $n = 2^{\tau}$, any permutation of this sequence will still satisfy the finite-time consensus property (3). The properties in Definition 3.1 are the minimum requirements needed for algorithm analysis in Section 5. Additional properties, such as the circulant property stated in Proposition 3.1, are desirable but unnecessary in algorithm design.

Our proof of Proposition 3.1 relies on some basic properties of *circulant matrices*, which are summarized in the following lemma and can be found in, *e.g.*, in Horn and Johnson (2013, Section 4.7.7).

Lemma 3.2. *The $n \times n$ circulant matrix associated with an n-vector $c = (c_0, c_1, \ldots, c_{n-1})$ is defined by*

$$C = \mathrm{Circ}(c_0, c_1, \ldots, c_{n-1}) \triangleq \begin{bmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & & c_2 \\ \vdots & c_1 & c_0 & \ddots & \ddots \\ c_{n-2} & & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \ddots & c_1 & c_0 \end{bmatrix}. \tag{6}$$

The eigenvalue decomposition of C (6) is given by

$$C = \left(\tfrac{1}{\sqrt{n}} \mathcal{F}\right) \cdot \left(\mathrm{diag}(\mathcal{F}c)\right) \cdot \left(\tfrac{1}{\sqrt{n}} \mathcal{F}^{\dagger}\right),$$

where $\mathcal{F}$ is the $n \times n$ DFT matrix, $\dagger$ denotes the Hermitian (conjugate transpose), and the diag *operator transforms an n-vector into an $n \times n$ diagonal matrix. Moreover, the eigenvalues of C (6) are given by*

$$\lambda_i = c_0 + c_1 \omega^{i} + c_2 \omega^{2i} + \cdots + c_{n-1} \omega^{(n-1)i}, \quad i = 0, 1, \ldots, n-1,$$

where $\omega = \exp\left(\frac{2\pi \hat{\jmath}}{n}\right)$ is a primitive n-th root of unity and $\hat{\jmath}$ is the imaginary number (i.e., $\hat{\jmath}^2 = -1$).

Now we present the proof for Proposition 3.1.

Proof of Proposition 3.1. Note that the mixing matrix of one-peer exponential graphs defined in (5) is a circulant matrix, and thus one has

$$W^{(l_{\tau-1})}\cdots W^{(l_1)}W^{(l_0)} = \big(\tfrac{1}{\sqrt{n}}\mathcal{F}\big)\cdot\big(\Lambda^{(l_{\tau-1})}\cdots\Lambda^{(l_1)}\Lambda^{(l_0)}\big)\cdot\big(\tfrac{1}{\sqrt{n}}\mathcal{F}^{\dagger}\big),$$

where $\Lambda^{(l_i)} = \operatorname{diag}(\mathcal{F}c^{(l_i)})$ and $c^{(l_i)}$ is the first column of $W^{(l_i)}$, for $i = 0, 1, \ldots, \tau-1$. For simplicity, we denote $\Lambda \triangleq \Lambda^{(l_{\tau-1})}\cdots\Lambda^{(l_1)}\Lambda^{(l_0)}$.

For each $i = 0, 1, \ldots, \tau-1$, the first element in $\mathcal{F}c^{(l_i)}$ is 1 because the first row of $\mathcal{F}$ is an all-one vector. This implies that the first element in Λ is also 1. Similarly, the $(j+1)$-th diagonal element in Λ can be found by expanding the definition $\Lambda^{(l_i)} = \operatorname{diag}(\mathcal{F}c^{(l_i)})$:

$$\begin{aligned}
&\frac{1}{2^{\tau}}\Big[(1+\omega^{(n-1)(i)})(1+\omega^{(n-2)(i)})(1+\omega^{(n-4)(i)})\cdots(1+\omega^{(n-2^{\tau-1})(i)})\Big]\\
&=\frac{1}{2^{\tau}}\Big[(1+\omega^{(-1)(i)})(1+\omega^{(-2)(i)})(1+\omega^{(-4)(i)})\cdots(1+\omega^{(-2^{\tau-1})(i)})\Big]\\
&=\frac{1}{2^{\tau}}\sum_{l=0}^{n-1}\omega^{-il} = \frac{1}{2^{\tau}}\left(\frac{1-\omega^{-in}}{1-\omega^{-i}}\right) = 0,
\end{aligned}$$

where in the first equation we use the assumption that $n = 2^{\tau}$ for some $\tau \in \mathbb{N}_{\geq 1}$. Hence, we have

$$\begin{aligned}
W^{(l_{\tau-1})}\cdots W^{(l_1)}W^{(l_0)} &= \big(\tfrac{1}{\sqrt{n}}\mathcal{F}\big)\Lambda^{(l_{\tau-1})}\cdots\Lambda^{(l_1)}\Lambda^{(l_0)}\big(\tfrac{1}{\sqrt{n}}\mathcal{F}^{\dagger}\big)\\
&= \big(\tfrac{1}{\sqrt{n}}\mathcal{F}\big)\big(\operatorname{diag}(1,0,0,\ldots,0)\big)\big(\tfrac{1}{\sqrt{n}}\mathcal{F}^{\dagger}\big)\\
&= \tfrac{1}{n}\mathbb{1}\mathbb{1}^{\mathsf{T}},
\end{aligned}$$

which establishes the desirable identity (3). To show (4), we consider the product of the following two terms

$$\big(W^{(l_{i+1})} - \tfrac{1}{n}\mathbb{1}\mathbb{1}^{\mathsf{T}}\big)\big(W^{(l_i)} - \tfrac{1}{n}\mathbb{1}\mathbb{1}^{\mathsf{T}}\big) = W^{(l_{i+1})}W^{(l_i)} - \tfrac{1}{n}\mathbb{1}\mathbb{1}^{\mathsf{T}}, \quad \text{for any } i = 0, \ldots, \tau-2.$$

This equality holds due to the doubly stochastic property of $W^{(l_i)}$. We can then repeat this process to obtain

$$\big(W^{(l_{\tau-1})} - \tfrac{1}{n}\mathbb{1}\mathbb{1}^{\mathsf{T}}\big)\cdots\big(W^{(l_1)} - \tfrac{1}{n}\mathbb{1}\mathbb{1}^{\mathsf{T}}\big)\big(W^{(l_0)} - \tfrac{1}{n}\mathbb{1}\mathbb{1}^{\mathsf{T}}\big) = W^{(l_{\tau-1})}\cdots W^{(l_1)}W^{(l_0)} - \tfrac{1}{n}\mathbb{1}\mathbb{1}^{\mathsf{T}} = 0,$$

where in the last step we use (3). □

Despite the desirable finite-time consensus property, incorporating one-peer exponential graphs into decentralized optimization algorithms is not straightforward. To see this, consider a simple example where $n = 8$ (and $\tau = \log_2(8) = 3$). Figure 1 shows the three one-peer exponential graphs with $n = 8$. None of these three graphs has symmetric weight matrices; the last two are not connected. As a result, the weight matrices of the last two graphs in Figure 1 have $\rho \triangleq \|W - \frac{1}{n}\mathbb{1}\mathbb{1}^{\mathsf{T}}\|_2 = 1$ while current analyses of decentralized algorithms often assume $\rho < 1$. Therefore, new analysis techniques are needed to exploit the properties of one-peer exponential graphs in decentralized algorithms (especially exact or bias-corrected methods), and a similar discussion also holds for the graph sequences studied in Sections 3.3–3.4.

3.3 One-Peer Hyper-Cube

Another example of graph sequences that satisfy Definition 3.1 is one-peer hyper-cubes (Shi et al., 2016) (with n a power of 2). Hyper-cubes have been extensively studied in theoretical computer science (see, *e.g.*, Harary et al. (1988) for a survey). Still, the specialization to *one-peer* hyper-cubes with finite-time consensus was first, to the best of our knowledge, discussed in Shi et al. (2016). The formal definition of one-peer hyper-cubes is presented here to keep our work self-contained and, more importantly, to motivate the extension to p-peer hyper-cuboids for any $n \in \mathbb{N}_{\geq 2}$ in Section 3.4.

Given an integer $\tau \in \mathbb{N}_{\geq 1}$ and $n \triangleq 2^{\tau}$, the weight matrices $\{W^{(l)}\}_{l\in\mathbb{N}} \subset \mathbb{R}^{n\times n}$ representing the one-peer hyper-cube $\{\mathcal{G}^{(l)}\}_{l\in\mathbb{N}}$ are defined by

$$w_{ij}^{(l)} = \begin{cases} \frac{1}{2} & \text{if } (i \wedge j) = 2^{\mathrm{mod}(l,\tau)}, \\ \frac{1}{2} & \text{if } i = j, \\ 0 & \text{otherwise}, \end{cases} \tag{7}$$

where the notation $i \wedge j$ represents the bit-wise XOR operation between integers i and j. The difference in the definition of one-peer hyper-cubes (7) and that of one-peer exponential graphs (5) is minor yet critical: The operation $(i \wedge j)$ is used in (7) while $\mathrm{mod}(j-i, n)$ in (5). Since $(i \wedge j) = (j \wedge i)$, it immediately follows that the mixing matrices of one-peer hyper-cubes are symmetric. To see the connection between one-peer hyper-cubes (7) and the *static* hyper-cubes, we represent the integer i in its binary form $(i_{\tau-1}i_{\tau-2}\cdots i_0)_2$. Then, the first if-condition in (7) can be rewritten as

$$(i_{\tau-1}i_{\tau-2}\cdots i_0)_2 \wedge (j_{\tau-1}j_{\tau-2}\cdots j_0)_2 = (0\cdots 0\,1\,\underbrace{0\cdots 0}_{\mathrm{mod}(l,\tau)})_2;$$

that is, only the $(\mathrm{mod}(l,\tau)+1)$-th digit in i's and j's binary representation is different, and the rest of the digits are the same. To construct one-peer hyper-cubes, we first index the vertices as τ-digit binary numbers, and then an edge is created between two distinct vertices if their binary representations differ by a single digit. Finally, the finite-time consensus property proof for one-peer hyper-cubes is postponed to Section 3.4, since one-peer hyper-cubes will be covered as a special case of the p-peer hyper-cuboids.

3.4 p-Peer Hyper-Cuboids

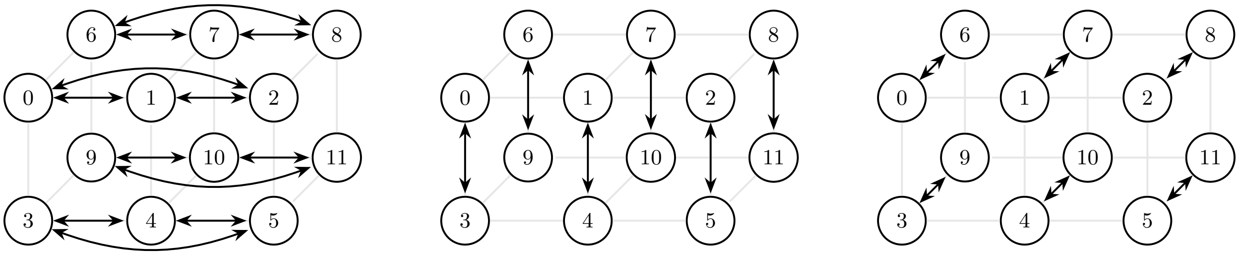
Both one-peer hyper-cubes and one-peer exponential graphs enjoy the finite-time consensus property when the number of agents is a power of 2. One natural extension of hyper-cubes to admit an arbitrary number of agents is the *hyper-cuboid*, which has many different names (*e.g.*, hyper-box, orthotope) and has been well studied in, *e.g.*, Coxeter (1973). So, borrowing the idea behind the one-peer hyper-cubes, we present a family of sparse graphs that achieves finite-time consensus for any integer $n \in \mathbb{N}_{\geq 2}$. Recently, Takezawa et al. (2023) present a constructive approach to build the so-called *k-peer hyper-hypercubes*. Yet, the authors do not present an explicit matrix representation for the proposed graphs, nor do they prove the finite-time consensus property.

Recall that one-peer hyper-cubes (7) are defined via the binary representation of integers. Then, the extension to arbitrary matrix size n relies on a *multi-base representation* of integers (Krenn et al., 2015). To be specific, the $(p_{\tau-1}, p_{\tau-2}, \ldots, p_0)$-based representation of an integer is an element in the group $\mathbb{N}_{p_{\tau-1}} \times \mathbb{N}_{p_{\tau-2}} \times \cdots \times \mathbb{N}_{p_0}$, where $\mathbb{N}_{p_k}$ is the group of nonnegative integers modulo $p_k \in \mathbb{N}_{\geq 2}$. Any natural integer smaller than $p_{\tau-1} \times p_{\tau-2} \times \cdots \times p_0$ finds a one-to-one mapping in this group. For example, a $(2, 2, \ldots, 2)$-based representation is equivalent to the binary representation of an integer. An informative example is the $(2, 3)$-based representation. In this case, we can map the integer in $\{0, 1, \ldots, 5\}$ according to the following rule:

$$\begin{aligned} &0 \to \{0\}_2 \times \{0\}_3, \quad 1 \to \{0\}_2 \times \{1\}_3, \quad 2 \to \{0\}_2 \times \{2\}_3, \\ &3 \to \{1\}_2 \times \{0\}_3, \quad 4 \to \{1\}_2 \times \{1\}_3, \quad 5 \to \{1\}_2 \times \{2\}_3. \end{aligned}$$

To shorten the notation, we overload binary representation and denote the $(p_{\tau-1}, p_{\tau-2}, \ldots, p_0)$-based representation of $i \in \mathbb{N}$ as $(i_{p_{\tau-1}} \cdots i_{p_1} i_{p_0})_{p_{\tau-1},\ldots,p_1,p_0}$, so that we can also re-write $\{a\}_{p_1} \times \{b\}_{p_0}$ as $(a, b)_{p_1,p_0}$.

Now, we are ready to construct p-peer hyper-cuboids with n agents. Suppose the prime factorization of n is given by $n = p_{\tau-1} \cdots p_1 p_0$, where all the p_j are prime numbers. (It is possible that $p_i = p_j$ for $i \neq j$, and the order of $\{p_j\}$ does not matter.) For example, the prime factor set of $n = 12$ is $(p_2, p_1, p_0) = (2, 2, 3)$.



(a) $\mathcal{G}^{(0)}$.

(b) $\mathcal{G}^{(1)}$.

(c) $\mathcal{G}^{(2)}$.

Figure 2: The three 2-peer hyper-cuboids $\{\mathcal{G}^{(l)}\}_{l=0}^{2}$ with $n = 12$, $(p_2, p_1, p_0) = (2, 2, 3)$, and $\tau = 3$. Note that all nodes have self-loops, although not explicitly shown in the figure.

Then, the weight matrices of p-peer hyper-cuboids, with $p = \max\{p_0, \ldots, p_{\tau-1}\} - 1$, are defined by

$$w_{ij}^{(l)} = \begin{cases} \frac{1}{p_{\mathrm{mod}(l,\tau)}} & \text{if } (i \wedge_{p_{\tau-1},\ldots,p_1,p_0} j) = (0, \cdots, 0, 1, \underbrace{0, \cdots, 0}_{\mathrm{mod}(l,\tau)})_{p_{\tau-1},\ldots,p_1,p_0} \\ \frac{1}{p_{\mathrm{mod}(l,\tau)}} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

where $i \wedge_{p_{\tau-1},\ldots,p_1,p_0} j$ denotes the bit-wise XOR operation between the $(p_{\tau-1}, \ldots, p_1, p_0)$-based representations of i and j; that is, if the $p_k \in \mathbb{N}_{p_k}$ element of i's multi-base representations is the same as that of j, then return $\{0\}_{p_k}$, and otherwise return $\{1\}_{p_k}$. Figure 2 shows all three distinct 2-peer hyper-cuboids with 12 agents. In this example, $n = 12$, $(p_2, p_1, p_0) = (2, 2, 3)$, and $\tau = 3$. To illustrate the definition (7), take the edge $(i, j) = (8, 11)$ in $\mathcal{G}^{(1)}$ as an example; see Figure 2b. The two integers $i = 8$ and $j = 11$ are mapped in the $(2, 2, 3)$-based representation as

$$8 \to \{1\}_2 \times \{0\}_2 \times \{2\}_3, \qquad 11 \to \{1\}_2 \times \{1\}_2 \times \{2\}_3.$$

These two representations differ only at the second sub-group $\mathbb{N}_{p_1} = \mathbb{N}_2$, and thus when $l = 1$, agents $i = 8$ and $j = 11$ are connected with weight $w_{ij}^{(l)} = w_{8,11}^{(1)} = \frac{1}{p_1} = \frac{1}{2}$.

The definition of p-peer hyper-cuboids (8) is clear as an extension from binary numbers to multi-base integer representations. Yet, the original definition (8) is less intuitive when we try to establish the properties of p-peer hyper-cuboids. It turns out that the weight matrix $W^{(l)}$ of p-peer hyper-cuboids defined in (7) also has an elegant representation in terms of Kronecker products:

$$W^{(l)} = W^{(l)}(p_{\tau-1}) \otimes \ldots \otimes W^{(l)}(p_1) \otimes W^{(l)}(p_0), \tag{9}$$

where each $p_r \times p_r$ matrix $W^{(l)}(p_r)$ is defined by

$$W^{(l)}(p_r) = \begin{cases} I_{p_r} & \text{if } \mathrm{mod}(l, \tau) \neq r \\ \frac{1}{p_r} \mathbb{1}\mathbb{1}^{\mathsf{T}} & \text{if } \mathrm{mod}(l, \tau) = r. \end{cases} \tag{10}$$

The equivalence between (8) and (9) can be established as follows.

$$\begin{aligned} W^{(l)} &= \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}^{(l)} e_i e_j^{\mathsf{T}} \\ &= \sum_{i_{p_{\tau-1}}} \cdots \sum_{i_{p_0}} \sum_{j_{p_{\tau-1}}} \cdots \sum_{j_{p_0}} w_{ij}^{(l)} (\hat{e}_{i_{p_{\tau-1}}} \otimes \cdots \otimes \hat{e}_{i_{p_0}})(\hat{e}_{j_{p_{\tau-1}}} \otimes \cdots \otimes \hat{e}_{j_{p_0}})^{\mathsf{T}} \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i_{p_{\tau-1}}} \cdots \sum_{i_{p_0}} \sum_{j_{p_r}:\, r=\operatorname{mod}(l,\tau)} w_{ij}^{(l)} (\hat{e}_{i_{p_{\tau-1}}} \hat{e}_{i_{p_{\tau-1}}}^{\mathsf{T}}) \otimes \cdots \otimes (\hat{e}_{i_{p_r}} \hat{e}_{j_{p_r}}^{\mathsf{T}}) \otimes \cdots \otimes (\hat{e}_{i_{p_0}} \hat{e}_{i_{p_0}}^{\mathsf{T}}) && \text{(11a)}\\
&= \Big(\sum_{i_{p_{\tau-1}}} \hat{e}_{i_{p_{\tau-1}}} \hat{e}_{i_{p_{\tau-1}}}^{\mathsf{T}} \Big) \otimes \cdots \otimes \Big(\sum_{i_{p_r}} \sum_{j_{p_r}} w_{ij}^{(l)} \hat{e}_{i_{p_r}} \hat{e}_{j_{p_r}}^{\mathsf{T}} \Big) \otimes \cdots \otimes \Big(\sum_{i_{p_0}} \hat{e}_{i_{p_0}} \hat{e}_{i_{p_0}}^{\mathsf{T}} \Big) && \text{(11b)}\\
&= W^{(l)}(p_{\tau-1}) \otimes \ldots \otimes W^{(l)}(p_1) \otimes W^{(l)}(p_0). && \text{(11c)}
\end{aligned}
$$

Here, the notation e_i is a base unit vector of length n, *i.e.*, all entries are 0 except that the i-th entry is 1. The notation $\hat{e}_{i_{p_r}}$ is a base unit vector of length p_r, where all entries are 0 except for the $(i_{p_r}+1)$-th entry, and recall p_r is a prime factor of n. Step (11a) uses the fact that the edge (i,j) exists if the $(p_{\tau-1},\ldots,p_1,p_0)$-based representation of i and j differ only at $r = \operatorname{mod}(l,\tau)$. Hence, we can cancel the pair of double summations between i_{p_k} and j_{p_k} except for j_{p_r}. Furthermore, we apply the transpose and mixed-product properties of the Kronecker product. Step (11b) distributes each summation into the corresponding Kronecker product, and (11c) uses the definition of $W^{(l)}(p_r)$ in (10).

