

# On graphs with finite-time consensus and their use in gradient tracking\*

Edward Duc Hien Nguyen<sup>†</sup>   Xin Jiang<sup>‡</sup>   Bicheng Ying<sup>§</sup>   César A. Uribe<sup>¶</sup>

January 29, 2025

## Abstract

This paper studies sequences of graphs satisfying the finite-time consensus property (*i.e.*, iterating through such a finite sequence is equivalent to performing global or exact averaging) and their use in the decentralized optimization algorithm Gradient Tracking. For each of the studied graph sequences, we provide an explicit weight matrix representation and prove their finite-time consensus property. Moreover, we incorporate such topology sequences 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 optimization problem

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

each  $f_i: \mathbb{R}^d \rightarrow \mathbb{R}$  is a smooth, possibly nonconvex function, and the symbol  $\mathbb{E}_{\xi_i}$  denotes the expected value of the random variable  $\xi_i$  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  $\xi_i$ . Algorithms for solving this type of problems often use stochastic gradients of  $f_i$ , and are performed in a decentralized manner; *i.e.*, each function  $f_i$  and probability space  $\{\Omega_i, \mathcal{F}_i, \mathbb{P}_i\}$  are held exclusively and privately by an agent  $i \in \{1, \dots, n\}$ . Consequently, in decentralized algorithms for solving (1), agents must communicate

---

\* Author's final version. Accepted for publication in *SIAM Journal on Optimization*.

<sup>†</sup>Department of Electrical and Computer Engineering, Rice University. Email: [en18@rice.edu](mailto:en18@rice.edu).

<sup>‡</sup>School of Operations Research and Information Engineering, Cornell University. Email: [xjiang@cornell.edu](mailto:xjiang@cornell.edu). Much of the work was done when XJ was at Lehigh University.

<sup>§</sup>Google Inc. Email: [ybc@google.com](mailto:ybc@google.com).

<sup>¶</sup>Department of Electrical and Computer Engineering and Ken Kennedy Institute, Rice University. Email: [cau-ribe@rice.edu](mailto:cau-ribe@rice.edu).

with one another according to some network topology. We are motivated by the application of decentralized optimization algorithms to the high-performance computing scenario in which computing resources can be abstracted as agents. Under this scenario, not only is the network topology robust but it is also flexible. Therefore, it is critical to select topologies that reduce communication costs [4, 12, 18, 29].

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 [22, 27, 29] 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 [9, 10], one-peer hyper-cubes [22], one-peer exponential graphs [29], 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 (e.g., [2, 3, 16]) 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. Hence, novel analysis for decentralized algorithms using finite-time consensus must be developed.

**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 (with  $n = 2^r$ ) in [29], we leverage their circulant property to develop an alternative proof for the finite-time consensus property. 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 establishes that the convergence rate is independent of the connectivity of any of the individual graphs used in the algorithm. Furthermore, we show that GT-FT using the presented graph sequences with finite-time consensus has the same iteration complexity as GT using the static counterparts. 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

In this section, we first review 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 straightforwardly extended to incorporate sequences of topologies that satisfy the finite-time consensus property. We then discuss prior work that studied sequences of topologies satisfying the finite-time consensus property.

The Decentralized Stochastic Gradient method (DSGD) [\[7, 19, 26\]](#) is arguably the most popular and widely studied decentralized algorithm due to its simplicity. However, current analysis of DSGD relies on a bounded heterogeneity assumption, which bounds the allowed difference between local functions/data. The effect of heterogeneity is further magnified by large and sparse topologies, which further degrades the performance of DSGD [\[15\]](#).

Exact or bias-corrected decentralized algorithms have been proposed to circumvent the negative effects caused by heterogeneity on the convergence of DSGD. Examples of these algorithms include EXTRA [\[23\]](#), Exact Diffusion [\[31\]](#), and gradient tracking methods [\[20\]](#). These algorithms have been studied in a unified framework in [\[1\]](#) and [\[2\]](#) without the bounded heterogeneity assumption.

The convergence of EXTRA and Exact Diffusion has only been established for symmetric and static mixing matrices [\[2\]](#). On the contrary, gradient tracking methods with time-varying topologies (TV-GT) have been studied under various assumptions. Nedić et al. [\[20\]](#) analyze TV-GT, which they call DIGing, for the strongly convex scenario, with true gradient evaluation and with  $\tau$ -connected graphs. The assumption of  $\tau$ -connected graphs means that the union of a sequence of  $\tau$ -length graphs is connected, and is weaker than assuming that the topology at each iteration is connected. Another example of TV-GT is the algorithm NEXT [\[11\]](#) (and its extension SONATA [\[21\]](#)), which are analyzed for the nonconvex composite optimization problems with true gradient evaluation and with  $\tau$ -connected graphs. We note that DIGing [\[20\]](#), NEXT [\[11\]](#), and SONATA [\[21\]](#) do not cover the case in which agents can only have access to stochastic gradient estimates of their local objective functions. However, extending the aforementioned analyses to account for stochastic gradient estimates is nontrivial and leads to a new line of research. The more recent analyses performed in [\[2, 16\]](#) successfully analyze Gradient Tracking using stochastic gradient estimates, but their analyses are limited to only considering static graphs. In particular, it is explicitly stated in [\[16\]](#) that the extension of their analysis to the time-varying setting is nontrivial. The analysis in [\[2, 16\]](#) relies on the eigendecomposition of the weighted adjacency matrix associated with the considered static graph. Thus, it is assumed in [\[2, 16\]](#) that the underlying graph is undirected, which does hold for, *e.g.*, one-peer exponential graphs considered in our setting. Even if all the graphs in our proposed sequence were undirected, the product of two symmetric weighted adjacency matrices is not necessarily symmetric. Therefore, the analyses in [\[2, 16\]](#) remain inapplicable. Later, Song et al. [\[24\]](#) provides convergence analysis for TV-GT under the nonconvex and stochastic setting and makes no assumption on the symmetry of the mixing matrices. Nevertheless, their convergence results rely on the smallest connectivity (in expectation) of the set of all (random) 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. In summary, existing analyses of gradient tracking are limited in their capacity to encapsulate both stochastic gradients and time-varying graphs, and thus cannot be readily extended to capture our setting

(graph sequence with finite-time consensus).

Graph sequences with the so-called *finite-time consensus* have been proposed and analyzed in various contexts. Delvenne et al. [10] establish the finite-time property for a sequence of static de Bruijn graphs. Shi et al. [22] justify the finite-time consensus property for a sequence of one-peer hyper-cubes (when  $n = 2^\tau$  for some  $\tau \in \mathbb{N}_{\geq 1}$ ). Assran et al. [4] observe in numerical experiments that one-peer exponential graphs (with  $n = 2^\tau$ ) have the finite-time consensus property, which is later theoretically justified in [29]. More recently, Takezawa et al. [27] claim to build graph sequences with finite-time consensus of any node size  $n \in \mathbb{N}$ , but they do not provide any theoretical justification. Although numerical evidence provided in [27] demonstrates the finite-time consensus property of their proposed graphs, no explicit matrix representation is given and further investigation is needed. In another line of research, Ding et al. [12] extend the optimal message passing algorithm developed in [5] and propose a communication-optimal exact consensus algorithm. The algorithm proposed in [12] requires an additional copy of the optimization variable at each agent, and with the help of these auxiliary variables, achieves “finite-time consensus” for an arbitrary number of agents. The discussion of this approach is out of the scope of this paper, and further investigation is left as future work. Table 1 summarizes the existing graph sequences for which finite-time consensus is proven to hold. Despite the usefulness of such graph sequences, incorporating them in decentralized optimization algorithms is not straightforward. The only theoretical result in this regard, to the best of our knowledge, is [29], which incorporates the one-peer exponential graphs into the Decentralized Momentum Stochastic Gradient (DmSGD) method. The authors of [29] show that the convergence rate of DmSGD using one-peer exponential graphs is the same as DmSGD using static exponential graphs. Yet, the final convergence rate derived in [29] fails to consider the stepsize condition imposed by their analysis. Specifically, to derive the final convergence rate in [29, Equation (107)], the authors select a stepsize that may be incompatible with some previously imposed stepsize conditions (*i.e.*, [29, Equation (105)]). Moreover, we note that the analysis of DSGD provided in [15] can be adapted to encapsulate finite-time consensus graphs while DSGD is a special case of DmSGD (with the momentum parameter  $\beta = 0$ ).

### 3 Finite-time consensus

In this section, we formally define the finite-time consensus property, and establish this property for several sequences of graphs. For one-peer exponential graphs (with  $n = 2^\tau$ ) presented in [29], we present an alternative proof based on the circulant property. Moreover, we present  $p$ -peer hyper-cuboids of an arbitrary node size, and establish their finite-time consensus property. Finally, we show that in certain cases, the presented  $p$ -peer hyper-cuboids are permutation equivalent to the de Bruijn graphs.

**Notation** Only in this section, we modify the convention for indexing a matrix  $W \in \mathbb{R}^{n \times n}$ . In this section, we start the indexing at 0, *i.e.*, the entries of an  $n \times n$  matrix  $W$  are

$$W = [w_{ij}], \quad \text{for } i, j = 0, 1, \dots, n-1.$$

In the rest of the paper, we follow the convention and start the indexing at 1, *i.e.*,

$$W = [w_{ij}], \quad \text{for } i, j = 1, 2, \dots, n.$$

Table 1: Classes of graph sequences that satisfy the finite-time consensus property.