Now, we are ready to establish the finite-time consensus property of p-peer hyper-cuboids.

Proposition 3.3. *Given $n \in \mathbb{N}_{\geq 2}$, let $\{W^{(l)}\}_{l\in\mathbb{N}} \subset \mathbb{R}^{n\times n}$ be the weight matrices defined in (8). Each matrix $W^{(l)}$ is symmetric and doubly stochastic. For an index sequence $\{l_i\}_{i=0}^{\tau-1}$ with $\{\operatorname{mod}(l_0,\tau),\ldots,\operatorname{mod}(l_{\tau-1},\tau)\} = \{0,\ldots,\tau-1\}$, the matrices $\{W^{(l_i)}\}_{i=0}^{\tau-1}$ satisfy the finite-time consensus property in Definition 3.1.*

Proof. First, the symmetry of $W^{(l)}$ follows directly from its definition. To see the row stochastic property, we have

$$
\begin{aligned}
W^{(l)}\mathbb{1} &= \Big(W^{(l)}(p_{\tau-1}) \otimes W^{(l)}(p_{\tau-2}) \otimes \cdots \otimes W^{(l)}(p_0) \Big) \left(\mathbb{1}_{p_{\tau-1}} \otimes \cdots \otimes \mathbb{1}_{p_0} \right)\\
&= \Big(W^{(l)}(p_{\tau-1}) \mathbb{1}_{p_{\tau-1}} \Big) \otimes \cdots \otimes \Big(W^{(l)}(p_0) \mathbb{1}_{p_0} \Big)\\
&= \mathbb{1},
\end{aligned}
$$

where we use the mixed-product property of the Kronecker product. The column stochastic property follows from symmetry and row stochasticity. Hence, $W^{(l)}$ is doubly stochastic. Next, we show that the sequence $\{W^{(l)}\}_{l=0}^{\tau-1}$ has the finite-time consensus property. Utilizing the Kronecker product property, we establish that

$$
\begin{aligned}
\prod_{l=0}^{\tau-1} W^{(l)} &= \prod_{l=0}^{\tau-1} \Big(W^{(l)}(p_{\tau-1}) \otimes W^{(l)}(p_{\tau-2}) \otimes \cdots \otimes W^{(l)}(p_0) \Big)\\
&= \Big(\prod_{l=0}^{\tau-1} W^{(l)}(p_{\tau-1}) \Big) \otimes \Big(\prod_{l=0}^{\tau-1} W^{(l)}(p_{\tau-2}) \Big) \otimes \cdots \otimes \Big(\prod_{l=0}^{\tau-1} W^{(l)}(p_0) \Big)\\
&= \Big(\tfrac{1}{p_{\tau-1}} \mathbb{1}_{p_{\tau-1}} \mathbb{1}_{p_{\tau-1}}^{\mathsf{T}} \Big) \otimes \cdots \otimes \Big(\tfrac{1}{p_0} \mathbb{1}_{p_0} \mathbb{1}_{p_0}^{\mathsf{T}} \Big)\\
&= \tfrac{1}{n} \mathbb{1}\mathbb{1}^{\mathsf{T}},
\end{aligned}
$$

where we again use the mixed-product property. Combining this result with the fact that p-peer hyper-cuboids are periodic, and all instances of p-peer hyper-cuboids are commutative, we obtain that any permutation of any sequence of $W^{(l)}$ of length larger than τ has the finite-time consensus property as well. □

Proposition 3.4. *For any $n = 2^\tau$ with $\tau \in \mathbb{N}_{\geq 1}$, the weight matrices of one-peer hyper-cubes defined in (7) are symmetric, double-stochastic, and have the finite-time consensus property (3).*

Proof. When the number of agents is $n = 2^\tau$, the one-peer hyper-cube is just a special case of p-peer hyper-cuboids (with $p = 1$). Decomposition of n into its $(2,2,\ldots,2)$-base representation (or equivalently, binary representation) yields the desired properties. □

Connection with de Bruijn graphs. The de Bruijn graph was first studied in de Bruijn (1946) and its finite-time consensus property has been studied in Delvenne et al. (2009). Here, we present a formulation of de Bruijn graphs closely related to p-peer hyper-cuboids (8). With this definition of de Bruijn graphs, we show that de Bruijn graphs are just permutations of p-peer hyper-cuboids.

Let $n = p^\tau$ with $(p, \tau) \in \mathbb{N}_{\geq 2} \times \mathbb{N}_{\geq 1}$, and define the p-based representation of $i \in \{0, 1, \ldots, n-1\}$ as $(i_{\tau-1} i_{\tau-2} \ldots i_0)_p$. Then, the $n \times n$ weight matrix of the de Bruijn graph is defined by

$$w_{ij} = \begin{cases} \frac{1}{p} & \text{if } (i_{\tau-2} i_{\tau-3} \ldots i_0)_p = (j_{\tau-1} j_{\tau-2} \ldots j_1)_p \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

The following proposition presents the connection between de Bruijn graphs and p-peer hyper-cuboids when the matrix size n is a power of $p \in \mathbb{N}_{\geq 2}$.

Proposition 3.5. *Given $n = p^\tau$ with $(p, \tau) \in \mathbb{N}_{\geq 2} \times \mathbb{N}_{\geq 1}$, let $W_{\mathrm{db}} \in \mathbb{R}^{n \times n}$ be the weight matrix of the de Bruijn matrix defined in (12) and $\{W_{\mathrm{hc}}^{(l)}\}_{l \in \mathbb{N}}$ be the weight matrices of the p-peer hyper-cuboids defined in (8). Then for any $l \in \mathbb{N}$, there exist $n \times n$ permutation matrices $P^{(l)}$ and $Q^{(l)}$ such that*

$$W_{\mathrm{hc}}^{(l)} = P^{(l)} W_{\mathrm{db}} (Q^{(l)})^\top.$$

Proof. See Appendix A. □

4 Algorithm Description

This section presents Gradient Tracking with time-varying topologies (TV-GT) and our modified version, Gradient Tracking with Finite-Time Consensus Topologies (GT-FT). Henceforth, we will refer to each algorithm as TV-GT and GT-FT, respectively.

4.1 Gradient Tracking with Time-Varying Topologies

Gradient Tracking (GT) (Di Lorenzo and Scutari, 2016; Nedić et al., 2017) is a well-studied decentralized algorithm for solving Problem (1), and various formulations of TV-GT exist in the literature. The presented form follows from Di Lorenzo and Scutari (2016) (called Semi-ATC-TV-GT), and its implementation involves a sequence of graphs $\mathcal{G}^{(k)} = (\mathcal{V}, W^{(k)}, \mathcal{E}^{(k)})$ which models the connections between the group of n agents. Here, $\mathcal{V}$ is the set of n nodes and $\mathcal{E}^{(k)} \subseteq \{(i,j) \mid (i,j) \in \mathcal{V} \times \mathcal{V}\}$ describes the set of connections between agents. The set of agents $\mathcal{V}$ remains static while the set of edges $\mathcal{E}^{(k)}$ can be time-varying. The entry $w_{ij}^{(k)}$ in the matrix $W^{(k)}$ applies a weighting factor to the parameters sent from agent i to agent j. If $w_{ij}^{(k)} = 0$, that means agent i is not a neighbor of agent j in $\mathcal{G}^{(k)}$; *i.e.*, $(i,j) \notin \mathcal{E}^{(k)}$.

Given an initial point $\{x_i^{(0)}\} \subset \mathbb{R}^d$ and stepsizes $(\alpha, \eta) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0}$, set $g_i^{(0)} = \eta \nabla F_i(x_i^{(0)}, \xi_i^{(0)})$ for $i = 1, \ldots, n$. Then, TV-GT takes the following iterations for $k = 0, 1, 2, \ldots$

$$\begin{aligned} &\text{Parameter:} && x_i^{(k+1)} = \sum_{j:\,(j,i)\in\mathcal{E}^{(k)}} w_{ji}^{(k)} (x_j^{(k)} - \alpha g_j^{(k)}) \\ &\text{Tracking Variable:} && g_i^{(k+1)} = \sum_{j:\,(j,i)\in\mathcal{E}^{(k)}} w_{ji}^{(k)} g_j^{(k)} + \eta \nabla F_i(x_i^{(k+1)}; \xi_i^{(k+1)}) - \eta \nabla F_i(x_i^{(k)}; \xi_i^{(k)}). \end{aligned} \tag{13}$$

Very often, the TV-GT iterations are written in a more compact form, which relies on the following augmented quantities

$$\mathbf{x}^{(k)} \triangleq \mathrm{col}\{x_1^{(k)}, \ldots, x_n^{(k)}\} \in \mathbb{R}^{dn},$$
$$\mathbf{g}^{(k)} \triangleq \mathrm{col}\{g_1^{(k)}, \ldots, g_n^{(k)}\} \in \mathbb{R}^{dn},$$

$$
\begin{aligned}
\mathbf{f}(\mathbf{x}) &\triangleq \frac{1}{n}\sum_{i=1}^{n} f_i(x_i),\\
\nabla\mathbf{f}(\mathbf{x}) &\triangleq \mathrm{col}\{\nabla f_1(x_1),\ldots,\nabla f_n(x_n)\} \in \mathbb{R}^{dn},\\
\nabla\mathbf{F}(\mathbf{x};\boldsymbol{\xi}) &\triangleq \mathrm{col}\{\nabla F_1(x_1;\xi_1),\ldots,\nabla F_n(x_n;\xi_n)\} \in \mathbb{R}^{dn},\\
\mathbf{W}^{(k)} &\triangleq W^{(k)} \otimes I_d \in \mathbb{R}^{dn\times dn}.
\end{aligned}
$$

With the augmented quantities, TV-GT (13) can be written compactly in the network form:

$$
\begin{aligned}
\mathbf{x}^{(k+1)} &= \mathbf{W}^{(k)}(\mathbf{x}^{(k)} - \alpha\mathbf{g}^{(k)})\\
\mathbf{g}^{(k+1)} &= \mathbf{W}^{(k)}\mathbf{g}^{(k)} + \eta\nabla\mathbf{F}(\mathbf{x}^{(k+1)};\boldsymbol{\xi}^{(k+1)}) - \eta\nabla\mathbf{F}(\mathbf{x}^{(k)};\boldsymbol{\xi}^{(k)}).
\end{aligned}
$$

The choice of the matrix sequence $\{W^{(k)}\}$ is critical yet challenging. One focus of this work is to analyze TV-GT in which the network topologies are restricted to topology sequences that satisfy Definition 3.1, and in particular, we focus on the nonconvex, stochastic setting. We have detailed the finite-time consensus property in Section 3 and motivated its usefulness in decentralized optimization algorithms. However, the existing analysis for TV-GT cannot handle the graphs described in Section 3. For example, work by Alghunaim and Yuan (2022; 2023); Koloskova et al. (2021) provide tight bounds for the convergence rate of Gradient Tracking with static topology ($W^{(k)} = W$ for all $k \in \mathbb{N}$). Yet, their work cannot be readily extended to handle time-varying topologies as they all assume the symmetry of the mixing matrix. Even if every mixing matrix in the topology sequence is symmetric, the product of these matrices does not necessarily remain symmetric. Moreover, prior work focusing exclusively on TV-GT either fails to consider the setting in which agents use stochastic gradients or assumes the connectivity of the topology at every iteration. See Table 1 and Section 2 for an in-depth discussion.

4.2 Gradient Tracking with Finite-Time Consensus Topologies

Algorithm 1 (GT-FT) presents our modified version of the TV-GT algorithm with sequences of topologies that satisfy Definition 3.1. In step (15b), the $x^{(k+1)}$ update involves a gradient descent step with the tracking variable $g^{(k)}$ as the update direction, and also a communication step with its neighbors to obtain a weighted average. In step (15c), the tracking variable update involves a weighted averaging step, subtracting the old local gradient, and adding the newly calculated local gradient. The update rules in GT-FT are known in GT literature. However, we are the first, to our knowledge, to propose and analyze this new scheme in which TV-GT is restricted to topology sequences that satisfy Definition 3.1. TV-GT, as presented in other works (*e.g.*, Nedić et al. (2017); Scutari and Sun (2019)), aims to be as general as possible when considering network topologies. In contrast, motivated by the useful finite-time consensus property (Definition 3.1) and the existence of such sparse graphs (see Section 3.2–3.4), we aim to specialize the analysis of GT to topology sequences that satisfy Definition 3.1, and to leverage the largely unexploited finite-time consensus property in decentralized optimization algorithms. To differentiate between the two approaches, we give our scheme the name GT-FT.

The analysis of GT-FT is presented in Section 5, and extensively uses the compact, networked form of Algorithm 1 (with the help of the augmented quantities):

$$
\begin{aligned}
\mathbf{W}^{(k)} &= W^{(\mathrm{mod}(k,\tau))} \otimes I_d && \text{(15a)}\\
\mathbf{x}^{(k+1)} &= \mathbf{W}^{(k)}(\mathbf{x}^{(k)} - \alpha\mathbf{g}^{(k)}) && \text{(15b)}\\
\mathbf{g}^{(k+1)} &= \mathbf{W}^{(k)}\mathbf{g}^{(k)} + \eta\nabla\mathbf{F}(\mathbf{x}^{(k+1)};\boldsymbol{\xi}^{(k+1)}) - \eta\nabla\mathbf{F}(\mathbf{x}^{(k)};\boldsymbol{\xi}^{(k)}). && \text{(15c)}
\end{aligned}
$$

5 Algorithm Analysis

This section presents the theoretical analysis of Algorithm 1. The assumptions needed for the analysis are listed in Section 5.1. In particular, we do not assume convexity and only have access to stochastic gradient

Algorithm 1 Gradient Tracking for Finite-Time Consensus Topologies (GT-FT)

```
Agent $i$ Input: $x_i^{(0)} \in \mathbb{R}^d$ and stepsizes $(\alpha, \eta) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0}$.
Global Input: The parameter $\tau \in \mathbb{N}_{\geq 1}$ for finite-time consensus, and the sequence of matrices $\{W^{(l)}\}$ that satisfies Definition 3.1.
Initialize $g_i^{(0)} = \eta \nabla F_i(x_i^{(0)}, \xi_i^{(0)}) \in \mathbb{R}^d$.
for $k = 0, 1, \dots$ do
    for $i = 1, \dots n$ (in parallel) do
        Deciding the combination coefficients:
            $w_{ij}^{(k)} = W^{(\mathrm{mod}(k,\tau))}[i, j], \quad \text{for all } j = 1, \dots, n.$   (14a)
        Parameter update:
            $x_i^{(k+1)} = \sum_{j\colon (j,i) \in \mathcal{E}^{(k)}} w_{ji}^{(k)} (x_j^{(k)} - \alpha g_j^{(k)}).$   (14b)
        Tracking variable update:
            $g_i^{(k+1)} = \sum_{j\colon (j,i) \in \mathcal{E}^{(k)}} w_{ji}^{(k)} g_j^{(k)} + \eta \nabla F_i(x_i^{(k+1)}; \xi_i^{(k+1)}) - \eta \nabla F_i(x_i^{(k)}; \xi_i^{(k)}).$   (14c)
    end for
end for
```

estimates of each local function f_i. Moreover, our analysis relies on a novel transformation of Algorithm 1 that decouples the updates of the parameters $\mathbf{x}^{(k)}$ and that of the tracking variables $\mathbf{g}^{(k)}$; see (16). This transformation is different from existing analyses of GT and is critical to leverage the finite-time consensus property of the mixing matrices. Finally, the convergence results are presented in Section 5.3, with detailed proofs postponed to Appendix B.

5.1 Assumptions and Transformation of Algorithm 1

In this subsection, we list all the assumptions needed for analysis. Our analysis does not need convexity of the objective function and holds for general nonconvex problems in the form of (1). In addition, we present the key transformation of Algorithm 1. This transformation is different from most analyses of GT methods in the literature (Alghunaim and Yuan, 2022; 2023; Koloskova et al., 2021; Song et al., 2022), and is critical in our analysis.

We make the following assumption on Problem (1).

Assumption 5.1. *Each function $f_i\colon \mathbb{R}^d \to \mathbb{R}$, $i = 1, \dots, n$ is continuously differentiable with an L-Lipschitz continuous gradient; i.e., there exists a constant $L \in \mathbb{R}_{>0}$ such that*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|, \quad \textit{for all } x, y \in \operatorname{int}\operatorname{dom} f_i, \textit{ and for all } i = 1, \dots, n.$$

In addition, the objective function $f\colon \mathbb{R}^d \to \mathbb{R}$ is bounded below, and the optimal value of Problem (1) is denoted by $f^* \in \mathbb{R}$.

At each iteration of Algorithm 1, a stochastic gradient estimator of each component function f_i is computed, based on the random variable $\xi_i^{(k)}$ in the probability space $(\Omega_i, \mathcal{F}_i, \mathbb{P}_i)$. Given initial conditions, let $\mathcal{F}^{(0)}$ denote the σ-algebra corresponding to the initial conditions and, for all $k \in \mathbb{N}_{\geq 1}$, let $\mathcal{F}^{(k)}$ denote the σ-algebra defined by the initial conditions and the random variables $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k)}\}$. Then, the following assumption is made on the stochastic gradient estimator.

Assumption 5.2. *For all $k \in \mathbb{N}$ and for all $i = 1, \dots, n$, the random variables $\xi_i^{(k)}$ are independent of each other. The stochastic gradient estimator satisfies*

$$\mathbb{E}[\nabla F_i(x_i^{(k)}; \xi_i^{(k)}) \mid \mathcal{F}^{(k)}] = \nabla f_i(x_i^{(k)}), \quad \textit{for all } i = 1, \dots, n.$$

In addition, there exists $\sigma \in \mathbb{R}_{>0}$ such that for all $k \in \mathbb{N}$ and for all $i = 1, \dots, n$, it holds that

$$\mathbb{E}[\|\nabla F_i(x_i^{(k)}; \xi_i^{(k)}) - \nabla f_i(x_i^{(k)})\|^2 \mid \mathcal{F}^{(k)}] \leq \sigma^2.$$

Besides Assumptions 5.1 and 5.2, our analysis of Algorithm 1 uses the change of variables $\mathbf{y}^{(k)} \triangleq \mathbf{g}^{(k)} - \eta\nabla\mathbf{F}(\mathbf{x}^{(k)}, \boldsymbol{\xi}^{(k)})$ and the following transformation of (15b)–(15c).

$$\mathbf{x}^{(k+1)} = \mathbf{W}^{(k)}\big(\mathbf{x}^{(k)} - \alpha\eta\nabla\mathbf{F}(\mathbf{x}^{(k)}; \boldsymbol{\xi}^{(k)}) - \alpha\mathbf{y}^{(k)}\big) \tag{16a}$$

$$\mathbf{y}^{(k+1)} = \mathbf{W}^{(k)}\mathbf{y}^{(k)} - \eta(\mathbf{I} - \mathbf{W}^{(k)})\nabla\mathbf{F}(\mathbf{x}^{(k)}, \boldsymbol{\xi}^{(k)}). \tag{16b}$$

This transformation, while just a simple change of variables, is critical to our analysis as it *decouples* the update of (16b) from (16a), *i.e.*, the update of $\mathbf{y}^{(k+1)}$ *does not depend on* $\mathbf{x}^{(k+1)}$. To the best of our knowledge, we are the first to establish this decoupled, Gauss–Seidel form for TV-GT. (Similar decoupling techniques have been used in Alghunaim and Yuan (2022; 2023); Alghunaim et al. (2021) for other decentralized algorithms, but have never been developed for TV-GT.) With this decoupled form, we can view the variables $(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k+1)})$ as one augmented quantity, and then develop new analysis techniques to incorporate topology sequences with finite-time consensus property. From an implementation point of view, the decoupled form (16) enables parallel computation of $\mathbf{x}^{(k+1)}$ and $\mathbf{y}^{(k+1)}$ while (15b)–(15c) must be performed in sequence.