Topology	Orientation	Size $n$	Maximum degree	Num. of iterations ( $\tau$ ) for finite-time consensus
de Bruijn [10]	Directed	Power of $p \in \mathbb{N}_{\geq 2}$	$p - 1$	$\log_p(n)$
One-peer exponential [29]	Directed	Power of 2 <sup>a</sup>	1	$\log_2(n)$
One-peer hyper-cube [22]	Undirected	Power of 2	1	$\log_2(n)$
$p$ -Peer hyper-cuboid	Undirected	Any $n \in \mathbb{N}$	flexible <sup>b</sup>	flexible <sup>b</sup>

<sup>a</sup> When  $n \neq 2^\tau$ , one-peer exponential graphs are still well defined but do not have finite-time consensus property.

<sup>b</sup> For  $p$ -peer hyper-cuboids, the maximum degree and the number of iterations for finite-time consensus depend on a factorization of the node size  $n$  and exhibits a trade-off. For example, when  $n = 20$  is factorized as  $n = 20 = 2 \times 2 \times 5$ , the number of iterations for finite-time consensus is  $\tau = 3$ , the number of factors, while the maximum degree in all the three graphs is 4, the largest factor minus one. Another possible factorization is  $n = 20 = 2 \times 10$ , in which case  $\tau = 2$  and the maximum degree (in the two graphs) is 9. Also note that when  $n$  is a prime number, the  $p$ -peer hyper-cuboids reduce to the fully-connected graph.

### 3.1 Definition

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.** *The sequence of graphs  $\mathcal{G}^{(l)} = (\mathcal{V}, W^{(l)}, \mathcal{E}^{(l)})$ ,  $l = 0, \dots, \tau - 1$ , 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*

$$W^{(\tau-1)}W^{(\tau-2)} \dots W^{(1)}W^{(0)} = \frac{1}{n}\mathbf{1}\mathbf{1}^\top. \quad (2)$$

For a sequence of doubly stochastic matrices  $\{W^{(l)}\}_{l=0}^{\tau-1}$ , (2) is equivalent to

$$(W^{(\tau-1)} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)(W^{(\tau-2)} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top) \dots (W^{(1)} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)(W^{(0)} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top) = 0. \quad (3)$$

The parameter  $\tau$  in Definition 3.1 is not arbitrary and might depend on the matrix size  $n$  and graph structure. For the examples provided in Sections 3.2 to 3.4, the parameter  $\tau$  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 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.

More discussion on the existing graph sequences with finite-time consensus can be found in Section 2. 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 [9, 10], we establish its connection to the  $p$ -peer hyper-cuboids.

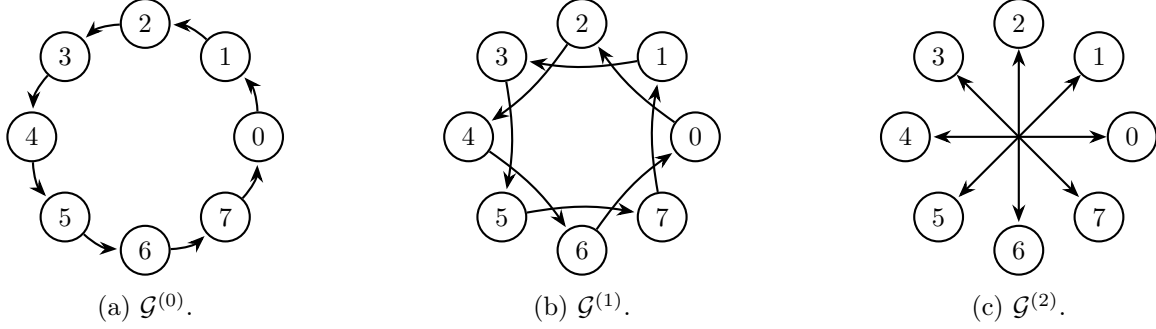


Figure 1: The three one-peer exponential graphs  $\{\mathcal{G}^{(l)}\}_{l=0}^2$  with  $n = 8$  and  $\tau = 3$ . Note that all nodes have self-loops, though not explicitly shown in the figure.

### 3.2 One-peer exponential graphs

We present the weight matrices of one-peer exponential graphs (which was developed in [29]) and list several properties of our interest. 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 (2). As a byproduct, we develop an alternative proof for the finite-time consensus property of one-peer exponential graphs (when  $n = 2^\tau$ ).

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

$$w_{ij}^{(l)} = \begin{cases} \frac{1}{2} & \text{if } \text{mod}(j - i, n) = 2^{\text{mod}(l, \tau)} \\ \frac{1}{2} & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

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 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  $\Theta(1)$ . 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 (4). Each matrix  $W^{(l)}$  is circulant and doubly stochastic, i.e.,  $W^{(l)}\mathbf{1} = \mathbf{1}$  and  $\mathbf{1}^\top W^{(l)} = \mathbf{1}^\top$ . 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  $\{\text{mod}(l_0, \tau), \dots, \text{mod}(l_{\tau-1}, \tau)\} = \{0, \dots, \tau-1\}$ , 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 (2). 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 [\[14, §4.7.7\]](#).

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

$$C = \text{Circ}(c_0, c_1, \dots, c_{n-1}) \triangleq \begin{bmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & & \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \ddots & c_1 & c_0 \end{bmatrix}. \quad (5)$$

Let  $\hat{j}$  be the imaginary number ( $\hat{j}^2 = -1$ ) and let  $\omega = \exp\left(\frac{2\pi\hat{j}}{n}\right)$  be a primitive  $n$ -th root of unity. Denote by  $\mathbf{F}$  by the  $n \times n$  discrete Fourier transform (DFT) matrix ( $\mathbf{F}_{ij} = \frac{1}{\sqrt{n}}\omega^{ij}$ ), and denote by  $\mathbf{F}^H$  the Hermitian (conjugate transpose) of  $\mathbf{F}$ . Then, the eigenvalue decomposition of  $C$  (5) is given by

$$C = \left(\frac{1}{\sqrt{n}}\mathbf{F}\right) \cdot (\text{diag}(\mathbf{F}c)) \cdot \left(\frac{1}{\sqrt{n}}\mathbf{F}^H\right),$$

where the  $\text{diag}$  operator transforms an  $n$ -vector into an  $n \times n$  diagonal matrix. Moreover, the eigenvalues of  $C$  (5) are given by

$$\lambda_k = c_0 + c_1\omega^k + c_2\omega^{2k} + \cdots + c_{n-1}\omega^{(n-1)k}, \quad k = 0, 1, \dots, n-1. \quad (6)$$

Now we present the proof for [Proposition 3.1](#). The original proof in [\[29\]](#) leverages the binary representation of the integer  $n$ , but our proof is inspired by the circulant property of one-peer exponential graphs. (Note that the circulant property has been exploited in [\[29\]](#) to compute the eigenvalues of the *static* exponential graph but was not used to establish finite-time consensus.)

*Proof of Proposition 3.1.* Note that the mixing matrices  $\{W^{(l)}\}_{l=0}^{\tau-1}$  of one-peer exponential graphs defined in [\(4\)](#) are circulant. Thus, we have

$$W^{(\tau-1)} \cdots W^{(1)} W^{(0)} = \left(\frac{1}{\sqrt{n}}\mathbf{F}\right) \cdot (\Lambda^{(\tau-1)} \cdots \Lambda^{(1)} \Lambda^{(0)}) \cdot \left(\frac{1}{\sqrt{n}}\mathbf{F}^H\right),$$

where  $\Lambda^{(l)} = \text{diag}(\mathbf{F}c^{(l)})$  and  $c^{(l)}$  is the first column of  $W^{(l)}$ , for  $l = 0, 1, \dots, \tau-1$ . For simplicity, we denote  $\Lambda \triangleq \Lambda^{(\tau-1)} \cdots \Lambda^{(1)} \Lambda^{(0)}$ .

For each  $l = 0, 1, \dots, \tau-1$ , it follows from [\(6\)](#) in [Lemma 3.2](#) that the first element in  $\mathbf{F}c^{(l)}$  is 1. This implies that the first element in  $\Lambda$  is also 1. Similarly, the  $k$ -th diagonal element in  $\Lambda$ , denoted by  $\Lambda[k]$ , can be found by expanding the definition  $\Lambda^{(l)} = \text{diag}(\mathbf{F}c^{(l)})$ :

$$\begin{aligned} \Lambda[k] &= (\Lambda^{(\tau-1)}[k]) \cdots (\Lambda^{(2)}[k]) (\Lambda^{(1)}[k]) (\Lambda^{(0)}[k]) \\ &= \frac{1}{2^\tau} \left( (1 + \omega^{k(n-2^{\tau-1})}) \cdots (1 + \omega^{k(n-4)}) (1 + \omega^{k(n-2)}) (1 + \omega^{k(n-1)}) \right) \\ &= \frac{1}{2^\tau} \left( (1 + \omega^{-2^{\tau-1}k}) \cdots (1 + \omega^{-4k}) (1 + \omega^{-2k}) (1 + \omega^{-k}) \right) \\ &= \frac{1}{2^\tau} \sum_{q=0}^{n-1} \omega^{-kq} = \frac{1}{2^\tau} \left( \frac{1 - \omega^{-kn}}{1 - \omega^{-k}} \right) = 0, \end{aligned}$$



where in the first equation we use the assumption that  $n = 2^\tau$ . Hence, we have

$$\begin{aligned} W^{(\tau-1)} \dots W^{(1)} W^{(0)} &= \left(\frac{1}{\sqrt{n}} \mathbf{F}\right) \Lambda^{(\tau-1)} \dots \Lambda^{(1)} \Lambda^{(0)} \left(\frac{1}{\sqrt{n}} \mathbf{F}^H\right) \\ &= \left(\frac{1}{\sqrt{n}} \mathbf{F}\right) \text{diag}(1, 0, 0, \dots, 0) \left(\frac{1}{\sqrt{n}} \mathbf{F}^H\right) \\ &= \frac{1}{n} \mathbf{1} \mathbf{1}^T, \end{aligned}$$

which establishes the desirable identity (2). To show (3), we first observe that

$$(W^{(l+1)} - \frac{1}{n} \mathbf{1} \mathbf{1}^T) (W^{(l)} - \frac{1}{n} \mathbf{1} \mathbf{1}^T) = W^{(l+1)} W^{(l)} - \frac{1}{n} \mathbf{1} \mathbf{1}^T, \text{ for all } l = 0, \dots, \tau - 2.$$