Our convergence results describe the asymptotic behavior of the centroid recursion $\bar{x}^{(k)} \triangleq \frac{1}{n}\sum_{i=1}^{n} x_i^{(k)}$ and the consensus error $\hat{\mathbf{e}}^{(k)}$ which helps to model the difference between the centroid parameter $\bar{x}^{(k)}$ and each agent's parameter $x_i^{(k)}$. In view of this and upon further computation (detailed in Section B.1), the iterations (16) are further converted into

$$\bar{x}^{(k+1)} = \bar{x}^{(k)} - \alpha\eta(\overline{\nabla f}(\mathbf{x}^{(k)}) + \bar{s}^{(k)}) \tag{17a}$$

$$\hat{\mathbf{e}}^{(k+1)} = \mathbf{G}^{(k)}\hat{\mathbf{e}}^{(k)} - \eta(\mathbf{h}^{(k+1)} + \mathbf{w}^{(k)}), \tag{17b}$$

where $\bar{\mathbf{x}}^{(k)} \triangleq \mathbb{1}_n \otimes \bar{x}^{(k)}$, $\widehat{\mathbf{W}}^{(k)} \triangleq \mathbf{W}^{(k)} - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\mathsf{T} \otimes I_d$, $\widehat{\mathbf{I}} \triangleq \mathbf{I} - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\mathsf{T} \otimes I_d$, $\mathbf{z}^{(k)} \triangleq \mathbf{y}^{(k)} + \eta\nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)})$, $\bar{\mathbf{z}}^{(k)} \triangleq \mathbb{1}_n \otimes \bar{z}^{(k)}$, and

$$\begin{gathered}\mathbf{h}^{(k+1)} = \begin{bmatrix} \alpha\widehat{\mathbf{W}}^{(k)}(\nabla\mathbf{f}(\mathbf{x}^{(k)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)})) \\ (\mathbf{I} - \mathbf{W}^{(k)})(\nabla\mathbf{f}(\mathbf{x}^{(k)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)})) - \widehat{\mathbf{I}}(\nabla\mathbf{f}(\bar{\mathbf{x}}^{(k+1)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)})) \end{bmatrix} \\ \hat{\mathbf{e}}^{(k)} = \begin{bmatrix} \mathbf{x}^{(k)} - \bar{\mathbf{x}}^{(k)} \\ \mathbf{z}^{(k)} - \bar{\mathbf{z}}^{(k)} \end{bmatrix}, \qquad \mathbf{w}^{(k)} = \begin{bmatrix} \alpha\widehat{\mathbf{W}}^{(k)}\mathbf{s}^{(k)} \\ (\mathbf{I} - \mathbf{W}^{(k)})\mathbf{s}^{(k)} \end{bmatrix}, \qquad \mathbf{G}^{(k)} = \begin{bmatrix} \widehat{\mathbf{W}}^{(k)} & -\alpha\widehat{\mathbf{W}}^{(k)} \\ \mathbf{0} & \widehat{\mathbf{W}}^{(k)} \end{bmatrix}.\end{gathered} \tag{18}$$

5.2 Properties of the Consensus Error Matrix

To analyze the asymptotic behavior of the consensus error $\hat{\mathbf{e}}^{(k)}$ in (17b), we need to examine the spectral norm of $\mathbf{G}^{(k)}$, or that of the product of a sequence of $\{\mathbf{G}^{(k)}\}$. In this section, we establish two lemmas that bound the spectral norm of $\coprod_{k=i+j}^{i} \mathbf{G}^{(k)} \overset{\triangle}{=} \mathbf{G}^{(i+j)}\mathbf{G}^{(i+j-1)}\cdots\mathbf{G}^{(i)}$ in two different scenarios. The first lemma shows that the matrix sequence $\{\mathbf{G}^{(k)}\}$ also has the finite-time consensus property (with parameter τ). The second lemma considers the case where the entire sequence of $\{W^{(l)}\}_{l=0}^{\tau-1}$ is not visited, and exact averaging does not hold.

Recall the assumption that $\{W^{(l)}\}_{l=0}^{\tau-1}$ satisfies the finite-time consensus property (3), and so

$$\coprod_{i=j+k-1}^{j} W^{(i)} = W^{(j+k-1)}W^{(j+k-2)}\cdots W^{(j)} = \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\mathsf{T},$$

where $k \in \mathbb{N}_{\geq \tau}$ and $W^{(i)} \triangleq W^{(\mathrm{mod}(i,\tau))}$; that is, finite-time consensus holds for the sequence $\{W^{(i)}\}_{i=j}^{j+k-1}$ as long as $k \geq \tau$. Hence, for any integer $k \geq \tau$, we define $m \triangleq \lfloor k/\tau \rfloor - 1$. (Technically, m is a function of k, *i.e.*, $m(k)$, but we shorthand the notation to m for brevity.) From the definition of m, it holds that $\tau \leq k - m\tau \leq 2\tau$. This fact further implies that from iteration $m\tau$ to k, there must be at least τ iterations, and thus finite-time consensus holds for the sequence $\{W^{(i)}\}_{i=m\tau}^{k}$. Then, we have the following two lemmas on the bounds of the spectral norm of $\{\mathbf{G}^{(i)}\}$.

Lemma 5.1. *Let $\tau \in \mathbb{N}_{\geq 1}$ be the finite-time consensus parameter of the matrix sequence $\{W^{(l)}\}_{l=0}^{\tau-1} \subset \mathbb{R}^{n \times n}$. Given $k \in \mathbb{N}_{\geq \tau}$, consider the sequence of $\{\mathbf{G}^{(i)}\}_{i=m\tau}^{k-1}$ defined in (18), where $m = \lfloor k/\tau \rfloor - 1$. The spectral norm of the product of the matrices in the sequence $\{\mathbf{G}^{(i)}\}$ satisfies*

$$\left\| \coprod_{i=k-1}^{m\tau} \mathbf{G}^{(i)} \right\|_2 = 0.$$

Lemma 5.2. *Let $\tau \in \mathbb{N}_{\geq 1}$ be the finite-time consensus parameter of the matrix sequence $\{W^{(l)}\}_{l=0}^{\tau-1} \subset \mathbb{R}^{n \times n}$. Given $k \in \mathbb{N}_{\geq 1}$, consider the sequence of $\{\mathbf{G}^{(i)}\}_{i=j+1}^{k-1}$ defined in (18), where $j \in [m\tau, k-2]$ and $m = \lfloor k/\tau \rfloor - 1$. The spectral norm of the product of the matrices in the sequence $\{\mathbf{G}^{(i)}\}$ satisfies*

$$\left\| \coprod_{i=k-1}^{j+1} \mathbf{G}^{(i)} \right\|_2 \leq 1 + \alpha(\tau - 1).$$

5.3 Convergence Analysis for Algorithm 1

This section presents the convergence results of Algorithm 1, which relies on two important inequalities. The *descent inequality* establishes the convergence of the averaged iterates $\bar{x}^{(k)}$ to a first-order stationary point of (1). The *consensus inequality* reveals the per-iteration behavior of the consensus error $\hat{\mathbf{e}}^{(k)}$, and will be used to show that each agent's parameter $x_i^{(k)}$ converges to the average $\bar{x}^{(k)}$.

Lemma 5.3 (Descent Inequality)**.** *Let Assumptions 5.1 and 5.2 hold, let the mixing matrices $\{W^{(l)}\}_{l=0}^{\tau-1} \subset \mathbb{R}^{n \times n}$ satisfy Definition 3.1, and let the stepsize η satisfy $\eta \in (0, \frac{1}{2L}]$. Then, the sequence generated by Algorithm 1 satisfies*

$$\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 + \mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2 \leq \frac{4}{\alpha\eta}\Big(\mathbb{E}\, f(\bar{x}^{(k)}) - \mathbb{E}\, f(\bar{x}^{(k+1)})\Big) + \frac{2L^2}{n}\mathbb{E}\|\hat{\mathbf{e}}^{(k)}\|^2 + \frac{2\alpha\eta L\sigma^2}{n}, \tag{19}$$

for all $k \in \mathbb{N}$.

The left-hand side of (19) is the (expected) gradient norm, which aligns with our main convergence result (see Theorem 5.5). Such a convergence result is common in stochastic unconstrained optimization; see, *e.g.*, Bertsekas and Tsitsiklis (2000), which analyzes (centralized) Stochastic Gradient methods (SGD).

The second lemma is on the consensus inequality and establishes that all agents' parameters converge to their average.

Lemma 5.4 (Consensus Inequality)**.** *Let Assumptions 5.1 and 5.2 hold, let the mixing matrices $\{W^{(l)}\}_{l=0}^{\tau-1} \subset \mathbb{R}^{n \times n}$ satisfy Definition 3.1, and let the stepsizes satisfy $\alpha \in \big(0, \frac{1}{\tau}\big]$, $\eta \in \big(0, \frac{1}{4\sqrt{5}\tau L}\big]$. Then, for $T \in \mathbb{N}_{\geq \tau}$, the sequence generated by Algorithm 1 satisfies*

$$\frac{1}{T+1}\sum_{k=0}^{T}\mathbb{E}\|\hat{\mathbf{e}}^{(k)}\|^2 \leq \frac{2}{T+1}\sum_{k=0}^{\tau-1}\mathbb{E}\|\hat{\mathbf{e}}^{(k)}\|^2 + \frac{64n\eta^4L^2}{T+1}\sum_{k=0}^{T}\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 + (48\tau\eta^2 + 64n\eta^4L^2)\sigma^2. \tag{20}$$

Note that the first summation term on the right-hand side of (20) relies on the first τ iterates of $\hat{\mathbf{e}}^{(k)}$, and recall that τ is a prescribed constant for Algorithm 1. Hence, the term $\frac{2}{T+1}\sum_{k=0}^{\tau-1}\mathbb{E}\|\hat{\mathbf{e}}^{(k)}\|^2$ can be viewed as a constant when we study the asymptotic behavior of the consensus error.

The consensus inequality (20) is used in tandem with the descent inequality (19) to show that the consensus error in (19) vanishes asymptotically. Hence, not only does the averaged parameter asymptotically reach a stationary point of (1), but all the agents' parameters also converge (because they reach a consensus). Theorem 5.5 formally presents this result.

Theorem 5.5. *Let Assumptions 5.1 and 5.2 hold, let the mixing matrices* $\{W^{(l)}\}_{l=0}^{\tau-1} \subset \mathbb{R}^{n\times n}$ *satisfy Definition 3.1, and let the stepsizes satisfy* $\alpha \in \left(0, \frac{1}{\tau}\right]$, $\eta \in \left(0, \frac{1}{4\sqrt{5}\tau L}\right]$. *Then, for* $T \in \mathbb{N}_{\geq \tau}$, *the sequence* $\{\mathbf{x}^{(k)}\}$ *generated by Algorithm 1 satisfies*

$$\frac{1}{T+1}\sum_{k=0}^{T}\left(\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 + \mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2\right) \leq \frac{\gamma_1\tau^2L^3}{T} + \frac{\gamma_2\sigma^2}{\tau} + \frac{\gamma_3\sigma^2}{n\tau^2},$$

with some constants $(\gamma_1, \gamma_2, \gamma_3) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0} \times \mathbb{R}_{>0}$.

After further tuning the stepsize η (and α), we can derive the final rate of convergence. (The stepsize tuning technique is common in the literature (Alghunaim, 2023; Karimireddy et al., 2020; Koloskova et al., 2020; Stich, 2019).)

Corollary 5.6. *Let Assumptions 5.1 and 5.2 hold, let the mixing matrices* $\{W^{(l)}\}_{l=0}^{\tau-1} \subset \mathbb{R}^{n\times n}$ *satisfy Definition 3.1, and let* $\alpha \in \left(0, \frac{1}{\tau}\right]$, $\eta = \min\left\{\left(\frac{c_0}{c_1T}\right)^{\frac{1}{2}}, \left(\frac{c_0}{c_2T}\right)^{\frac{1}{3}}, \frac{1}{4\sqrt{5}\tau L}\right\}$, *where* $c_0 = \tau L^2$, $c_1 = \frac{L\sigma^2}{n\tau}$, *and* $c_2 = \tau L^2\sigma^2$. *Then, for* $T \in \mathbb{N}_{\geq \tau}$, *the sequence* $\{\mathbf{x}^{(k)}\}$ *generated by Algorithm 1 satisfies*

$$\frac{1}{T+1}\sum_{k=0}^{T}\left(\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 + \mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2\right) \leq \frac{\gamma_4\tau^2L^3}{T} + \frac{\gamma_5\tau L^2\sigma^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \gamma_6\left(\frac{L^3\sigma}{nT}\right)^{\frac{1}{2}},$$

for some constants $(\gamma_4, \gamma_5, \gamma_6) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0} \times \mathbb{R}_{>0}$.

Similarly to Ying et al. (2021), we weaken the rate's dependence on L by imposing a warm-up strategy (*e.g.*, AllReduce (Assran et al., 2019)) to force all agents' parameters and tracking variables in the first period to be the same. This leads us to the following corollary.

Corollary 5.7. *Let Assumptions 5.1 and 5.2 hold, let the mixing matrices* $\{W^{(l)}\}_{l=0}^{\tau-1} \subset \mathbb{R}^{n\times n}$ *satisfy Definition 3.1, and let* $\alpha \in \left(0, \frac{1}{\tau}\right]$, $\eta = \min\left\{\left(\frac{c_0}{c_1T}\right)^{\frac{1}{2}}, \left(\frac{c_0}{c_2T}\right)^{\frac{1}{3}}, \frac{1}{4\sqrt{5}\tau L}\right\}$, *where* $c_0 = \tau L^2$, $c_1 = \frac{L\sigma^2}{n\tau}$, *and* $c_2 = \tau L^2\sigma^2$. *Suppose a warm-up strategy (e.g., AllReduce) is applied to force all agents' parameters and tracking variables in the first period to be the same:*

$$\sum_{k=0}^{\tau-1}\left(\mathbb{E}\|\mathbf{x}^{(k)} - \bar{\mathbf{x}}^{(k)}\|^2 + \mathbb{E}\|\mathbf{g}^{(k)} - \bar{\mathbf{g}}^{(k)}\|^2\right) = 0.$$

Then, for $T \in \mathbb{N}_{\geq \tau}$, *the sequence* $\{\mathbf{x}^{(k)}\}$ *generated by Algorithm 1 satisfies*

$$\frac{1}{T+1}\sum_{k=0}^{T}\left(\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 + \mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2\right) \leq \frac{\gamma_7\tau^2L}{T} + \gamma_8\tau\left(\frac{L\sigma}{T}\right)^{\frac{2}{3}} + \gamma_9\left(\frac{L\sigma}{nT}\right)^{\frac{1}{2}},$$

for some constants $(\gamma_7, \gamma_8, \gamma_9) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0} \times \mathbb{R}_{>0}$.

Discussion and comparison with other analyses for TV-GT. We note that our rate depends on the finite-time consensus parameter τ, and, remarkably, is independent of the connectivity of any of the individual elements of the finite-time consensus graphs. This contrasts, for example, the rate derived in Song et al. (2022), which depends on the smallest connectivity (in expectation) of the set of all time-varying topologies used in the algorithm. Thus, their analysis cannot handle finite-time consensus graphs because some elements in such a deterministic topology sequence can be disconnected. The underlying reason for the incapability is that their analysis examines the "worst" topology in the sequence but fails to consider the joint effect of the entire topology sequence.

Moreover, the superiority of GT-FT can be demonstrated via comparison with TV-GT restricted to the static exponential graphs. Without loss of generality, we assume the number of agents is $n = 2^\tau$ for some $\tau \in \mathbb{N}_{\geq 1}$, and a number of τ one-peer exponential graphs (satisfying Definition 3.1) are used in GT-FT. In this case, the best existing rate (to our knowledge) (Song et al., 2022) depends on the spectral gap $1 - \rho = 2/(1 + \tau)$ of the static exponential graph with node size $n = 2^\tau$ (Ying et al., 2021), and reads as

$$O\left(\frac{\sigma^2}{n\epsilon^2}\right) + O\left(\frac{\sigma(1+\tau)^{\frac{3}{2}}}{\epsilon^{\frac{3}{2}}}\right) + O\left(\frac{(1+\tau)^2}{\epsilon}\right),$$

where the Lipschitz constant L is omitted in Song et al. (2022). In comparison, it follows from Corollary 5.7 that the iteration complexity of GT-FT using a sequence of τ one-peer exponential graphs is given by

$$O\left(\frac{L\sigma}{n\epsilon^2}\right) + O\left(\frac{\tau^{\frac{3}{2}}L\sigma}{\epsilon^{\frac{3}{2}}}\right) + O\left(\frac{\tau^2 L}{\epsilon}\right).$$

Ignoring the Lipschitz constants as in Song et al. (2022), we find that this implementation of GT-FT has a slightly lower iteration complexity than TV-GT using a static exponential graph. Remarkably, this slight improvement in convergence rate comes with a significant decrease in communication cost: The maximum degree of a static exponential graph is $\Omega(\log_2 n)$ while that of a single one-peer exponential graph is $\Omega(1)$. Similar comparison can also be performed for p-peer hyper-cuboids and static hyper-cuboids.

6 Numerical Experiments

In this section, we present numerical results to verify our theoretical findings. The purpose of the numerical experiments is two-fold. First, numerical evidence is provided to verify that the graph sequences studied in Section 3 satisfy the finite-time consensus property. Moreover, we conduct numerical experiments that incorporate the studied graph sequences into decentralized optimization algorithms. The numerical results demonstrate that GT-FT using graph sequences with finite-time consensus property converges at the same rate as TV-GT using the static counterparts.

6.1 Finite-Time Consensus Property

In this section, we verify in numerical experiments that the presented topology sequences satisfy the exact averaging property within a finite number of iterations. To do so, we simulate an average consensus problem. Each agent is initialized with a random vector $x_i^{(0)} \sim \mathcal{N}(0, \Sigma)$ drawn from a Gaussian distribution (with $\Sigma \in \mathbb{S}^d_{++}$). The iterates $x_i^{(k)}$ evolve according to the recursion $x_i^{(k+1)} = W^{(k)} x_i^{(k)}$ for $i = 1, \ldots, n$, and the consensus error at each iteration is defined as

$$\Xi^{(k)} \triangleq \frac{1}{n} \sum_{i=1}^{n} \| x_i^{(k)} - \bar{x}^{(0)} \|^2,$$

where we define $\bar{x}^{(0)} \triangleq \frac{1}{n} \sum_{i=1}^{n} x_i^{(0)}$.

Besides illustrating the exact averaging property of the studied time-varying topologies, we compare these dynamic graphs with their corresponding static variant. Given a sequence of graphs $\mathcal{G}^{(l)} = (\mathcal{V}, W^{(l)}, \mathcal{E}^{(l)})$ with $l = 0, 1, \ldots, \tau - 1$, its static variant $\mathcal{G}^{(\text{static})} = \left(\mathcal{V}, W^{(\text{static})}, \mathcal{E}^{(\text{static})}\right)$ is defined by

$$\mathcal{E}^{(\text{static})} = \mathcal{E}^{(0)} \cup \mathcal{E}^{(1)} \cup \cdots \cup \mathcal{E}^{(\tau-1)},$$

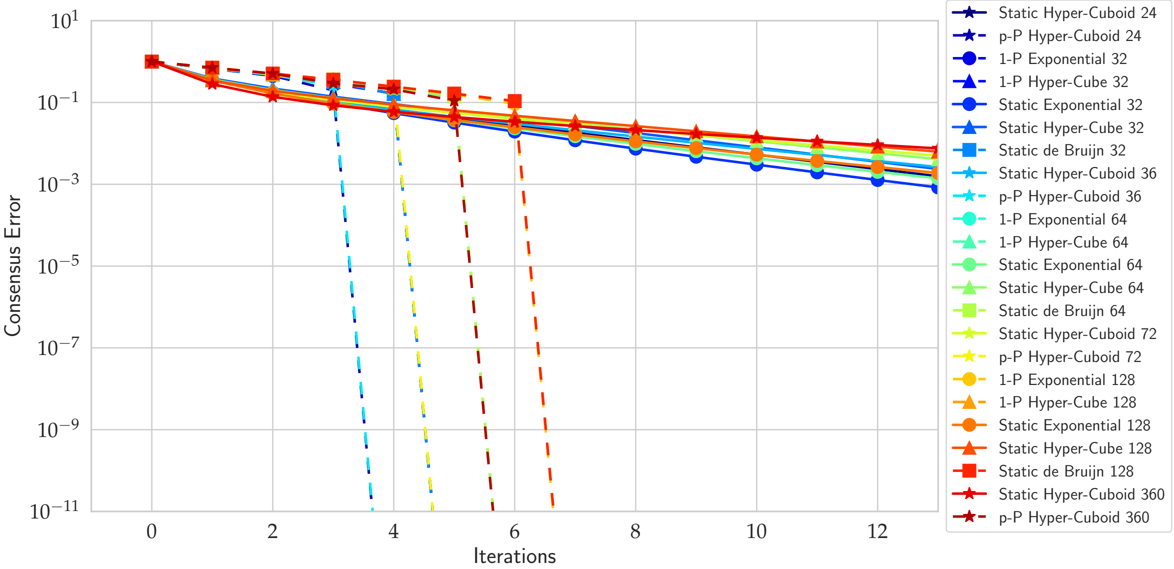


Figure 3: Consensus error versus the number of iterations. The legend is composed of three parts. The first part is either “Static”, “1-P”, or “p-P”, standing for static graphs, one-peer time-varying graphs, and p-peer time-varying graphs, respectively. The second part of the legend describes the graph type: exponential, hyper-cube, de Bruijn, or hyper-cuboid. The third part is for the number of agents. All graphs satisfying Definition 3.1 are plotted with dashed lines, while others are plotted with a solid line.

and the weight matrix $W^{(\text{static})}$ is normalized to be doubly stochastic. (More discussion on the static variants of a sequence of (dynamic) graphs can be found in de Bruijn (1946); Harary et al. (1988); Shi et al. (2016); Ying et al. (2021).)

Figure 3 presents the simulation results. In Figure 3, the topology sequences satisfying Definition 3.1 have a steep drop in the consensus error (see dashed lines), indicating the vanishing of the consensus error. For topologies that do not satisfy Definition 3.1, we observe that the consensus error decreases asymptotically (at an exponential rate).

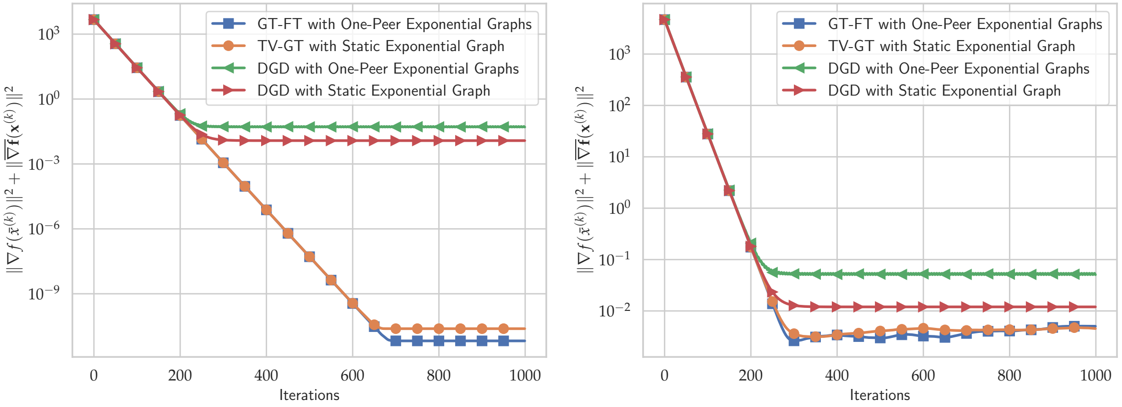
6.2 Gradient Tracking with Finite-Time Consensus Topologies

The simulation in Section 6.1 demonstrates the usefulness of some topology sequences in solving the average consensus problem in a few iterations. However, the benefits of using these topology sequences may not necessarily translate when we use decentralized optimization algorithms. In this section, we provide numerical evidence to verify the theoretical guarantees established in Section 5 and to demonstrate the potential benefits of the finite-time consensus property in decentralized optimization algorithms.

We apply GT-FT to solve the least squares problem with a nonconvex regularization term:

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n}\|A_i x - b_i\|^2 + \mu \sum_{j=1}^{d} \frac{x[j]^2}{1+x[j]^2},$$

where the optimization variable is $x \in \mathbb{R}^d$, $x[j]$ denotes the jth component of x, and the data $\{A_i, b_i\}$ is held exclusively by agent i. This problem instance is used extensively when studying decentralized algorithms for nonconvex problems, and we follow existing conventions to construct the problem data (see, *e.g.*, Alghunaim and Yuan (2022); Xin et al. (2021)). The entries in each data matrix $A_i \in \mathbb{R}^{m\times d}$ are drawn IID from the distribution $\mathcal{N}(0,1)$, and so are the vectors $\{\tilde{x}_i\}_{i=1}^{n} \subset \mathbb{R}^d$. The vector $b_i \in \mathbb{R}^d$ is then computed by $b_i = A_i\tilde{x}_i + \delta z_i$, where $\delta \in \mathbb{R}_{>0}$ is a prescribed constant and $z_i \in \mathbb{R}^d$ is random noise with entries drawn IID



(a) Agents use true gradients.

(b) Agents use stochastic gradients.

Figure 4: Comparison of the use of one-peer exponential graphs and static exponential graphs in decentralized optimization algorithms. One-peer exponential graphs are used in GT-FT and DGD, and static exponential graphs are used in TV-GT and DGD.

from $\mathcal{N}(0,1)$. In all the experiments, we set $m = 500$, $d = 20$, and $\delta = 10$. The number of agents (n) might vary in the experiments and will be specified later.

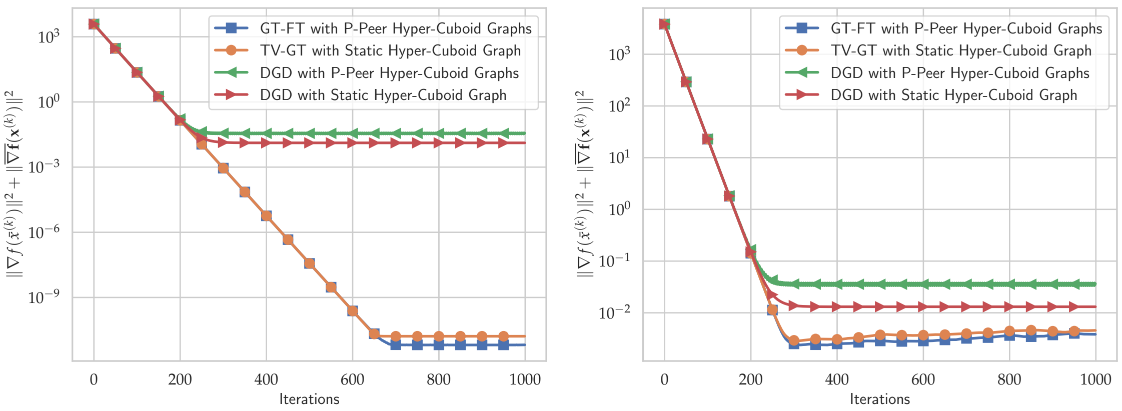
6.2.1 One-Peer Exponential Graphs

We first analyze the use of one-peer exponential graphs in decentralized optimization algorithms. We consider a sequence of one-peer exponential graphs of size $n = 64$ and its static counterpart. The sequence of one-peer exponential graphs is incorporated into GT-FT and DGD, while the static exponential graph is incorporated into TV-GT and DGD. For GT-FT and TV-GT, we use the stepsizes $\alpha = 10^{-4}$, $\eta = 1$ while for DGD, we use the stepsize $\alpha = 10^{-4}$. Furthermore, we consider the case where agents have access to the true gradients and the case where agents only have access to the stochastic gradient estimates. The stochastic gradient is formed by adding Gaussian noise to the true gradient, *i.e.*, $\widehat{\nabla f}_i(x) = \nabla f_i(x) + s_i$ with $s_i \sim \mathcal{N}(0, \sigma^2 I_d)$. The magnitude of the gradient noise can be controlled by the constant σ^2, and we set $\sigma^2 = 10^{-4}$ in the experiments.

The simulation results are presented in Figure 4 and match the theoretical findings in Section 5.3. We observe from Figure 4 that when true gradients are used, the convergence rate of decentralized algorithms using one-peer exponential graphs is similar to that using static exponential graphs. For example, the convergence rate of GT-FT using one-peer exponential graphs matches that of TV-GT using static exponential graphs. Similar performance is observed for DGD, but DGD using one-peer exponential graphs converges to a slightly worse solution than DGD using static exponential graphs. The same observations translate when stochastic gradients are used. In view of this similar rate of convergence, using one-peer exponential graphs in decentralized algorithms shows another advantage (besides the finite-time consensus property). Recall that the maximum degree of a static exponential graph is $\Omega(\log_2 n)$ while that of a single one-peer exponential graph is $\Omega(1)$. Thus, the mixing matrices $\{W^{(l)}\}_{l=0}^{\tau-1}$ of one-peer exponential graphs are much sparser than the weight matrix $W^{(\text{static})}$ of the static exponential graph. In the context of decentralized algorithms, it means that using one-peer exponential graphs would reduce communication costs compared with using the static counterpart.

6.2.2 p-Peer Hyper-Cuboids

Next, we analyze the use of p-peer hyper-cuboids in decentralized optimization algorithms. We consider a sequence of p-peer hyper-cuboids of size $n = 72$ and its static counterpart, the static hyper-cuboid. The



(a) Agents use true gradients.

(b) Agents use stochastic gradients.

Figure 5: Comparison of the use of p-peer hyper-cuboids and static hyper-cuboids in decentralized optimization algorithms. p-peer hyper-cuboids are used in GT-FT and DGD, and static hyper-cuboids are used in TV-GT and DGD.

sequence of p-peer hyper-cuboids is incorporated into GT-FT and DGD, while the static hyper-cuboid is incorporated into TV-GT and DGD. For GT-FT and TV-GT, we use the stepsizes $\alpha = 10^{-4}$, $\eta = 1$ while for DGD, we use the stepsize $\alpha = 10^{-4}$. Again, we consider the case where agents have access to the true gradients and the case where stochastic gradients are used. The stochastic gradients are formed just as in the one-peer exponential experiment, and we again use $\sigma^2 = 10^{-4}$.

The simulation results are presented in Figure 5 and match the theoretical findings in Section 5.3. It is observed from Figure 5 that p-peer hyper-cuboids exhibit the same behavior as one-peer exponential graphs. When true gradients are available, the convergence rate of decentralized algorithms using p-peer hyper-cuboids is similar to that using static hyper-cuboids. For example, the convergence rate of GT-FT using p-peer hyper-cuboids matches that of TV-GT using static hyper-cuboids. Similar performance is observed for DGD, but DGD using p-peer hyper-cuboids converges to a slightly worse solution. The same observations translate when agents only have access to the stochastic gradients. Again, this matching convergence rate indicates that using p-peer hyper-cuboids reduces the per-iteration communication cost as each p-peer hyper-cuboid is much sparser than its static counterpart. Note that the maximum degree of a p-peer hyper-cuboid is the largest prime factor of n, while the maximum degree of a static hyper-cuboid is equal to the sum of the prime factors of n minus the number of prime factors plus one.

6.2.3 Discussion on GT-FT and DGD

We conclude this section with a short discussion on the performance of GT-FT and DGD. When true gradients are used (and a constant stepsize is applied), GT-FT converges to a significantly better solution than DGD. The theoretical rationale for the better performance of GT methods over DGD has been thoroughly studied in the literature (Alghunaim and Yuan, 2022; Alghunaim et al., 2021; Koloskova et al., 2020; Yuan et al., 2016). With a constant stepsize, DGD converges to a sub-optimal solution biased proportionally to the magnitude of heterogeneity between agents (Koloskova et al., 2020; Yuan et al., 2016). In comparison, as a GT method, GT-FT is able to correct this bias caused by heterogeneity, and converges to a better solution (Alghunaim and Yuan, 2022; Alghunaim et al., 2021). However, when agents can only use stochastic gradients, our experiments indicate that the difference in the quality of the solutions returned by GT-FT and DGD is much smaller. This is because both the gradient noise and the heterogeneity bias affect the computed solution returned by GT-FT and DGD. This phenomenon has already been observed for GT methods and DGD using a static topology (Alghunaim and Yuan, 2022).

7 Conclusions

We study several sequences of graphs that satisfy the finite-time consensus property, including the one-peer exponential graphs, one-peer hyper-cubes, p-peer hyper-cuboids, and de Bruijn graphs. For each class of graphs, we present an explicit weight matrix representation and theoretically justify their finite-time consensus property. In particular, to the best of our knowledge, p-peer hyper-cuboids are the only available class of sparse graphs with arbitrary node sizes for which the finite-time consensus property is proven to hold. Moreover, we incorporate the studied topology sequences into the Gradient Tracking methods for decentralized optimization. Our analysis shows that the convergence rate of the proposed algorithmic scheme does not depend on the connectivity of any individual graph in the topology sequence, and the new scheme requires significantly lower communication costs compared with Gradient Tracking using the static counterpart of the topology sequence.

Although incorporating graph sequences with finite-time consensus in decentralized optimization algorithms has shown to be successful, several open questions remain. Despite the various graph sequences studied in Section 3, it is still unclear how to formulate sufficient and necessary conditions for the finite-time consensus property. Besides, the incorporation of finite-time consensus topologies into other decentralized algorithms, such as EXTRA or Exact Diffusion, is not straightforward and is left for future work.

References

Sulaiman A. Alghunaim. Local exact-diffusion for decentralized optimization and learning. *arXiv preprint arXiv:2302.00620*, 2023.

Sulaiman A. Alghunaim and Kun Yuan. A unified and refined convergence analysis for non-convex decentralized learning. *IEEE Transactions on Signal Processing*, 70:3264–3279, June 2022.

Sulaiman A. Alghunaim and Kun Yuan. An enhanced gradient-tracking bound for distributed online stochastic convex optimization. *arXiv preprint arXiv:2301.02855*, 2023.

Sulaiman A. Alghunaim, Ernest K. Ryu, Kun Yuan, and Ali H. Sayed. Decentralized proximal gradient algorithms with linear convergence rates. *IEEE Transactions on Automatic Control*, 66(6):2787–2794, June 2021.

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, 2019.

Amotz Bar-Noy, Shlomo Kipnis, and Baruch Schieber. An optimal algorithm for computing census functions in message-passing systems. *Parallel Processing Letters*, 3(01):19–23, 1993.

Dimitri P. Bertsekas and John N. Tsitsiklis. Gradient convergence in gradient methods with errors. *SIAM Journal on Optimization*, 10(3):627–642, 2000.

Federico S. Cattivelli and Ali H. Sayed. Diffusion LMS strategies for distributed estimation. *IEEE Trans. Signal Process*, 58(3):1035, 2010.

Harold Scott Macdonald Coxeter. *Regular Polytopes*. Courier Corporation, 1973.

Tao Cui, Lijun Chen, and Tracey Ho. Distributed optimization in wireless networks using broadcast advantage. In *2007 46th IEEE Conference on Decision and Control*, pages 5839–5844, 2007.

Nicolaas G. de Bruijn. A combinatorial problem. *Proceedings of the Section of Sciences of the Koninklijke Nederlandse Akademie van Wetenschappen te Amsterdam*, 49(7):758–764, 1946.

Jean-Charles Delvenne, Ruggero Carli, and Sandro Zampieri. Optimal strategies in the average consensus problem. *Systems & Control Letters*, 58(10-11):759–765, 2009.

Paolo Di Lorenzo and Gesualdo Scutari. Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, 2016.

Lisang Ding, Kexin Jin, Bicheng Ying, Kun Yuan, and Wotao Yin. DSGD-CECA: Decentralized SGD with communication-optimal exact consensus algorithm. *arXiv preprint arXiv:2306.00256*, 2023.

Frank Harary, John P. Hayes, and Horng-Jyh Wu. A survey of the theory of hypercube graphs. *Computers & Mathematics with Applications*, 15(4):277–289, 1988. ISSN 0898-1221.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2013.

Norm Jouppi, George Kurian, Sheng Li, Peter Ma, Rahul Nagarajan, Lifeng Nai, Nishant Patil, Suvinay Subramanian, Andy Swing, Brian Towles, Cliff Young, Xiang Xhou, Zongwei Zhou, and David Patterson. TPU v4: an optically reconfigurable supercomputer for machine learning with hardware support for embeddings. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, pages 1–14, 2023.

Soummya Kar and José M. F. Moura. Distributed consensus algorithms in sensor networks: Link failures and channel noise. *IEEE Transactions on Signal Processing*, 57(1):355–369, Jan. 2009.

Soummya Kar and José M. F. Moura. Distributed consensus algorithms in sensor networks: Quantized data and random link failures. *IEEE Transactions on Signal Processing*, 58(3):1383–1400, 2010.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized SGD with changing topology and local updates. In *International Conference on Machine Learning*, pages 5381–5393, 2020.

Anastasiia Koloskova, Tao Lin, and Sebastian U. Stich. An improved analysis of gradient tracking for decentralized machine learning. *Advances in Neural Information Processing Systems*, 34:11422–11435, 2021.

Daniel Krenn, Dimbinaina Ralaivaosaona, and Stephan Wagner. Multi-base representations of integers: asymptotic enumeration and central limit theorems. *Applicable Analysis and Discrete Mathematics*, 9(2): 285–312, 2015.

Guanghui Lan, Soomin Lee, and Yi Zhou. Communication-efficient algorithms for decentralized and stochastic optimization. *Mathematical Programming*, 2018.

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, Broomfield, CO, Oct. 2014.

Zhi Li, Wei Shi, and Ming Yan. A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *IEEE Transactions on Signal Processing*, 67(17):4494–4506, Sept. 2019.

Cassio G. Lopes and Ali H. Sayed. Diffusion least-mean squares over adaptive networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 917–920, Honolulu, HI, USA, 2007.

Angelia Nedić, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017.

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

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

Shi Pu, Wei Shi, Jinming Xu, and Angelia Nedić. Push–pull gradient methods for distributed optimization in networks. *IEEE Transactions on Automatic Control*, 66(1):1–16, 2020.

Guannan Qu and Na Li. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3):1245–1260, Sept. 2018.

Gesualdo Scutari and Ying Sun. Distributed nonconvex constrained optimization over time-varying digraphs. *Mathematical Programming*, 176(1-2):497–544, 2019.

Guodong Shi, Bo Li, Mikael Johansson, and Karl Henrik Johansson. Finite-time convergent gossiping. *IEEE/ACM Transactions on Networking*, 24(5):2782–2794, oct 2016.

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.

Zhuoqing Song, Weijian Li, Kexin Jin, Lei Shi, Ming Yan, Wotao Yin, and Kun Yuan. Communication-efficient topologies for decentralized learning with $O(1)$ consensus rate. *Advances in Neural Information Processing Systems*, 35:1073–1085, 2022.