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

$$(W^{(\tau-1)} - \frac{1}{n} \mathbf{1} \mathbf{1}^T) \dots (W^{(0)} - \frac{1}{n} \mathbf{1} \mathbf{1}^T) = W^{(\tau-1)} \dots W^{(0)} - \frac{1}{n} \mathbf{1} \mathbf{1}^T = 0,$$

where in the last step we use (2). Combining this result with the fact that one-peer exponential graphs are periodic, and all instances of one-peer exponential graphs are commutative, we obtain that any permutation of any sequence of  $W^{(l)}$  of length larger than  $\tau$  has the finite-time consensus property as well.  $\square$

### 3.3 One-peer hyper-cubes

Another example of graph sequences that satisfy Definition 3.1 is one-peer hyper-cubes [22] (with  $n$  a power of 2). Hyper-cubes have been extensively studied in theoretical computer science (see, e.g., [13] 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 [22]. 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.

Let  $n = 2^\tau$  with  $\tau \in \mathbb{N}_{\geq 1}$ , the weight matrices  $\{W^{(l)}\}_{l \in \mathbb{N}} \subset \mathbb{R}^{n \times n}$  representing the one-peer hyper-cubes  $\{\mathcal{G}^{(l)}\}_{l \in \mathbb{N}}$  are defined by

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

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 (4) is minor yet critical: The operation  $(i \wedge j)$  is used in (7) while  $\text{mod}(j-i, n)$  in (4). Since  $(i \wedge j) = (j \wedge i)$ , it immediately follows that the mixing matrices of one-peer hyper-cubes are symmetric. We represent the integer  $i$  in its binary form  $(i_{\tau-1} i_{\tau-2} \dots i_0)_2$ . Then, the first if-condition in (7) can be rewritten as

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

that is, only the  $(\text{mod}(l, \tau) + 1)$ -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  $\tau$ -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.



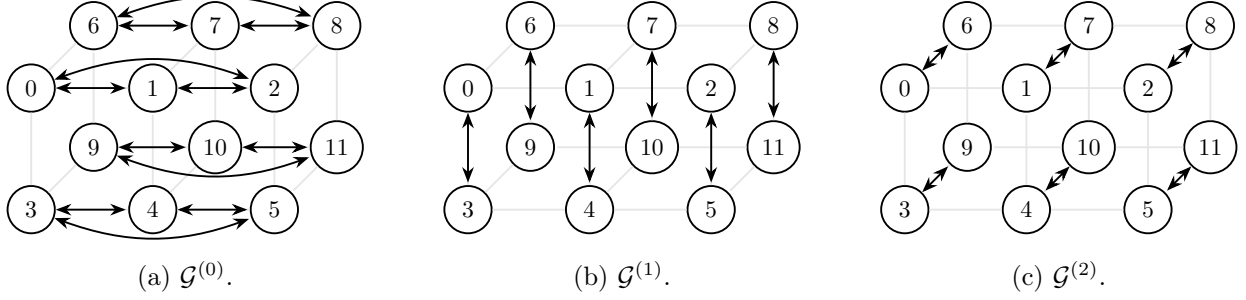


Figure 2: The three 2-peer hyper-cuboids  $\{\mathcal{G}^{(l)}\}_{l=0}^2$  with  $n = 12$ ,  $(p_2, p_1, p_0) = (2, 2, 3)$ , and  $\tau = 3$ . Note that all nodes have self-loops, though not explicitly shown here.

### 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.*, [8]. 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  $n \in \mathbb{N}_{\geq 2}$ .

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 [17]. To be specific, the  $(p_{\tau-1}, p_{\tau-2}, \dots, p_0)$ -based representation of an integer is an element in the group  $\mathbb{N}_{p_{\tau-1}} \times \mathbb{N}_{p_{\tau-2}} \times \dots \times \mathbb{N}_{p_0}$ , where  $\mathbb{N}_{p_k}$  is the group of nonnegative integers modulo  $p_k \in \mathbb{N}_{\geq 2}$ . Any natural integer smaller than  $p_{\tau-1} \times p_{\tau-2} \times \dots \times p_0$  finds a one-to-one mapping in this group. For example, a  $(2, 2, \dots, 2)$ -based representation is equivalent to the binary representation of an integer. A more informative example is the  $(2, 3)$ -based representation. In this case, we can map the integers in  $\{0, 1, \dots, 5\}$  according to the following rule:

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

To shorten the notation, we overload binary representation and denote the  $(p_{\tau-1}, p_{\tau-2}, \dots, p_0)$ -based representation of  $i \in \mathbb{N}$  as  $(i_{p_{\tau-1}} \dots i_{p_1} i_{p_0})_{p_{\tau-1}, \dots, p_1, p_0}$ , 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  $n = p_{\tau-1} \dots p_1 p_0$ , where all the  $p_j$  are prime numbers, and define the  $\tau$ -vector  $p = (p_0, p_1, \dots, p_{\tau-1}) \in \mathbb{N}^\tau$ . (It is possible that  $p_i = p_j$  for  $i \neq j$ , and the order of  $\{p_j\}$  does not matter. Also, prime factorization is considered here only for simplicity. Any factorization of  $n$  would build a sequence of  $p$ -peer hyper-cuboids; see also the footnote of Table 1.) Then, the weight matrices of  $p$ -peer hyper-cuboids are defined by

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

for  $l = 0, 1, \dots, \tau - 1$ , where  $i \wedge_{p_{\tau-1}, \dots, p_1, p_0} j$  denotes the bit-wise XOR operation between the  $(p_{\tau-1}, \dots, p_1, p_0)$ -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

$$8 \rightarrow \{1\}_2 \times \{0\}_2 \times \{2\}_3, \quad 11 \rightarrow \{1\}_2 \times \{1\}_2 \times \{2\}_3.$$

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 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 (8) also has an elegant representation in terms of Kronecker products:

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

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

$$W^{(l)}(p_r) = \begin{cases} I_{p_r} & \text{if } \text{mod}(l, \tau) \neq r \\ \frac{1}{p_r} \mathbf{1}\mathbf{1}^\top & \text{if } \text{mod}(l, \tau) = r. \end{cases} \quad (10)$$

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

$$\begin{aligned} W^{(l)} &= \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} w_{ij}^{(l)} e_i e_j^\top \\ &= \sum_{i_{p_{\tau-1}}, \dots, i_{p_0}} \dots \sum_{j_{p_{\tau-1}}, \dots, j_{p_0}} w_{ij}^{(l)} (\hat{e}_{i_{p_{\tau-1}}} \otimes \dots \otimes \hat{e}_{i_{p_0}}) (\hat{e}_{j_{p_{\tau-1}}} \otimes \dots \otimes \hat{e}_{j_{p_0}})^\top \\ &= \sum_{i_{p_{\tau-1}}, \dots, i_{p_0}} \sum_{j_{p_r} : r=\text{mod}(l, \tau)} \frac{1}{p_r} (\hat{e}_{i_{p_{\tau-1}}} \hat{e}_{i_{p_{\tau-1}}}^\top) \otimes \dots \otimes (\hat{e}_{i_{p_r}} \hat{e}_{i_{p_r}}^\top) \otimes \dots \otimes (\hat{e}_{i_{p_0}} \hat{e}_{i_{p_0}}^\top) \end{aligned} \quad (11a)$$

$$= \left( \sum_{i_{p_{\tau-1}}} \hat{e}_{i_{p_{\tau-1}}} \hat{e}_{i_{p_{\tau-1}}}^\top \right) \otimes \dots \otimes \left( \sum_{i_{p_r}} \sum_{j_{p_r}} \frac{1}{p_r} \hat{e}_{i_{p_r}} \hat{e}_{j_{p_r}}^\top \right) \otimes \dots \otimes \left( \sum_{i_{p_0}} \hat{e}_{i_{p_0}} \hat{e}_{i_{p_0}}^\top \right) \quad (11b)$$

$$= W^{(l)}(p_{\tau-1}) \otimes \dots \otimes W^{(l)}(p_1) \otimes W^{(l)}(p_0). \quad (11c)$$

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. (Recall that the index starts with 0.) The notation  $\hat{e}_{i_{p_r}}$  is a base unit vector of length  $p_r$ , where all entries are 0 except for the  $(i_{p_r})$ -th entry, and recall  $p_r$  is a prime factor of  $n$ . Step (11a) removes the terms that are 0 using definition (8), and then 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 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*

$$\{\text{mod}(l_0, \tau), \dots, \text{mod}(l_{\tau-1}, \tau)\} = \{0, \dots, \tau - 1\},$$

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

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

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

$$\begin{aligned} \prod_{l=0}^{\tau-1} W^{(l)} &= \prod_{l=0}^{\tau-1} \left( W^{(l)}(p_{\tau-1}) \otimes W^{(l)}(p_{\tau-2}) \otimes \cdots \otimes W^{(l)}(p_0) \right) \\ &= \left( \prod_{l=0}^{\tau-1} W^{(l)}(p_{\tau-1}) \right) \otimes \left( \prod_{l=0}^{\tau-1} W^{(l)}(p_{\tau-2}) \right) \otimes \cdots \otimes \left( \prod_{l=0}^{\tau-1} W^{(l)}(p_0) \right) \\ &= \left( \frac{1}{p_{\tau-1}} \mathbf{1}_{p_{\tau-1}} \mathbf{1}_{p_{\tau-1}}^\top \right) \otimes \cdots \otimes \left( \frac{1}{p_0} \mathbf{1}_{p_0} \mathbf{1}_{p_0}^\top \right) = \frac{1}{n} \mathbf{1} \mathbf{1}^\top, \end{aligned}$$

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  $\tau$  has the finite-time consensus property as well.  $\square$

**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 (2).*

*Proof.* When  $n = 2^\tau$ , the one-peer hyper-cube is just a special case of  $p$ -peer hyper-cuboids. Decomposition of  $n$  into its  $(2, 2, \dots, 2)$ -based representation (or equivalently, binary representation) yields the desired properties.  $\square$

**Connection with de Bruijn graphs** The de Bruijn graph was first studied in [9], and its finite-time consensus property has been established in [10]. Here, we present a formulation of de Bruijn graphs closely related to  $p$ -peer hyper-cuboids (8) and 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  $i \in \{0, 1, \dots, n-1\}$  as  $(i_{\tau-1} i_{\tau-2} \dots i_0)_p$ . Then, the  $n \times n$  weight matrix of the de Bruijn graph is defined by

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

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  $W_{\text{db}} \in \mathbb{R}^{n \times n}$  be the weight matrix of the de Bruijn graph defined in (12) and  $\{W_{\text{hc}}^{(l)}\}_{l \in \mathbb{N}}$  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*

$$W_{\text{hc}}^{(l)} = P^{(l)} W_{\text{db}} (Q^{(l)})^\top. \quad (13)$$

*Proof.* See Appendix A. □

## 4 Algorithm description

This section presents Gradient Tracking with time-varying topologies (TV-GT) and our modified version, Gradient Tracking for 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) [11, 20] is a well-studied decentralized algorithm for solving the problem (1), and various formulations of TV-GT exist in the literature. The presented form follows from [11] (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 stepsize  $\alpha \in \mathbb{R}_{>0}$ , set  $g_i^{(0)} = \nabla F_i(x_i^{(0)}, \xi_i^{(0)})$  for  $i = 1, \dots, n$ . Then, TV-GT takes the following iterations for  $k = 0, 1, 2, \dots$

$$\begin{aligned} x_i^{(k+1)} &= \sum_{j: (j,i) \in \mathcal{E}^{(k)}} w_{ji}^{(k)} (x_j^{(k)} - \alpha g_j^{(k)}) \\ g_i^{(k+1)} &= \sum_{j: (j,i) \in \mathcal{E}^{(k)}} w_{ji}^{(k)} g_j^{(k)} + \nabla F_i(x_i^{(k+1)}, \xi_i^{(k+1)}) - \nabla F_i(x_i^{(k)}, \xi_i^{(k)}), \end{aligned} \quad (14)$$

where the iterates  $g_i^{(k)}$  are often called the *tracking variables*. The TV-GT iterations can be written in a more compact form, which relies on the augmented quantities

$$\begin{aligned} \mathbf{x}^{(k)} &\triangleq \text{col}\{x_1^{(k)}, \dots, x_n^{(k)}\} \in \mathbb{R}^{dn}, & \mathbf{g}^{(k)} &\triangleq \text{col}\{g_1^{(k)}, \dots, g_n^{(k)}\} \in \mathbb{R}^{dn}, \\ \bar{\mathbf{x}}^{(k)} &\triangleq \frac{1}{n} \sum_{i=1}^n x_i^{(k)}, & \mathbf{f}(\mathbf{x}) &\triangleq \frac{1}{n} \sum_{i=1}^n f_i(x_i), & \bar{\nabla} \mathbf{f}(\mathbf{x}^{(k)}) &\triangleq \frac{1}{n} \sum_{i=1}^n \nabla f_i(x_i^{(k)}), \\ \bar{\mathbf{x}}^{(k)} &\triangleq \mathbf{1}_n \otimes \bar{\mathbf{x}}^{(k)}, & \hat{\mathbf{x}}^{(k)} &\triangleq \mathbf{x}^{(k)} - \bar{\mathbf{x}}^{(k)}, \\ \mathbf{W}^{(k)} &\triangleq W^{(k)} \otimes I_d \in \mathbb{R}^{dn \times dn}, & \nabla \mathbf{f}(\mathbf{x}) &\triangleq \text{col}\{\nabla f_1(x_1), \dots, \nabla f_n(x_n)\} \in \mathbb{R}^{dn}, \\ & & \nabla \mathbf{F}(\mathbf{x}; \boldsymbol{\xi}) &\triangleq \text{col}\{\nabla F_1(x_1; \xi_1), \dots, \nabla F_n(x_n; \xi_n)\} \in \mathbb{R}^{dn}. \end{aligned} \quad (15)$$

With the augmented quantities, TV-GT (14) can be written compactly as

$$\mathbf{x}^{(k+1)} = \mathbf{W}^{(k)} (\mathbf{x}^{(k)} - \alpha \mathbf{g}^{(k)})$$

$$\mathbf{g}^{(k+1)} = \mathbf{W}^{(k)} \mathbf{g}^{(k)} + \nabla \mathbf{F}(\mathbf{x}^{(k+1)}; \boldsymbol{\xi}^{(k+1)}) - \nabla \mathbf{F}(\mathbf{x}^{(k)}; \boldsymbol{\xi}^{(k)}).$$

The choice of the matrix sequence  $\{\mathbf{W}^{(k)}\}$  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. This scenario is not encapsulated by existing analysis of TV-GT; see [Section 2](#) for an in-depth discussion.