Sebastian U. Stich. Unified optimal analysis of the (stochastic) gradient method. *arXiv preprint arXiv:1907.04232*, 2019.

Srinivasan Sundhar Ram, Angelia Nedić, and Venugopal V. Veeravalli. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of Optimization Theory and Applications*, 147(3): 516–545, 2010.

Yuki Takezawa, Ryoma Sato, Han Bao, Kenta Niwa, and Makoto Yamada. Beyond exponential graph: Communication-efficient topologies for decentralized learning via finite-time convergence. *arXiv preprint arXiv:2305.11420*, 2023.

Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D^2: Decentralized training over decentralized data. In *International Conference on Machine Learning*, pages 4848–4856, Stockholm, Sweden, 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 S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Chenguang Xi, Van Sy Mai, Ran Xin, Eyad H Abed, and Usman A. Khan. Linear convergence in optimization over directed graphs with row-stochastic matrices. *IEEE Transactions on Automatic Control*, 63(10):3558–3565, 2018.

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, 2021.

Jinming Xu, Shanying Zhu, Yeng Chai Soh, and Lihua Xie. Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. In Proc. 54th *IEEE Conference on Decision and Control (CDC)*, pages 2055–2060, Osaka, Japan, 2015.

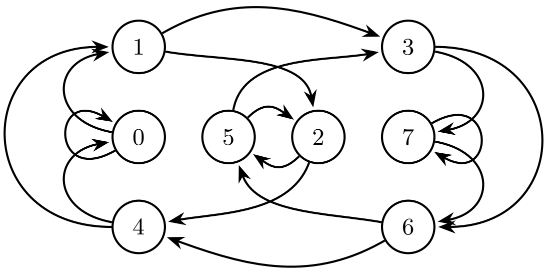


Figure 6: The de Bruijn graph of with node size $n = 8$, $p = 2$ and $\tau = 3$.

Bicheng Ying, Kun Yuan, Yiming Chen, Hanbin Hu, Pan Pan, and Wotao Yin. Exponential graph is provably efficient for decentralized deep training. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 13975–13987. Curran Associates, Inc., 2021.

Kun Yuan, Qing Ling, and Wotao Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

Kun Yuan, Bicheng Ying, Xiaochuan Zhao, and Ali H. Sayed. Exact diffusion for distributed optimization and learning–Part I: Algorithm development. *IEEE Transactions on Signal Processing*, 67(3):708–723, 2019. doi: 10.1109/TSP.2018.2875898.

Baosen Zhang, Albert Y. S. Lam, Alejandro D. Domínguez-García, and David Tse. An optimal and distributed method for voltage regulation in power distribution systems. *IEEE Transactions on Power Systems*, 30(4):1714–1726, 2015. doi: 10.1109/TPWRS.2014.2347281.

A More Details for de Bruijn Graphs

This section includes more details of the de Bruijn graphs. Section A.1 presents the Kronecker representation of de Bruijn graphs. Section A.2 describes the connection between de Bruijn graphs and p-peer hyper-cuboids and presents the proof of Proposition 3.5. In Section A.3, we apply Algorithm 1 with de Bruijn graphs and present the numerical results.

A.1 Definition and Kronecker representation

A de Bruijn graph has $n = p^\tau$ nodes with $(p, \tau) \in \mathbb{N}_{\geq 2} \times \mathbb{N}_{\geq 1}$, and the construction of its edges relies on a p-based representation of integers. Such a representation is an element in the group $\prod_{i=0}^{\tau-1} \mathbb{N}_p$, where $\mathbb{N}_p$ is the group of nonnegative integers modulo p. As before, we shorten the notation by overloading binary representation and denote the p-based representation of $i \in \{0, 1, \ldots, n-1\}$ as $(i_{\tau-1}i_{\tau-2}\ldots i_0)_p$, so that we can also re-write $\{a\} \times \{b\}$ as $(a, b)_p$. Then, recall from (12) that the mixing matrix of a de Bruijn graph with node size $n = p^\tau$ is defined by

$$w_{ij} = \begin{cases} \frac{1}{q} & \text{if } (i_{\tau-2}i_{\tau-3}\ldots i_0)_p = (j_{\tau-1}j_{\tau-2}\ldots j_1)_p \\ 0 & \text{otherwise.} \end{cases}$$

The above connection condition is equivalent to that a (directed) edge exists from node i to node j if the p-based representation of i is shifted to the left and a new bit is added to the end of the representation that equals j. As an example, consider the de Bruijn graph with node size $n = 2^3$. The vertex $i = 3$ has a 2-based representation of $(011)_2$. Shifting to the left and adding a new bit to the end of the representation leads us to find $(110)_2$ and $(111)_2$. So this means an edge exists from $i = 3$ to $j = 6, 7$.

Furthermore, similar to the p-peer hyper-cuboids, we show that the mixing matrix of a de Bruijn graph also has a concise Kronecker product form (with minor modifications). To do so, we need the following lemma on the properties of Kronecker products.

Lemma A.1. *For $n = p^\tau$ with $(p, \tau) \in \mathbb{N}_{\geq 2} \times \mathbb{N}_{\geq 1}$ and for all vectors $\{a_l\}_{l=0}^{\tau-1}$, there exists a permutation matrix $P_\mathrm{s} \in \mathbb{R}^{n\times n}$ such that*

$$P_\mathrm{s}(a_{\tau-1} \otimes a_{\tau-2} \otimes \cdots \otimes a_0) = a_0 \otimes a_{\tau-1} \otimes \cdots \otimes a_1.$$

Moreoever, it holds that $P_\mathrm{s}^\tau = I_n$.

Proof. The first result follows directly from Horn and Johnson (2013, Section 12.3). The second result follows by repeatedly applying the first one:

$$P_\mathrm{s}^\tau(a_{\tau-1} \otimes a_{\tau-2} \otimes \cdots \otimes a_0) = P_\mathrm{s}^{\tau-1}(a_0 \otimes a_{\tau-1} \otimes \cdots \otimes a_1) = \cdots = a_{\tau-1} \otimes a_{\tau-2} \otimes \cdots \otimes a_0. \qquad \square$$

This permutation matrix P_s is called the *perfect shuffle matrix* (Horn and Johnson, 2013, Section 12.3), hence the subscript "s" in P_s. We then present the matrix form of de Bruijn graphs:

$$\begin{aligned}
W &= \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} e_i e_j^\mathsf{T} \\
&= \underbrace{\sum_{i_{\tau-1}}\sum_{i_{\tau-2}}\cdots\sum_{i_1}\sum_{i_0}}_{i}\underbrace{\sum_{j_{\tau-1}}\sum_{j_{\tau-2}}\cdots\sum_{j_1}\sum_{j_0}}_{j} w_{ij}(\hat{e}_{i_{\tau-1}} \otimes \cdots \otimes \hat{e}_{i_0})(\hat{e}_{j_{\tau-1}} \otimes \cdots \otimes \hat{e}_{j_0})^\mathsf{T} \\
&= \sum_{i_{\tau-1}}\sum_{i_{\tau-2}}\cdots\sum_{i_1}\sum_{i_0}\sum_{j_{\tau-1}}\sum_{j_{\tau-2}}\cdots\sum_{j_1}\sum_{j_0} w_{ij}(\hat{e}_{i_{\tau-1}} \otimes \cdots \otimes \hat{e}_{i_0})(\hat{e}_{j_0} \otimes \cdots \otimes \hat{e}_{j_1})^\mathsf{T} P_\mathrm{s}^\mathsf{T} \\
&= \sum_{i_{\tau-1}}\sum_{j_0}\sum_{i_{\tau-2}}\cdots\sum_{i_1}\sum_{i_0}\Big(w_{ij}(\hat{e}_{i_{\tau-1}}\hat{e}_{j_0}^\mathsf{T}) \otimes (\hat{e}_{i_{\tau-2}}\hat{e}_{i_{\tau-2}}^\mathsf{T}) \otimes \cdots \otimes (\hat{e}_{i_1}\hat{e}_{i_1}^\mathsf{T}) \otimes (\hat{e}_{i_0}\hat{e}_{i_0}^\mathsf{T})\Big) P_\mathrm{s}^\mathsf{T} \\
&= \Bigg(\Big(\sum_{i_{\tau-1}}\sum_{j_0} w_{ij}\hat{e}_{i_{\tau-1}}\hat{e}_{j_0}^\mathsf{T}\Big) \otimes \Big(\sum_{i_{\tau-2}}\hat{e}_{i_{\tau-2}}\hat{e}_{i_{\tau-2}}^\mathsf{T}\Big) \otimes \cdots \otimes \Big(\sum_{i_1}\hat{e}_{i_1}\hat{e}_{i_1}^\mathsf{T}\Big) \otimes \Big(\sum_{i_0}\hat{e}_{i_0}\hat{e}_{i_0}^\mathsf{T}\Big)\Bigg) P_\mathrm{s}^\mathsf{T} \\
&= \big(J_p \otimes I \otimes \cdots \otimes I \otimes I\big) P_\mathrm{s}^\mathsf{T},
\end{aligned} \tag{21}$$

where $J_p \triangleq \frac{1}{p}\mathbb{1}_p\mathbb{1}_p^\mathsf{T}$ and the above derivation uses the same properties as in (11a)–(11c).

Compared with the topology sequences studied in Section 3, de Bruijn graphs achieve finite-time consensus without varying the instance; $W^\tau = \frac{1}{n}\mathbb{1}\mathbb{1}^\mathsf{T}$ for $W \in \mathbb{R}^{n\times n}$ defined in (12). Delvenne et al. (2009) has already proved this result, and here we provide an alternative proof using the Kronecker representation (21).

Proposition A.2. *Given $n = p^\tau$ with $(p, \tau) \in \mathbb{N}_{\geq 2} \times \mathbb{N}_{\geq 1}$, let $W \in \mathbb{R}^{n\times n}$ be the weight matrix defined in (12). The matrix W is symmetric and doubly stochastic. In addition, it holds that*

$$W^\tau = \underbrace{W \cdots W}_{\tau \text{ times}} = \tfrac{1}{n}\mathbb{1}\mathbb{1}^\mathsf{T} = J_n;$$

i.e., the finite-time consensus property (3.1) holds with $W^{(0)} = \cdots = W^{(\tau-1)} := W$.

Proof. The key step to prove the finite-time consensus of de Bruijn graph is the following identity

$$P_\mathrm{s}\big(A_{\tau-1} \otimes A_{\tau-2} \otimes \cdots \otimes A_1 \otimes A_0\big)P_\mathrm{s}^\mathsf{T} = \Big(A_0 \otimes A_{\tau-1} \otimes \cdots \otimes A_2 \otimes A_1\Big), \tag{22}$$

where $\{A_i\}_{i=0}^{\tau-1}$ are any arbitrary $p \times p$ matrices. The above identity is straightforward to see because left-multiplying P_{s} is equivalent to shifting the rows of the matrix and right-multiplying $P_{\mathrm{s}}]^{\mathsf{T}}$ is equivalent to shift the columns of the matrix. Using this property, it follows that

$$\begin{aligned} W^2 &= \big(J_p \otimes I \otimes \cdots \otimes I \otimes I\big) P_{\mathrm{s}}^{\mathsf{T}} \big(J_p \otimes I \otimes \cdots \otimes I \otimes I\big) P_{\mathrm{s}}^{\mathsf{T}} \\ &= (P_{\mathrm{s}}^{\mathsf{T}})^2 P_{\mathrm{s}}^2 \big(J_p \otimes I \otimes \cdots \otimes I \otimes I\big)(P_{\mathrm{s}}^{\mathsf{T}})^2 P_{\mathrm{s}} \big(J_p \otimes I \otimes \cdots \otimes I \otimes I\big) P_{\mathrm{s}}^{\mathsf{T}} \\ &= (P_{\mathrm{s}}^{\mathsf{T}})^2 \big(I \otimes I \otimes J_p \otimes \cdots \otimes I\big)\big(I \otimes J_p \otimes \cdots \otimes I \otimes I\big) \\ &= (P_{\mathrm{s}}^{\mathsf{T}})^2 \big(I \otimes J_p \otimes J_p \otimes \cdots \otimes I\big), \end{aligned} \tag{23}$$

where (23) uses $P_{\mathrm{s}}^{\mathsf{T}} P_{\mathrm{s}} = I$.

Continuing the above step τ-times, we establish

$$\begin{aligned} W^\tau &= W^{\tau-3}\big(J_p \otimes I \otimes \cdots \otimes I \otimes I\big)(P_{\mathrm{s}}^{\mathsf{T}})^3\big(I \otimes J_p \otimes J_p \otimes \cdots \otimes I\big) \\ &\vdots \\ &= (P_{\mathrm{s}}^{\mathsf{T}})^\tau \big(J_p \otimes J_p \otimes J_p \otimes \cdots \otimes J_p\big) = J_n, \end{aligned}$$

where the last equality uses $(P_{\mathrm{s}})^\tau = I$.

The doubly stochastic property follows by definition of the de Bruijn graph, and can also be verified using the same approach as done for the hyper-cuboids. □

A.2 Connection between de Bruijn Graphs and p-Peer Hyper-Cuboids

In this section, we establish the connection between de Bruijn graphs and p-peer hyper-cuboids (when $n = p^\tau$). In particular, we restate and prove Proposition 3.5.

Proposition A.3. *Given $n = p^\tau$ with $(p, \tau) \in \mathbb{N}_{\geq 2} \times \mathbb{N}_{\geq 1}$, let $W_{\mathrm{db}} \in \mathbb{R}^{n \times n}$ be the weight matrix of the de Bruijn matrix defined in (12) and $\{W_{\mathrm{hc}}^{(l)}\}_{l \in \mathbb{N}}$ be the weight matrices of the p-peer hyper-cuboids defined in (8). Then for any $l \in \{0, 1, \ldots, \tau - 1\}$, it holds that*

$$W_{\mathrm{hc}}^{(\tau-l)} = (P_{\mathrm{s}}^{\mathsf{T}})^{l+1} W_{\mathrm{db}} P_{\mathrm{s}}^l.$$

Proof. It follows from the definition of p-peer hyper-cuboids (8) that

$$\begin{aligned} W_{\mathrm{hc}}^{(\tau-l)} &= \underbrace{I \otimes \cdots \otimes I}_{l-1 \text{ times}} \otimes J_p \otimes I \cdots \otimes I && \text{(24a)} \\ &= (P_{\mathrm{s}}^{\mathsf{T}})^l (J_p \otimes I \otimes \cdots \otimes I \otimes I) P_{\mathrm{s}}^l && \text{(24b)} \\ &= (P_{\mathrm{s}}^{\mathsf{T}})^l P_{\mathrm{s}}^{\mathsf{T}} P_{\mathrm{s}} (J_p \otimes I \otimes \cdots \otimes I \otimes I) P_{\mathrm{s}}^l && \text{(24c)} \\ &= P_{\mathrm{s}}^{\mathsf{T}} (P_{\mathrm{s}}^{\mathsf{T}})^l W_{\mathrm{db}} P_{\mathrm{s}}^l. && \text{(24d)} \end{aligned}$$

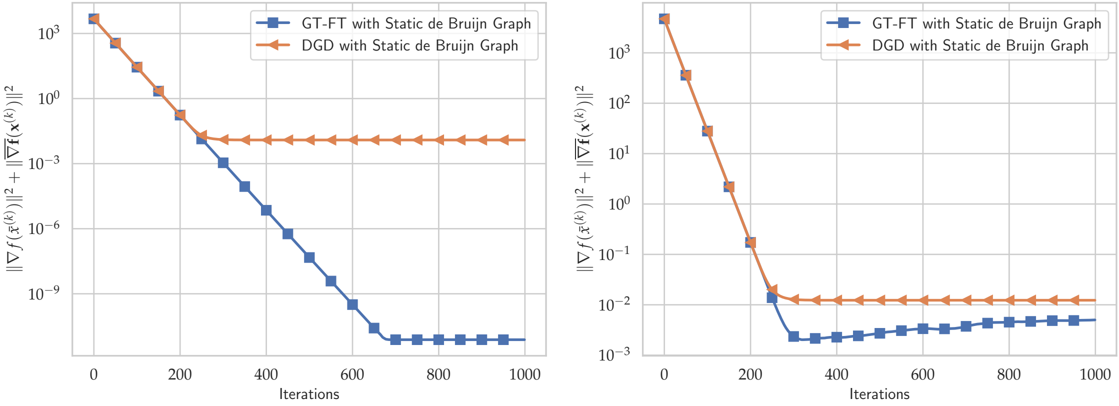
Step (24a) uses the definition of the hyper-cuboid presented in (11c). Step (24b) applies the property described in (22) l times to shift J_p to the beginning. Step (24c) uses the fact that $P_{\mathrm{s}}^{\mathsf{T}} P_{\mathrm{s}} = I$. Step (24d) uses the definition of de Bruijn graphs in (21). □

Proposition A.3 implies that de Bruijn graphs and hyper-cuboids are permutation equivalent. Moreover, note that the similarity transformation $(P_{\mathrm{s}}^{\mathsf{T}})^l W_{\mathrm{db}} P_{\mathrm{s}}^l$ is merely relabeling the nodes.

A.3 Numerical Experiments

We analyze the performance of de Bruijn graphs in decentralized optimization algorithms. We consider a de Bruijn graph of size $n = 64$ and consider its performance in GT-FT and DGD. We use the stepsizes $\alpha = 10^{-4}$,

$\eta = 1$ in GT-FT and the stepsize $\alpha = 10^{-4}$ in DGD. We again consider the case where agents have access to true gradients and the case where agents only have access to stochastic gradients. The stochastic gradients are formed as in the one-peer exponential experiment, and we set $\sigma^2 = 10^{-4}$. The simulation results are presented in Figure 7, and the performance is similar to the experiments performed in Section 6.2.



(a) Agents use true gradients.

(b) Agents use stochastic gradients.

Figure 7: Results of using de Bruijn graphs in GT-FT and DGD.

B Supplementary Materials for Section 5

This section includes the missing proofs from Section 5.

B.1 Transformation of Algorithm 1

The following list of notations will be used in the transformations:

$$\bar{\mathbf{x}}^k \triangleq \mathbb{1}_n \otimes \bar{x}^k, \quad \bar{x}^k \triangleq \frac{1}{n}\sum_{i=1}^n x_i^k, \tag{25a}$$

$$\hat{\mathbf{x}}^k \triangleq \mathbf{x}^k - \bar{\mathbf{x}}^k, \tag{25b}$$

$$\overline{\nabla f}(\mathbf{x}^k) \triangleq \frac{1}{n}\sum_{i=1}^n \nabla f_i(x_i^k), \tag{25c}$$

$$\overline{\nabla \mathbf{f}}(\mathbf{x}^k) \triangleq \overline{\nabla f}(\mathbf{x}^k) \otimes I_d, \tag{25d}$$

$$\mathbf{s}^k \triangleq \nabla \mathbf{F}(\mathbf{x}^k, \boldsymbol{\xi}^k) - \nabla \mathbf{f}(\mathbf{x}^k), \tag{25e}$$

$$\bar{s}^k \triangleq \frac{1}{n}\sum_{i=1}^n (\nabla F_i(\mathbf{x}_i^k; \xi_i^k) - \nabla f_i(\mathbf{x}_i^k)), \tag{25f}$$

$$\widehat{\mathbf{W}}^{(k)} \triangleq \mathbf{W}^{(k)} - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\mathsf{T} \otimes I_d, \tag{25g}$$

$$\hat{\mathbf{I}} \triangleq \mathbf{I} - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\mathsf{T} \otimes I_d. \tag{25h}$$

With the introduced notations, Algorithm 1 can be rewritten as

$$\mathbf{x}^{(k+1)} = \mathbf{W}^{(k)}\left(\mathbf{x}^{(k)} - \alpha\eta\nabla\mathbf{F}(\mathbf{x}^{(k)}; \boldsymbol{\xi}^{(k)}) - \alpha\mathbf{y}^{(k)}\right) \tag{26a}$$

$$\mathbf{y}^{(k+1)} = \mathbf{W}^{(k)}\mathbf{y}^{(k)} - \eta(\mathbf{I} - \mathbf{W}^{(k)})\nabla\mathbf{F}(\mathbf{x}^{(k)}, \boldsymbol{\xi}^{(k)}), \tag{26b}$$

where $\mathbf{y}^{(0)} = \mathbf{0}$ and $\mathbf{y}^{(k)} \triangleq \mathbf{g}^{(k)} - \eta\nabla\mathbf{F}(\mathbf{x}^{(k)}, \boldsymbol{\xi}^{(k)})$. It follows from the definition of $\mathbf{y}^{(k)}$ and the doubly-stochasticity of $W^{(k)}$ that $(\mathbb{1}^\intercal \otimes I_d)\mathbf{y}^{(k+1)} = (\mathbb{1}^\intercal \otimes I_d)\mathbf{y}^{(k)}$. Then the initialization $\mathbf{y}^{(0)} = \mathbf{0}$ guarantees that $(\mathbb{1}^\intercal \otimes I_d)\mathbf{y}^{(k)} = 0$ for all $k = 0, 1, \ldots$, and multiplying (26a) by $(\mathbb{1}^\intercal \otimes I_d)$ on both sides yields

$$\bar{x}^{(k+1)} = \bar{x}^{(k)} - \alpha\eta(\overline{\nabla f}(\mathbf{x}^{(k)}) + \bar{s}^{(k)}), \tag{27}$$

where we use the definitions (25c) and (25f). This recursion reveals that for TV-GT, the evolution of the average of agents' parameters is the same as the Centralized Gradient Descent. Thus, by demonstrating that the agents' parameters converge to the average of all agents' parameters, we can establish the convergence of TV-GT. We now introduce a second transformation

$$\mathbf{z}^{(k)} \triangleq \mathbf{y}^{(k)} + \eta\nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)}), \text{ where } \bar{\mathbf{z}}^{(k)} \triangleq \mathbb{1}_n \otimes \bar{z}^{(k)}, \;\; \bar{z}^{(k)} \triangleq \frac{1}{n}\sum_{i=1^n} z_i^{(k)}, \;\; \hat{\mathbf{z}}^{(k)} \triangleq \mathbf{z}^{(k)} - \bar{\mathbf{z}}^{(k)}.$$

Substituting $\mathbf{z}^{(k)}$ in (26) and eliminating $\mathbf{y}^{(k)}$ gives

$$\mathbf{x}^{k+1} = \mathbf{W}^{(k)}\left(\mathbf{x}^k - \alpha\mathbf{z}^k - \alpha\eta(\nabla\mathbf{f}(\mathbf{x}^k) - \nabla\mathbf{f}(\bar{\mathbf{x}}^k) + \mathbf{s}^k)\right) \tag{28a}$$

$$\mathbf{z}^{k+1} = \mathbf{W}^{(k)}(\mathbf{z}^k - \eta\nabla\mathbf{f}(\bar{\mathbf{x}}^k)) - \eta(\mathbf{I} - \mathbf{W}^{(k)})\nabla\mathbf{F}(\mathbf{x}^k, \boldsymbol{\xi}^k) + \eta\nabla\mathbf{f}(\bar{\mathbf{x}}^{k+1}). \tag{28b}$$

Adding and subtracting $\eta\nabla\mathbf{f}(\bar{\mathbf{x}}^k)$ on the right hand side of (28b) yields

$$\mathbf{x}^{k+1} = \mathbf{W}^{(k)}\left(\mathbf{x}^k - \alpha\mathbf{z}^k - \alpha\eta(\nabla\mathbf{f}(\mathbf{x}^k) - \nabla\mathbf{f}(\bar{\mathbf{x}}^k) + \mathbf{s}^k)\right) \tag{29a}$$

$$\begin{aligned}\mathbf{z}^{k+1} = {} & \mathbf{W}^{(k)}\mathbf{z}^k - \eta(\mathbf{I} - \mathbf{W}^{(k)})(\nabla\mathbf{f}(\mathbf{x}^k) - \nabla\mathbf{f}(\bar{\mathbf{x}}^k) + \mathbf{s}^k) \\ & + \eta(\nabla\mathbf{f}(\bar{\mathbf{x}}^{k+1}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^k)).\end{aligned} \tag{29b}$$

By multiplying both sides of the equations in (29) by $\widehat{\mathbf{I}}$ on the left we can then find

$$\begin{aligned}\hat{\mathbf{x}}^{(k+1)} &= \widehat{\mathbf{W}}^{(k)}\left(\hat{\mathbf{x}}^{(k)} - \alpha\hat{\mathbf{z}}^{(k)} - \alpha\eta(\nabla\mathbf{f}(\mathbf{x}^{(k)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)}) + \mathbf{s}^{(k)})\right) \\ \hat{\mathbf{z}}^{(k+1)} &= \widehat{\mathbf{W}}^{(k)}\hat{\mathbf{z}}^{(k)} - \eta(\mathbf{I} - \mathbf{W}^{(k)})(\nabla\mathbf{f}(\mathbf{x}^{(k)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)}) + \mathbf{s}^{(k)}) + \eta\widehat{\mathbf{I}}(\nabla\mathbf{f}(\bar{\mathbf{x}}^{(k+1)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)})).\end{aligned}$$

We can rewrite the above more compactly into matrix form:

$$\begin{aligned}\begin{bmatrix}\hat{\mathbf{x}}^{(k+1)} \\ \hat{\mathbf{z}}^{(k+1)}\end{bmatrix} = {} & \begin{bmatrix}\widehat{\mathbf{W}}^{(k)} & -\alpha\widehat{\mathbf{W}}^{(k)} \\ \mathbf{0} & \widehat{\mathbf{W}}^{(k)}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{x}}^{(k)} \\ \hat{\mathbf{z}}^{(k)}\end{bmatrix} \\ & - \eta\begin{bmatrix}\alpha\widehat{\mathbf{W}}^{(k)}(\nabla\mathbf{f}(\mathbf{x}^k) - \nabla\mathbf{f}(\bar{\mathbf{x}}^k) + \mathbf{s}^k) \\ (\mathbf{I} - \mathbf{W}^{(k)})(\nabla\mathbf{f}(\mathbf{x}^k) - \nabla\mathbf{f}(\bar{\mathbf{x}}^k) + \mathbf{s}^k) - \widehat{\mathbf{I}}(\nabla\mathbf{f}(\bar{\mathbf{x}}^{k+1}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^k))\end{bmatrix}.\end{aligned}$$

This result provides a model of the evolution of the consensus error $\|\hat{\mathbf{e}}^{(k)}\|^2 = \|\hat{\mathbf{x}}^k\|^2 + \|\hat{\mathbf{z}}^k\|^2$ with the number of iterations. We can then obtain the recursion (17), which is reiterated below:

$$\bar{x}^{(k+1)} = \bar{x}^{(k)} - \alpha\eta(\overline{\nabla f}(\mathbf{x}^{(k)}) + \bar{s}^{(k)}) \tag{30a}$$

$$\hat{\mathbf{e}}^{(k+1)} = \mathbf{G}^{(k)}\hat{\mathbf{e}}^{(k)} - \eta(\mathbf{h}^{(k+1)} + \mathbf{w}^{(k)}), \tag{30b}$$

where

$$\hat{\mathbf{e}}^{(k)} = \begin{bmatrix}\hat{\mathbf{x}}^{(k)} \\ \hat{\mathbf{z}}^{(k)}\end{bmatrix}, \tag{31a}$$

$$\mathbf{h}^{(k+1)} = \begin{bmatrix}\alpha\widehat{\mathbf{W}}^{(k)}(\nabla\mathbf{f}(\mathbf{x}^{(k)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)})) \\ (\mathbf{I} - \mathbf{W}^{(k)})(\nabla\mathbf{f}(\mathbf{x}^k) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)})) - \widehat{\mathbf{I}}(\nabla\mathbf{f}(\bar{\mathbf{x}}^{(k+1)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)}))\end{bmatrix}, \tag{31b}$$

$$\mathbf{w}^{(k)} = \begin{bmatrix} \alpha \hat{\mathbf{W}}^{(k)} \mathbf{s}^{(k)} \\ (\mathbf{I} - \mathbf{W}^{(k)}) \mathbf{s}^{(k)} \end{bmatrix}, \tag{31c}$$

$$\mathbf{G}^{(k)} = \begin{bmatrix} \widehat{\mathbf{W}}^{(k)} & -\alpha \widehat{\mathbf{W}}^{(k)} \\ \mathbf{0} & \widehat{\mathbf{W}}^{(k)} \end{bmatrix}. \tag{31d}$$

B.2 Properties of the Consensus Error Matrix

Recall that we define $\coprod_{k=i+j}^{i} \mathbf{G}^{(k)} \triangleq \mathbf{G}^{(i+j)} \mathbf{G}^{(i+j-1)} \dots \mathbf{G}^{(i)}$.

Proof of Lemma 5.1. From the definition of $\mathbf{G}^{(k)}$ (31d), it follows that

$$\begin{aligned} \left\| \coprod_{i=k-1}^{m\tau} \mathbf{G}^{(i)} \right\|_2 &= \left\| \begin{bmatrix} \coprod_{i=k-1}^{m\tau} \widehat{\mathbf{W}}^{(i)} & -\alpha(k-1-m\tau) \coprod_{i=k-1}^{m\tau} \widehat{\mathbf{W}}^{(i)} \\ \mathbf{0} & \coprod_{i=k-1}^{m\tau} \widehat{\mathbf{W}}^{(i)} \end{bmatrix} \right\|_2 \\ &\leq \left\| \coprod_{i=k-1}^{m\tau} \widehat{\mathbf{W}}^{(i)} \right\|_2 \left\| \begin{bmatrix} \mathbf{I} & -\alpha(k-1-m\tau)\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \right\|_2 \\ &= 0. \end{aligned}$$

In the second, we use the sub-multiplicative property of matrix norms; and in the last line, we use the exact averaging property of the one-peer exponential graph. □

Proof of Lemma 5.2. From the definition of $\mathbf{G}^{(k)}$ (31d), it follows that

$$\begin{aligned} \left\| \coprod_{i=k-1}^{j+1} \mathbf{G}^{(i)} \right\|_2 &= \left\| \begin{bmatrix} \coprod_{i=k-1}^{j+1} \widehat{\mathbf{W}}^{(i)} & -\alpha[(k-1)-(j+1)] \coprod_{i=k-1}^{j+1} \widehat{\mathbf{W}}^{(i)} \\ \mathbf{0} & \coprod_{i=k-1}^{j+1} \widehat{\mathbf{W}}^{(i)} \end{bmatrix} \right\|_2 \\ &\leq \left\| \coprod_{i=k-1}^{j+1} \widehat{\mathbf{W}}^{(i)} \right\|_2 \left\| \begin{bmatrix} \mathbf{I} & -\alpha[(k-1)-(j+1)]\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \right\|_2 && \text{(32a)} \\ &\leq \left\| \begin{bmatrix} \mathbf{I} & -\alpha[(k-1)-(j+1)]\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \right\|_2 && \text{(32b)} \\ &= \left\| \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & -\alpha[(k-1)-(j+1)]\mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right\|_2 && \text{(32c)} \\ &\leq 1 + \alpha(\tau - 1). && \text{(32d)} \end{aligned}$$

In (32a), we use the sub-multiplicative property of matrix norms. (32b) follows from the bound $\|\widehat{\mathbf{W}}^{(k)}\|_2 \leq 1$, which can be derived from the doubly-stochastic property of $\mathbf{W}^{(k)}$. In (32c), we write the matrix as the sum of two matrices, and then in (32d), we use the sub-additive property of matrix norms and the fact that $(k-1)-(j+1) < \tau$. □

B.3 Proof of Descent Inequality

Proof of Lemma 5.3. The L-smoothness of f in Assumption 5.1 implies that

$$f(y) \leq f(z) + \langle \nabla f(z), y - z \rangle + \frac{L}{2}\|y - z\|^2,$$

for all $y, z \in \operatorname{int} \operatorname{dom} f$. Setting $y = \bar{x}^{(k+1)}$, $z = \bar{x}^{(k)}$, and using the recursion (27) yields

$$f(\bar{x}^{(k+1)}) \leq f(\bar{x}^{(k)}) - \alpha\eta \langle \nabla f(\bar{x}^{(k)}), \overline{\nabla f}(\mathbf{x}^{(k)}) + \bar{s}^{(k)} \rangle + \frac{\alpha^2\eta^2 L}{2} \|\overline{\nabla f}(\mathbf{x}^{(k)}) + \bar{s}^{(k)}\|^2.$$

Taking the conditional expectation on the filtration $\mathcal{F}^{(k)}$ gives

$$\begin{aligned}\mathbb{E}[f(\bar{x}^{(k+1)}) \mid \mathcal{F}^{(k)}] \le f(\bar{x}^k) &- \alpha\eta\mathbb{E}[\langle \nabla f(\bar{x}^{(k)}), \overline{\nabla f}(\mathbf{x}^{(k)})\bar{s}^{(k)}\rangle \mid \mathcal{F}^{(k)}] \\ &+ \frac{\alpha^2\eta^2 L}{2}\mathbb{E}[\|\overline{\nabla f}(\mathbf{x}^{(k)}) + \bar{s}^{(k)}\|^2 \mid \mathcal{F}^{(k)}].\end{aligned} \tag{33}$$

The unbiasedness of the stochastic gradient estimator in Assumption 5.2 implies

$$\mathbb{E}[\langle \nabla f(\bar{x}^{(k)}), \bar{s}^{(k)}\rangle \mid \mathcal{F}^{(k)}] = \langle \nabla f(\bar{x}^{(k)}), \mathbb{E}[\bar{s}^{(k)} \mid \mathcal{F}^{(k)}]\rangle = 0. \tag{34}$$

Similarly, one has from Assumption 5.2 that

$$\begin{aligned}&\mathbb{E}[\|\overline{\nabla f}(\mathbf{x}^{(k)}) + \bar{s}^{(k)}\|^2 \mid \mathcal{F}^{(k)}] \\ &= \mathbb{E}[\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 \mid \mathcal{F}^{(k)}] + \mathbb{E}[\|\bar{s}^{(k)}\|^2 \mid \mathcal{F}^{(k)}] + \mathbb{E}[\langle \overline{\nabla f}(\mathbf{x}^{(k)}), \bar{s}^{(k)}\rangle \mid \mathcal{F}^{(k)}] \\ &= \mathbb{E}[\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 \mid \mathcal{F}^{(k)}] + \mathbb{E}[\|\bar{s}^{(k)}\|^2 \mid \mathcal{F}^{(k)}] + \mathbb{E}[\langle \overline{\nabla f}(\mathbf{x}^{(k)}), \bar{s}^{(k)}\rangle \mid \mathcal{F}^{(k)}] \\ &= \mathbb{E}[\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 \mid \mathcal{F}^{(k)}] + \mathbb{E}[\|\bar{s}^{(k)}\|^2 \mid \mathcal{F}^{(k)}] + \langle \overline{\nabla f}(\mathbf{x}^{(k)}), \mathbb{E}[\bar{s}^{(k)} \mid \mathcal{F}^{(k)}]\rangle \\ &= \mathbb{E}[\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 \mid \mathcal{F}^{(k)}] + \tfrac{\sigma^2}{n}.\end{aligned} \tag{35}$$

Substituting (34) and (35) into (33) gives

$$\begin{aligned}\mathbb{E}[f(\bar{x}^{(k+1)}) \mid \mathcal{F}^{(k)}] \le f(\bar{x}^{(k)}) &- \alpha\eta\mathbb{E}[\langle \nabla f(\bar{x}^{(k)}), \overline{\nabla f}(\mathbf{x}^{(k)})\rangle \mid \mathcal{F}^{(k)}] \\ &+ \frac{\alpha^2\eta^2 L}{2}\mathbb{E}[\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 \mid \mathcal{F}^{(k)}] + \frac{\alpha^2\eta^2 L\sigma^2}{2n}.\end{aligned} \tag{36}$$

It follows from the identity $2\langle a, b\rangle = \|a\|^2 + \|b\|^2 - \|a-b\|^2$ that

$$\begin{align*}&-\langle \nabla f(\bar{x}^{(k)}), \overline{\nabla f}(\mathbf{x}^{(k)})\rangle \\ &= -\frac{1}{2}\|\nabla f(\bar{x}^{(k)})\|^2 - \frac{1}{2}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 + \frac{1}{2}\|\nabla f(\bar{x}^{(k)}) - \overline{\nabla f}(\mathbf{x}^{(k)})\|^2 \\ &= -\frac{1}{2}\|\nabla f(\bar{x}^{(k)})\|^2 - \frac{1}{2}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 + \frac{1}{2}\left\|\frac{1}{n}\sum_{i=1}^n \nabla f_i(\bar{x}^{(k)}) - \frac{1}{n}\sum_{i=1}^n \nabla f_i(x_i^{(k)})\right\|^2 \tag{37} \\ &\le -\frac{1}{2}\|\nabla f(\bar{x}^{(k)})\|^2 - \frac{1}{2}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 + \frac{1}{2n}\sum_{i=1}^n \left\|\nabla f_i(\bar{x}^{(k)}) - \nabla f_i(x_i^{(k)})\right\|^2 \tag{38} \\ &\le -\frac{1}{2}\|\nabla f(\bar{x}^{(k)})\|^2 - \frac{1}{2}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 + \frac{L^2}{2n}\sum_{i=1}^n \left\|\bar{x}^{(k)} - x_i^{(k)}\right\|^2 \tag{39} \\ &\le -\frac{1}{2}\|\nabla f(\bar{x}^{(k)})\|^2 - \frac{1}{2}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 + \frac{L^2}{2n}\|\hat{\mathbf{e}}^{(k)}\|^2. \tag{40}\end{align*}$$

In (37), we apply the definitions in (1) and (25c), and (38) uses Jensen's inequality. In (39), we use Assumption 5.1, and finally in (40), we use the definition (31a) and hence the inequality $\|\hat{\mathbf{x}}^{(k)}\|^2 \le \|\hat{\mathbf{e}}^{(k)}\|^2$.

Substituting (40) into (36) and taking total expectation, one finds that

$$\begin{aligned}\mathbb{E}\, f(\bar{x}^{(k+1)}) \le \mathbb{E}\, f(\bar{x}^{(k)}) &- \frac{\alpha\eta(1-\alpha\eta L)}{2}\,\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 - \frac{\alpha\eta}{2}\,\mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2 \\ &+ \frac{\alpha\eta L^2}{2n}\,\mathbb{E}\|\hat{\mathbf{e}}^{(k)}\|^2 + \frac{\alpha^2\eta^2 L\sigma^2}{2n}.\end{aligned}$$

It then follows from the stepsize condition $\eta \le \frac{1}{2L}$ that

$$\mathbb{E}\, f(\bar{x}^{(k+1)}) \le \mathbb{E}\, f(\bar{x}^{(k)}) - \frac{\alpha\eta}{4}\,\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 - \frac{\alpha\eta}{2}\,\mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2 + \frac{\alpha\eta L^2}{2n}\,\mathbb{E}\|\hat{\mathbf{e}}^{(k)}\|^2 + \frac{\alpha^2\eta^2 L\sigma^2}{2n}.$$

Forming a looser bound by replacing $-\frac{\alpha\eta}{2}\,\mathbb{E}\|\nabla f(\bar{x}^k)\|^2$ with $-\frac{\alpha\eta}{4}\,\mathbb{E}\|\nabla f(\bar{x}^k)\|^2$, rearranging, and then adding and subtracting f^* on the right hand side, one finally obtains that

$$\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 + \mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2 \leq \frac{4}{\alpha\eta}\left(\mathbb{E}\,\tilde{f}(\bar{x}^{(k)}) - \mathbb{E}\,\tilde{f}(\bar{x}^{(k+1)})\right) + \frac{2L^2}{n}\,\mathbb{E}\|\hat{\mathbf{e}}^{(k)}\|^2 + \frac{2\alpha\eta L\sigma^2}{n},$$

where $\tilde{f} \triangleq f - f^*$. This completes the proof. □

B.4 Proof of Consensus Inequality

The proof of Lemma 5.4 needs the following two lemmas, which provide upper bounds for $\|\mathbf{w}^{(k)}\|^2$ and $\|\mathbf{h}^{(k)}\|^2$, respectively.

Lemma B.1. *For all $k \in \mathbb{N}$, the iterates $\mathbf{w}^{(k)}$ defined in (31c) satisfies*

$$\|\mathbf{w}^{(k)}\|^2 \leq 3\|\mathbf{s}^{(k)}\|^2.$$

Proof. It follows from the definition of $\mathbf{w}^{(k)}$ (31c) that

$$\begin{aligned}
\|\mathbf{w}^k\|^2 &= \left\|\begin{bmatrix} \alpha\widehat{\mathbf{W}}^{(k)}\mathbf{s}^{(k)} \\ (\mathbf{I}-\mathbf{W}^{(k)})\mathbf{s}^{(k)} \end{bmatrix}\right\|^2 \\
&= \|\alpha\widehat{\mathbf{W}}^{(k)}(\mathbf{s}^{(k)})\|^2 + \|(\mathbf{I}-\mathbf{W}^{(k)})\mathbf{s}^{(k)}\|^2 \\
&\leq 3\|\mathbf{s}^{(k)}\|^2.
\end{aligned}$$

In the second line, we expand the squared norm term; and the last line follows from the fact that $\alpha \leq 1$, $\|\widehat{\mathbf{W}}^{(k)}\|^2 \leq 1$, and $\|\mathbf{I}-\mathbf{W}^{(k)}\|^2 \leq 2$. □

Lemma B.2. *For all $k \in \mathbb{N}$, the iterates $\mathbf{h}^{(k)}$ defined in (31b) satisfies*

$$\|\mathbf{h}^{(k+1)}\|^2 \leq 5\|\nabla\mathbf{f}(\mathbf{x}^{(k)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)})\|^2 + 2\|\nabla\mathbf{f}(\bar{\mathbf{x}}^{(k+1)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)})\|^2.$$

Proof. Taking the squared norm of the definition of $\mathbf{h}^{(k)}$ (31b) yields

$$\begin{aligned}
\|\mathbf{h}^{k+1}\|^2 &= \left\|\begin{bmatrix} \alpha\widehat{\mathbf{W}}^{(k)}(\nabla\mathbf{f}(\mathbf{x}^{(k)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)})) \\ (\mathbf{I}-\mathbf{W}^{(k)})(\nabla\mathbf{f}(\mathbf{x}^{(k)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)})) - \hat{\mathbf{I}}(\nabla\mathbf{f}(\bar{\mathbf{x}}^{(k+1)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)})) \end{bmatrix}\right\|^2 \\
&= \|\alpha\widehat{\mathbf{W}}^{(k)}(\nabla\mathbf{f}(\mathbf{x}^{(k)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)}))\|^2 \\
&\qquad + \|(\mathbf{I}-\mathbf{W}^{(k)})(\nabla\mathbf{f}(\mathbf{x}^{(k)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)})) - \hat{\mathbf{I}}(\nabla\mathbf{f}(\bar{\mathbf{x}}^{(k+1)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)}))\|^2 && \text{(41a)} \\
&\leq \|\alpha\widehat{\mathbf{W}}^{(k)}(\nabla\mathbf{f}(\mathbf{x}^{(k)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)}))\|^2 \\
&\quad + 2\|(\mathbf{I}-\mathbf{W}^{(k)})(\nabla\mathbf{f}(\mathbf{x}^{(k)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)}))\|^2 + 2\|\widehat{\mathbf{I}}(\nabla\mathbf{f}(\bar{\mathbf{x}}^{(k+1)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)}))\|^2 && \text{(41b)} \\
&\leq \|(\nabla\mathbf{f}(\mathbf{x}^{(k)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)}))\|^2 + 4\|(\nabla\mathbf{f}(\mathbf{x}^{(k)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)}))\|^2 + 2\|(\nabla\mathbf{f}(\bar{\mathbf{x}}^{(k+1)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)}))\|^2 && \text{(41c)} \\
&\leq 5\|(\nabla\mathbf{f}(\mathbf{x}^{(k)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)}))\|^2 + 2\|(\nabla\mathbf{f}(\bar{\mathbf{x}}^{(k+1)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(k)}))\|^2. && \text{(41d)}
\end{aligned}$$

In (41a), we expand the squared norm term, and (41b) uses Jensen's inequality. In (41c), we use the fact that $\alpha \leq 1$, $\|\widehat{\mathbf{W}}^{(k)}\|^2 \leq 1$, $\|\widehat{\mathbf{I}}\|^2 \leq 1$, and $\|(\mathbf{I}-\mathbf{W}^{(k)})\|^2 \leq 2$. Finally in (41d), we group similar terms. □

Finally, the construction of the consensus recursion follows from the reformulation of (30b) and the exact averaging property of the one-peer exponential graphs.

Proof of Lemma 5.4. In order to exploit the periodic exact averaging property of the mixing matrices $\{W^{(k)}\}$, we will analyze the main recursion in terms of τ iterations, where $\tau = \ln(n)$ is assumed to be an integer. We also make the assumption that $k - m\tau \geq \tau$, where recall by definition $m \triangleq \lfloor k/\tau \rfloor - 1$. This choice of k guarantees that from iteration $m\tau$ to k, there must be at least one period.

We start recursion (30b) at iteration k and recurse back to $m\tau$. The iterations of $\hat{\mathbf{e}}^{(k)}$ then yield

$$\hat{\mathbf{e}}^k = \coprod_{i=k-1}^{m\tau} \mathbf{G}^{(i)} \hat{\mathbf{e}}^{m\tau} - \eta(\mathbf{h}^{(k)} + \mathbf{w}^{(k-1)}) - \eta \sum_{j=m\tau}^{k-2} \coprod_{i=k-1}^{j+1} \mathbf{G}^{(i)} (\mathbf{h}^{(j+1)} + \mathbf{w}^{(j)}).$$

The first term on the right-hand side is actually zero, due to the exact averaging property of $W^{(k)}$:

$$\begin{aligned}
\coprod_{i=k-1}^{m\tau} \mathbf{G}^{(i)} \hat{\mathbf{e}}^{m\tau} &= \begin{bmatrix} \coprod_{i=k-1}^{m\tau} \widehat{\mathbf{W}}^{(i)} & -\alpha(k-1-m\tau) \coprod_{i=k-1}^{m\tau} \widehat{\mathbf{W}}^{(i)} \\ \mathbf{0} & \coprod_{i=k-1}^{m\tau} \widehat{\mathbf{W}}^{(i)} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}^{(k)} \\ \hat{\mathbf{z}}^{(k)} \end{bmatrix} \\
&= \begin{bmatrix} \coprod_{i=k-1}^{m\tau} \widehat{\mathbf{W}}^{(i)} \hat{\mathbf{x}}^{(k)} - \alpha(k-1-m\tau) \coprod_{i=k-1}^{m\tau} \widehat{\mathbf{W}}^{(i)} \hat{\mathbf{z}}^{(k)} \\ \coprod_{i=k-1}^{m\tau} \widehat{\mathbf{W}}^{(i)} \hat{\mathbf{z}}^{(k)} \end{bmatrix} \\
&= \begin{bmatrix} \coprod_{i=k-1}^{m\tau} \widehat{\mathbf{W}}^{(i)} \hat{\mathbf{x}}^{(k)} - \alpha(k-1-m\tau) \coprod_{i=k-1}^{m\tau} \widehat{\mathbf{W}}^{(i)} \hat{\mathbf{z}}^{(k)} \\ \coprod_{i=k-1}^{m\tau} \widehat{\mathbf{W}}^{(i)} \hat{\mathbf{z}}^{(k)} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \qquad (42)
\end{aligned}$$

where the last equality uses Lemma 3.1. Consequently, one finds that

$$\hat{\mathbf{e}}^k = -\eta(\mathbf{h}^k + \mathbf{w}^{k-1}) - \eta \sum_{j=m\tau}^{k-2} \coprod_{i=k-1}^{j+1} \mathbf{G}^{(i)} (\mathbf{h}^{j+1} + \mathbf{w}^j).$$

Next, we take the squared norm and the conditional expectation on $\mathcal{F}^{(k)}$ on both sides:

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\mathbf{e}}^{(k)}\right\|^2 \middle| \mathcal{F}^{(k)}\right] &= \mathbb{E}\left[\left\| -\eta(\mathbf{h}^{(k)} + \mathbf{w}^{(k-1)}) - \eta \sum_{j=m\tau}^{k-2} \coprod_{i=k-1}^{j+1} \mathbf{G}^{(i)} (\mathbf{h}^{(j+1)} + \mathbf{w}^{(j)}) \right\|^2 \,\middle|\, \mathcal{F}^{(k)}\right] \\
&\leq 2\eta^2 \, \mathbb{E}\left[\left\| \mathbf{h}^{(k)} + \sum_{j=m\tau}^{k-2} \coprod_{i=k-1}^{j+1} \mathbf{G}^{(i)} (\mathbf{h}^{(j+1)}) \right\|^2 \,\middle|\, \mathcal{F}^{(k)}\right] \\
&\quad + 2\eta^2 \, \mathbb{E}\left[\left\| \mathbf{w}^{(k-1)} + \sum_{j=m\tau}^{k-2} \coprod_{i=k-1}^{j+1} \mathbf{G}^{(i)} (\mathbf{w}^{(j)}) \right\|^2 \,\middle|\, \mathcal{F}^{(k)}\right], \qquad (43)
\end{aligned}$$

where the second line uses Jensen's inequality. We then bound the two expectation terms on the right-hand of (43) one-by-one. We first handle the second expectation term which only contains the gradient noise.

$$\begin{aligned}
&2\eta^2 \, \mathbb{E}\left[\left\| \mathbf{w}^{(k-1)} + \sum_{j=m\tau}^{k-2} \coprod_{i=k-1}^{j+1} \mathbf{G}^{(i)} (\mathbf{w}^{(j)}) \right\|^2 \,\middle|\, \mathcal{F}^{(k)}\right] \\
&= 2\eta^2 \Big(\mathbb{E}\left[\left\| \mathbf{w}^{(k-1)} \right\|^2 \,\middle|\, \mathcal{F}^{(k)}\right] + \sum_{j=m\tau}^{k-2} \mathbb{E}\left[\left\| \coprod_{i=k-1}^{j+1} \mathbf{G}^{(i)} (\mathbf{w}^{(j)}) \right\|^2 \,\middle|\, \mathcal{F}^{(k)}\right] \Big) \\
&\leq 2\eta^2 (1 + \alpha(\tau - 1))^2 \Big(\sum_{j=m\tau}^{k-1} \mathbb{E}\left[\left\| \mathbf{w}^{(j)} \right\|^2 \,\middle|\, \mathcal{F}^{(k)}\right] \Big) \\
&\leq 6\tau\eta^2 (1 + \alpha(\tau - 1))^2 \sigma^2. \qquad (44)
\end{aligned}$$

The second line follows from the independence of the gradient noise, and the third line uses Lemma 5.2 and the sub-multiplicative property of norms. In the last line, we use Lemma B.1 and the bounded variance property from Assumption 5.2.

We then handle the first expectation term on the right-hand side of (43).

$$\begin{aligned}
&2\eta^2\,\mathbb{E}\left[\left\|\mathbf{h}^{(k)} + \sum_{j=m\tau}^{k-2}\prod_{i=k-1}^{j+1}\mathbf{G}^{(i)}(\mathbf{h}^{(j+1)})\right\|^2 \;\Big|\; \mathcal{F}^{(k)}\right] \\
&\le 2\tau\eta^2\Big(\mathbb{E}\left[\|\mathbf{h}^{(k)}\|^2 \mid \mathcal{F}^{(k)}\right] + \sum_{j=m\tau}^{k-2}\mathbb{E}\left[\left\|\prod_{i=k-1}^{j+1}\mathbf{G}^{(i)}(\mathbf{h}^{(j+1)})\right\|^2 \;\Big|\; \mathcal{F}^{(k)}\right]\Big) \\
&\le 2\tau\eta^2(1+\alpha(\tau-1))^2\sum_{j=m\tau}^{k-1}\mathbb{E}\left[\|\mathbf{h}^{(j+1)}\|^2 \mid \mathcal{F}^{(k)}\right]. \qquad (45)
\end{aligned}$$

The second line uses Jensen's inequality. (Recall that the independence of random variables no longer holds.) The last line uses Lemma 5.2 and the sub-multiplicative property of norms. We then bound the term $\mathbb{E}[\|\mathbf{h}^{(j+1)}\|^2 \mid \mathcal{F}^{(k)}]$ for $j = m\tau, \ldots, k-1$ below.

$$\begin{aligned}
&\mathbb{E}\left[\|\mathbf{h}^{(j+1)}\|^2 \mid \mathcal{F}^{(k)}\right] \\
&\le 5\,\mathbb{E}[\|\nabla\mathbf{f}(\mathbf{x}^{(j)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(j)})\|^2 \mid \mathcal{F}^{(k)}] + 2\,\mathbb{E}[\|\nabla\mathbf{f}(\bar{\mathbf{x}}^{(j+1)}) - \nabla\mathbf{f}(\bar{\mathbf{x}}^{(j)})\|^2 \mid \mathcal{F}^{(k)}] && \text{(46a)} \\
&\le L^2\big(5\,\mathbb{E}[\|\mathbf{x}^{(j)} - \bar{\mathbf{x}}^{(j)}\|^2 \mid \mathcal{F}^{(k)}] + 2n\,\mathbb{E}[\|\bar{x}^{(j+1)} - \bar{x}^{(j)}\|^2 \mid \mathcal{F}^{(k)}]\big) && \text{(46b)} \\
&\le L^2\big(5\,\mathbb{E}[\|\mathbf{x}^{(j)} - \bar{\mathbf{x}}^{(j)}\|^2 \mid \mathcal{F}^{(k)}] + 2n\alpha^2\eta^2\,\mathbb{E}[\|\overline{\nabla f}(\mathbf{x}^{(j)} + \bar{s}^{(j)})\|^2 \mid \mathcal{F}^{(k)}]\big) && \text{(46c)} \\
&\le L^2\big(5\,\mathbb{E}[\|\hat{\mathbf{e}}^{(j)}\|^2 \mid \mathcal{F}^{(k)}] + 4n\alpha^2\eta^2\big(\mathbb{E}[\|\overline{\nabla f}(\mathbf{x}^{(j)})\|^2 \mid \mathcal{F}^{(k)}] + \mathbb{E}[\|\bar{s}^{(j)}\|^2 \mid \mathcal{F}^{(k)}]\big)\big) && \text{(46d)} \\
&\le L^2\big(5\,\mathbb{E}[\|\hat{\mathbf{e}}^{(j)}\|^2 \mid \mathcal{F}^{(k)}] + 4n\alpha^2\eta^2\,\mathbb{E}[\|\overline{\nabla f}(\mathbf{x}^{(j)})\|^2 \mid \mathcal{F}^{(k)}] + 4n\alpha^2\eta^2\sigma^2\big). && \text{(46e)}
\end{aligned}$$

In (46a), we use Lemma B.2, and in (46b), Assumption 5.1. In (46c), we use the update rule (30a). In (46d), we use the fact $\|\bar{\mathbf{x}}^{(j)}\|^2 \le \|\hat{\mathbf{e}}^{(j)}\|^2$ for all $j \in \mathbb{N}$, and also apply Jensen's inequality. In (46e), we use the bounded variance property from Assumption 5.2.

Then, plugging (44), (45), and (46e) into (43) yields

$$\begin{aligned}
\mathbb{E}[\|\hat{\mathbf{e}}^{(k)}\|^2 \mid \mathcal{F}^{(k)}] \le{}& 10\tau\eta^2L^2(1+\alpha(\tau-1))^2\sum_{j=m\tau}^{k-1}\mathbb{E}[\|\hat{\mathbf{e}}^{(j)}\|^2 \mid \mathcal{F}^{(k)}] \\
&+ 8n\tau\alpha^2\eta^4L^2(1+\alpha(\tau-1))^2\sum_{j=m\tau}^{k-1}\mathbb{E}[\|\overline{\nabla f}(\mathbf{x}^{(j)})\|^2 \mid \mathcal{F}^{(k)}] \\
&+ \big(6\tau\eta^2(1+\alpha(\tau-1))^2 + 8n\tau^2\alpha^2\eta^4L^2(1+\alpha(\tau-1))^2\ \big)\sigma^2. \qquad (47)
\end{aligned}$$

Recall that the above analysis uses the exact averaging property (in (42)), and thus the iteration counter k must satisfy $k \ge \tau$. Summing up (47) over iteration k from τ to T ($T \ge \tau$) yields

$$\begin{aligned}
\sum_{k=\tau}^{T}\mathbb{E}[\|\hat{\mathbf{e}}^{(k)}\|^2 \mid \mathcal{F}^{(k)}] \le{}& 10\tau\eta^2L^2(1+\alpha(\tau-1))^2\sum_{k=\tau}^{T}\sum_{j=m\tau}^{k-1}\mathbb{E}[\|\hat{\mathbf{e}}^{(j)}\|^2 \mid \mathcal{F}^{(k)}] \\
&+ 8n\tau\alpha^2\eta^4L^2(1+\alpha(\tau-1))^2\sum_{k=\tau}^{T}\sum_{j=m\tau}^{k-1}\mathbb{E}[\|\overline{\nabla f}(\mathbf{x}^{(j)})\|^2 \mid \mathcal{F}^{(k)}] \\
&+ (T-\tau+1)\big(6\tau\eta^2(1+\alpha(\tau-1))^2 + 8n\tau^2\alpha^2\eta^4L^2(1+\alpha(\tau-1))^2\big)\sigma^2.
\end{aligned}$$

Adding $\sum_{k=0}^{\tau-1}\mathbb{E}[\|\hat{\mathbf{e}}^{(k)}\|^2 \mid \mathcal{F}^{(k)}]$ and dividing $T+1$ on both sides yields

$$\frac{1}{T+1}\sum_{k=0}^{T}\mathbb{E}[\|\hat{\mathbf{e}}^{(k)}\|^2 \mid \mathcal{F}^{(k)}] \le \frac{10\tau\eta^2L^2(1+\alpha(\tau-1))^2}{T+1}\sum_{k=\tau}^{T}\sum_{j=m\tau}^{k-1}\mathbb{E}[\|\hat{\mathbf{e}}^{(j)}\|^2 \mid \mathcal{F}^{(k)}]$$

$$
\begin{aligned}
&+ \frac{1}{T+1}\sum_{k=0}^{\tau-1}\mathbb{E}[\|\hat{\mathbf{e}}^{(k)}\|^2 \mid \mathcal{F}^{(k)}] \\
&+ \frac{8n\tau\alpha^2\eta^4L^2(1+\alpha(\tau-1))^2}{T+1}\sum_{k=\tau}^{T}\sum_{j=m\tau}^{k-1}\mathbb{E}[\|\overline{\nabla f}(\mathbf{x}^{(j)})\|^2 \mid \mathcal{F}^{(k)}] \\
&+ \frac{T-\tau+1}{T+1}\big(6\tau\eta^2(1+\alpha(\tau-1))^2 + 8n\tau^2\alpha^2\eta^4L^2(1+\alpha(\tau-1))^2\ \big)\sigma^2. \qquad (48)
\end{aligned}
$$

Observe that for any constant $T \in \mathbb{N}$ and for any sequence $\{\psi_j\} \subset \mathbb{R}$, there exists a nonnegative sequence $\{\beta_j\} \subset \mathbb{R}_{\geq 0}$ such that $\beta_j \leq 2\tau$ for $j = 0, 1, \ldots, T$ and

$$
\sum_{k=\tau}^{T}\sum_{j=m\tau}^{k-1}\psi_j = \sum_{k=0}^{T}\beta_k\psi_k \leq 2\tau\sum_{k=0}^{T}\psi_k.
$$

Applying this fact to (48) gives

$$
\begin{aligned}
\frac{1}{T+1}\sum_{k=0}^{T}\mathbb{E}[\|\hat{\mathbf{e}}^{(k)}\|^2 \mid \mathcal{F}^{(k)}] \leq{}& \frac{20\tau\eta^2L^2(1+\alpha(\tau-1))^2}{T+1}\sum_{k=0}^{T}\mathbb{E}[\|\hat{\mathbf{e}}^{(k)}\|^2 \mid \mathcal{F}^{(k)}] + \frac{1}{T+1}\sum_{k=0}^{\tau-1}\mathbb{E}[\|\hat{\mathbf{e}}^{(k)}\|^2 \mid \mathcal{F}^{(k)}] \\
&+ \frac{16n\tau\alpha^2\eta^4L^2(1+\alpha(\tau-1))^2}{T+1}\sum_{k=0}^{T}\mathbb{E}[\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 \mid \mathcal{F}^{(k)}] \\
&+ \frac{T-\tau+1}{T+1}\big(6\tau\eta^2(1+\alpha(\tau-1))^2 + 8n\tau^2\alpha^2\eta^4L^2(1+\alpha(\tau-1))^2\big)\sigma^2,
\end{aligned}
$$

or equivalently,

$$
\begin{aligned}
&\frac{1-20\tau\eta^2L^2(1+\alpha(\tau-1))^2}{T+1}\,\mathbb{E}[\|\hat{\mathbf{e}}^{(k)}\|^2 \mid \mathcal{F}^{(k)}] \\
&\leq \frac{1}{T+1}\sum_{k=0}^{\tau-1}\mathbb{E}[\|\hat{\mathbf{e}}^{(k)}\|^2 \mid \mathcal{F}^{(k)}] + \frac{16n\tau\alpha^2\eta^4L^2(1+\alpha(\tau-1))^2}{T+1}\sum_{k=0}^{T}\mathbb{E}[\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 \mid \mathcal{F}^{(k)}] \\
&\quad + \frac{T-\tau+1}{T+1}\big(6\tau\eta^2(1+\alpha(\tau-1))^2 + 8n\tau^2\alpha^2\eta^4L^2(1+\alpha(\tau-1))^2\big)\sigma^2. \qquad (49)
\end{aligned}
$$

The stepsize conditions $\alpha \leq \frac{1}{\tau}$ and $\eta \leq \frac{1}{4\sqrt{5}\tau L}$ implies that

$$
(1+\alpha(\tau-1))^2 \leq 4, \qquad 1-20\tau\eta^2L^2(1+\alpha(\tau-1))^2 \geq 1-80\tau\eta^2L^2 \geq \frac{1}{2}.
$$

Substituting it into (49) gives

$$
\begin{aligned}
\frac{1}{T+1}\,\mathbb{E}[\|\hat{\mathbf{e}}^{(k)}\|^2 \mid \mathcal{F}^{(k)}] \leq{}& \frac{2}{T+1}\sum_{k=0}^{\tau-1}\mathbb{E}[\|\hat{\mathbf{e}}^{(k)}\|^2 \mid \mathcal{F}^{(k)}] + \frac{64n\eta^4L^2}{T+1}\sum_{k=0}^{T}\mathbb{E}[\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 \mid \mathcal{F}^{(k)}] \\
&+ \Big(1-\frac{\tau}{T+1}\Big)\big(48\tau\eta^2(1+\alpha(\tau-1))^2 + 64n\tau^2\alpha^2\eta^4L^2(1+\alpha(\tau-1))^2\big)\sigma^2.
\end{aligned}
$$

and the desired result is attained by taking the total expectation and relaxing $(1-\frac{\tau}{T+1})$ to 1. □

B.5 Proof of Theorem 5.5

Proof of Theorem 5.5. The result in Lemma 5.3 implies that for all $k \in \mathbb{N}$,

$$
\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 + \mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2 \leq \frac{4}{\alpha\eta}\Big(\mathbb{E}\,\tilde{f}(\bar{x}^{(k)}) - \mathbb{E}\,\tilde{f}(\bar{x}^{(k+1)})\Big) + \frac{2L^2}{n}\,\mathbb{E}\|\hat{\mathbf{e}}^{(k)}\|^2 + \frac{2\alpha\eta L\sigma^2}{n},
$$

where $\tilde{f} \triangleq f - f^*$. Taking the average over $k = 0, \dots, T$ gives

$$\frac{1}{T+1}\sum_{k=0}^{T}\Big(\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2+\mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2\Big)\le \frac{4}{\alpha\eta(T+1)}\,\mathbb{E}\,\tilde{f}(\bar{x}^{(0)})+\frac{2L^2}{n(T+1)}\sum_{k=0}^{T}\mathbb{E}\|\hat{\mathbf{e}}^{(k)}\|^2+\frac{2\alpha\eta L\sigma^2}{n},$$

where we use the fact $\tilde{f}(x) \geq 0$ for any $x \in \operatorname{dom} f$. Substituting the result from Lemma 5.4 yields

$$\begin{aligned}
&\frac{1}{T+1}\sum_{k=0}^{T}\Big(\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2+\mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2\Big)\\
&\le \frac{4}{\alpha\eta(T+1)}\,\mathbb{E}\,\tilde{f}(\bar{x}^{(0)})+\frac{4L^2}{n(T+1)}\sum_{k=0}^{\tau-1}\mathbb{E}\|\hat{\mathbf{e}}^{(k)}\|^2+\frac{128\eta^4L^4}{T+1}\sum_{k=0}^{T}\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2\\
&\quad+(2n^{-1}\alpha\eta L+96n^{-1}\tau\eta^2L^2+128\eta^4L^4)\sigma^2\\
&\le \frac{4}{\alpha\eta(T+1)}\,\mathbb{E}\,\tilde{f}(\bar{x}^{(0)})+\frac{4L^2}{n(T+1)}\sum_{k=0}^{\tau-1}\mathbb{E}\|\hat{\mathbf{e}}^{(k)}\|^2+\frac{128\eta^4L^4}{T+1}\sum_{k=0}^{T}\Big(\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2+\mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2\Big)\\
&\quad+(2n^{-1}\alpha\eta L+96n^{-1}\tau\eta^2L^2+128\eta^4L^4)\sigma^2.
\end{aligned}$$

By grouping similar terms on the left hand side and simplifying, one finds that

$$\begin{aligned}
&\frac{1-128\eta^4L^4}{T+1}\sum_{k=0}^{T}\Big(\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2+\mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2\Big)\\
&\quad\le \frac{4}{\alpha\eta(T+1)}\,\mathbb{E}\,\tilde{f}(\bar{x}^{(0)})+\frac{4L^2}{n(T+1)}\sum_{k=0}^{\tau-1}\mathbb{E}\|\hat{\mathbf{e}}^{(k)}\|^2+(2n^{-1}\alpha\eta L+96n^{-1}\tau\eta^2L^2+128\eta^4L^4)\sigma^2.
\end{aligned}$$

The stepsize condition $\eta \leq \frac{1}{2L}$ (required in Lemma 5.3) implies $128\eta^4L^4 \leq \frac{1}{2}$. Thus, the above bound becomes

$$\begin{aligned}
&\frac{1}{T+1}\sum_{k=0}^{T}\Big(\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2+\mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2\Big)\\
&\le \frac{8}{\alpha\eta(T+1)}\,\mathbb{E}\,\tilde{f}(\bar{x}^{(0)})+\frac{8L^2}{n(T+1)}\sum_{k=0}^{\tau-1}\mathbb{E}\|\hat{\mathbf{e}}^{(k)}\|^2+(4n^{-1}\alpha\eta L+192n^{-1}\tau\eta^2L^2+256\eta^4L^4)\sigma^2.
\end{aligned}\tag{50}$$

From now on, we use the notation $\lesssim$ to hide irrelevant constants. Note that $a \lesssim b$ means that there exists a positive constant $\gamma \in \mathbb{R}_{>0}$ such that $a \leq \gamma b$. In our case, the important quantities that we keep are α, η, n, L, and σ. Then, (50) can be written as

$$\frac{1}{T+1}\sum_{k=0}^{T}\Big(\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2+\mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2\Big)\lesssim\frac{1}{\alpha\eta T}+\frac{L^2}{nT}+\eta^4L^4\sigma^2+\frac{\tau\eta^2L^2\sigma^2}{n}+\frac{\alpha\eta L\sigma^2}{n}.$$

We then apply the stepsize condition $\alpha \leq \frac{1}{\tau}$ and $\eta \in \big(0, \frac{1}{4\sqrt{5}\tau L}\big]$ and obtain

$$\begin{aligned}
\frac{1}{T+1}\sum_{k=0}^{T}\Big(\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2+\mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2\Big)&\lesssim\frac{1}{\alpha\eta T}+\frac{L^2}{nT}+\eta^4L^4\sigma^2+\frac{\tau\eta^2L^2\sigma^2}{n}+\frac{\alpha\eta L\sigma^2}{n}\\
&\lesssim\frac{\tau L^2}{\eta T}+\tau\eta^2L^2\sigma^2+\frac{\eta L\sigma^2}{n\tau}\\
&\lesssim\frac{\tau^2L^3}{T}+\frac{\sigma^2}{\tau}+\frac{\sigma^2}{n\tau^2}.
\end{aligned}$$

$\square$

Next, we present the proof of Corollary 5.6.

Proof of Corollary 5.6. The result in Theorem 5.5 can be further simplified as

$$\frac{1}{T+1}\sum_{k=0}^{T}\left(\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^k)\|^2 + \mathbb{E}\|\nabla f(\bar{x}^k)\|^2\right) \lesssim \frac{c_0}{\eta T} + c_1\eta + c_2\eta^2,$$

where $c_0 = \tau L^2$, $c_1 = \frac{L\sigma^2}{n\tau}$, and $c_2 = \tau L^2\sigma^2$. Now we set the stepsize η as

$$\eta = \min\left\{\left(\frac{c_0}{c_1 T}\right)^{\frac{1}{2}}, \left(\frac{c_0}{c_2 T}\right)^{\frac{1}{3}}, \frac{1}{2L}, \frac{1}{4\sqrt{5}\tau L}\right\}.$$

By definition, this choice of η satisfies the stepsize condition in Theorem 5.5: $\eta \leq \frac{1}{\tilde{\eta}} \triangleq \min\left\{\frac{1}{2L}, \frac{1}{4\sqrt{5}\tau L}\right\}$. We then discuss the following three cases.

(a) $\eta = \min\left\{\frac{1}{2L}, \frac{1}{4\sqrt{5}\tau L}\right\} \leq \min\left\{\left(\frac{c_0}{c_1 K}\right)^{\frac{1}{2}}, \left(\frac{c_0}{c_2 T}\right)^{\frac{1}{3}}\right\} \quad \Longrightarrow \quad \frac{c_0}{\eta T} + \tilde{\eta}c_1 + c_2\eta^2 \lesssim \frac{\eta c_0}{T} + \left(\frac{c_0 c_1}{T}\right)^{\frac{1}{2}} + \frac{c_0^{\frac{2}{3}} c_2^{\frac{1}{3}}}{T^{\frac{1}{3}}}$;

(b) $\eta = \left(\frac{c_0}{c_1 T}\right)^{\frac{1}{2}} \leq \left(\frac{c_0}{c_2 T}\right)^{\frac{1}{3}} \quad \Longrightarrow \quad \frac{c_0}{\eta T} + c_1\eta + c_2\eta^2 \lesssim \frac{c_0^{\frac{1}{2}} c_1^{\frac{1}{2}}}{T^{\frac{1}{2}}} + \frac{c_0^{\frac{2}{3}} c_2^{\frac{1}{3}}}{T^{\frac{2}{3}}}$;

(c) $\eta = \left(\frac{c_0}{c_2 T}\right)^{\frac{1}{3}} \leq \left(\frac{c_0}{c_1 T}\right)^{\frac{1}{2}} \quad \Longrightarrow \quad \frac{c_0}{\eta T} + c_1\eta + c_2\eta^2 \lesssim \frac{c_0^{\frac{2}{3}} c_2^{\frac{1}{3}}}{T^{\frac{2}{3}}} + \left(\frac{c_0 c_1}{T}\right)^{\frac{1}{2}}$.

Combining all three cases yields

$$\begin{aligned}
\frac{c_0}{\eta T} + c_1\eta + c_2\eta^2 &\lesssim \frac{\tilde{\eta}c_0}{T} + c_2^{\frac{1}{3}}\left(\frac{c_0}{T}\right)^{\frac{2}{3}} + c_1^{\frac{1}{2}}\left(\frac{c_0}{T}\right)^{\frac{1}{2}} \\
&\lesssim \frac{\tau L c_0}{T} + c_2^{\frac{1}{3}}\left(\frac{c_0}{T}\right)^{\frac{2}{3}} + c_1^{\frac{1}{2}}\left(\frac{c_0}{T}\right)^{\frac{1}{2}} \\
&\lesssim \frac{\tau^2 L^3}{K} + (L^2\tau\sigma^2)^{\frac{1}{3}}\left(\frac{\tau L^2}{K}\right)^{\frac{2}{3}} + \left(\frac{L\sigma^2}{\tau n}\right)^{\frac{1}{2}}\left(\frac{\tau L^2}{K}\right)^{\frac{1}{2}} \\
&\lesssim \frac{\tau^2 L^3}{T} + L^2\tau\left(\frac{\sigma}{T}\right)^{\frac{2}{3}} + \left(\frac{L^3\sigma}{nT}\right)^{\frac{1}{2}}.
\end{aligned}$$

□

Now we present the proof of Corollary 5.7.

Proof of Corollary 5.7. If we perform the AllReduce warm-up strategy, it holds that $\sum_{k=0}^{\tau-1}\left(\mathbb{E}\|\hat{\mathbf{e}}^{(k)}\|^2\right) = 0$. Starting from (50), we can then find

$$\begin{aligned}
&\frac{1}{T+1}\sum_{k=0}^{T}\left(\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 + \mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2\right) \\
&\leq \frac{8}{\alpha\eta(T+1)}\,\mathbb{E}\,\tilde{f}(\bar{x}^{(0)}) + (4n^{-1}\alpha\eta L + 192n^{-1}\tau\eta^2 L^2 + 256\eta^4 L^4)\sigma^2 \\
&\lesssim \frac{1}{\alpha\eta T} + \eta^4 L^4\sigma^2 + \frac{\tau\eta^2 L^2\sigma^2}{n} + \frac{\alpha\eta L\sigma^2}{n}.
\end{aligned}$$

We then apply the stepsize condition $\alpha \leq \frac{1}{\tau}$ and $\eta \in \left(0, \frac{1}{4\sqrt{5}\tau L}\right]$ and obtain

$$\frac{1}{T+1}\sum_{k=0}^{T}\left(\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^{(k)})\|^2 + \mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2\right) \lesssim \frac{1}{\alpha\eta T} + \eta^4 L^4\sigma^2 + \frac{\tau\eta^2 L^2\sigma^2}{n} + \frac{\alpha\eta L\sigma^2}{n}$$

$$
\begin{aligned}
&\lesssim \frac{1}{\alpha\eta T} + \tau\eta^2 L^2\sigma^2 + \frac{\eta L\sigma^2}{n\tau} \\
&\lesssim \frac{\tau L}{T} + \frac{\sigma^2}{\tau} + \frac{\sigma^2}{n\tau^2}.
\end{aligned}
$$

The result in Theorem 5.5 can be further simplified as

$$
\frac{1}{T+1}\sum_{k=0}^{T}\left(\mathbb{E}\|\overline{\nabla f}(\mathbf{x}^k)\|^2 + \mathbb{E}\|\nabla f(\bar{x}^k)\|^2\right) \lesssim \frac{c_0}{\eta T} + c_1\eta + c_2\eta^2,
$$

where $c_0 = \tau$, $c_1 = \frac{L\sigma^2}{n\tau}$, and $c_2 = \tau L^2\sigma^2$. Now we set the stepsize η as

$$
\eta = \min\left\{\left(\frac{c_0}{c_1 T}\right)^{\frac{1}{2}}, \left(\frac{c_0}{c_2 T}\right)^{\frac{1}{3}}, \frac{1}{2L}, \frac{1}{4\sqrt{5}\tau L}\right\}.
$$

By definition this choice of η satisfies the stepsize condition in Theorem 5.5: $\eta \le \frac{1}{\tilde{\eta}} \triangleq \min\left\{\frac{1}{2L}, \frac{1}{4\sqrt{5}\tau L}\right\}$. We then discuss the following three cases.

(a) $\eta = \min\left\{\frac{1}{2L}, \frac{1}{\sqrt{80}\tau L}\right\} \le \min\left\{\left(\frac{c_0}{c_1 K}\right)^{\frac{1}{2}}, \left(\frac{c_0}{c_2 T}\right)^{\frac{1}{3}}\right\} \quad \Longrightarrow \frac{c_0}{\eta T} + \tilde{\eta}c_1 + c_2\eta^2 \lesssim \frac{\eta c_0}{T} + \left(\frac{c_0 c_1}{T}\right)^{\frac{1}{2}} + \frac{c_0^{\frac{2}{3}} c_2^{\frac{1}{3}}}{T^{\frac{1}{3}}}$;

(b) $\eta = \left(\frac{c_0}{c_1 T}\right)^{\frac{1}{2}} \le \left(\frac{c_0}{c_2 T}\right)^{\frac{1}{3}} \quad \Longrightarrow \frac{c_0}{\eta T} + c_1\eta + c_2\eta^2 \lesssim \frac{c_0^{\frac{1}{2}} c_1^{\frac{1}{2}}}{T^{\frac{1}{2}}} + \frac{c_0^{\frac{2}{3}} c_2^{\frac{1}{3}}}{T^{\frac{2}{3}}}$;

(c) $\eta = \left(\frac{c_0}{c_2 T}\right)^{\frac{1}{3}} \le \left(\frac{c_0}{c_1 T}\right)^{\frac{1}{2}} \quad \Longrightarrow \frac{c_0}{\eta T} + c_1\eta + c_2\eta^2 \lesssim \frac{c_0^{\frac{2}{3}} c_2^{\frac{1}{3}}}{T^{\frac{2}{3}}} + \left(\frac{c_0 c_1}{T}\right)^{\frac{1}{2}}$.

Combining all three cases yields

$$
\begin{aligned}
\frac{c_0}{\eta T} + c_1\eta + c_2\eta^2 &\lesssim \frac{\tilde{\eta}c_0}{T} + c_2^{\frac{1}{3}}\left(\frac{c_0}{T}\right)^{\frac{2}{3}} + c_1^{\frac{1}{2}}\left(\frac{c_0}{T}\right)^{\frac{1}{2}} \\
&\lesssim \frac{\tau L c_0}{T} + c_2^{\frac{1}{3}}\left(\frac{c_0}{T}\right)^{\frac{2}{3}} + c_1^{\frac{1}{2}}\left(\frac{c_0}{T}\right)^{\frac{1}{2}} \\
&\lesssim \frac{\tau^2 L}{K} + (L^2\tau\sigma^2)^{\frac{1}{3}}\left(\frac{\tau}{K}\right)^{\frac{2}{3}} + \left(\frac{L\sigma^2}{\tau n}\right)^{\frac{1}{2}}\left(\frac{\tau}{K}\right)^{\frac{1}{2}} \\
&\lesssim \frac{\tau^2 L}{T} + \tau\left(\frac{L\sigma}{T}\right)^{\frac{2}{3}} + \left(\frac{L\sigma}{nT}\right)^{\frac{1}{2}}.
\end{aligned}
$$

□