## 4.2 Gradient tracking for finite-time consensus topologies

We present the modified version of TV-GT in [Algorithm 1](#), where a graph sequence satisfying [Definition 3.1](#) is used for communication. 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.*, [\[20, 21\]](#)), aims to be as general as possible when considering network topologies. In contrast, we aim to specialize the analysis of GT to leverage the largely unexploited finite-time consensus property. To differentiate between the two approaches, we call our scheme GT-FT.

---

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

---

- 1: Agent  $i$  input:  $x_i^{(0)} \in \mathbb{R}^d$  and stepsize  $\alpha \in \mathbb{R}_{>0}$ .
- 2: Global input: The parameter  $\tau \in \mathbb{N}_{\geq 1}$  for finite-time consensus, and the sequence of matrices  $\{\mathbf{W}^{(l)}\}$  that satisfies [Definition 3.1](#).
- 3: Initialize  $\mathbf{g}_i^{(0)} = \nabla F_i(x_i^{(0)}, \xi_i^{(0)}) \in \mathbb{R}^d$ ,  $i = 1, \dots, n$ .
- 4: **for**  $k = 0, 1, 2, \dots$  **do**
- 5:     **for**  $i = 1, \dots, n$  (in parallel) **do**

$$w_{ij}^{(k)} = W^{(\text{mod}(k, \tau))}[i, j], \quad \text{for all } j = 1, \dots, n \quad (16a)$$

$$x_i^{(k+1)} = \sum_{j: (j, i) \in \mathcal{E}^{(k)}} w_{ji}^{(k)} (x_j^{(k)} - \alpha g_j^{(k)}) \quad (16b)$$

$$\mathbf{g}_i^{(k+1)} = \sum_{j: (j, i) \in \mathcal{E}^{(k)}} w_{ji}^{(k)} \mathbf{g}_j^{(k)} + \nabla F_i(x_i^{(k+1)}; \xi_i^{(k+1)}) - \nabla F_i(x_i^{(k)}; \xi_i^{(k)}). \quad (16c)$$

- 6:     **end for**
  - 7: **end for**
- 

Our 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):

$$\mathbf{W}^{(k)} = W^{(\text{mod}(k, \tau))} \otimes I_d \quad (17a)$$

$$\mathbf{x}^{(k+1)} = \mathbf{W}^{(k)} (\mathbf{x}^{(k)} - \alpha \mathbf{g}^{(k)}) \quad (17b)$$

$$\mathbf{g}^{(k+1)} = \mathbf{W}^{(k)} \mathbf{g}^{(k)} + \nabla \mathbf{F}(\mathbf{x}^{(k+1)}; \boldsymbol{\xi}^{(k+1)}) - \nabla \mathbf{F}(\mathbf{x}^{(k)}; \boldsymbol{\xi}^{(k)}). \quad (17c)$$

## 5 Algorithm analysis

This section presents the convergence analysis of [Algorithm 1](#). The assumptions needed for the analysis are listed in [Section 5.1](#). In particular, we do not assume convexity, and we only have access

to stochastic gradient estimates of each local function  $f_i$ . The convergence results are presented in [Section 5.2](#), with detailed proofs postponed to [Appendix B](#).

### 5.1 Assumptions

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). We start with the following assumption on the problem (1).

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

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

*In addition, the objective function  $f: \mathbb{R}^d \rightarrow \mathbb{R}$  is bounded below, and the optimal value of the 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  $\sigma$ -algebra corresponding to the initial conditions and, for all  $k \in \mathbb{N}_{\geq 1}$ , let  $\mathcal{F}^{(k)}$  denote the  $\sigma$ -algebra defined by the initial conditions and the random variables  $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k)}\}$ , and denote the expectation conditioned on  $\mathcal{F}^{(k)}$  by  $\mathbb{E}_k[\cdot] \triangleq \mathbb{E}[\cdot \mid \mathcal{F}^{(k)}]$ .

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*

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

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

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

### 5.2 Convergence analysis for [Algorithm 1](#)

This section presents the convergence results of [Algorithm 1](#). To leverage the finite-time consensus property, we analyze the evolution of iterates over  $\tau$  iterations and demonstrate that part of the consensus error vanishes after  $\tau$  iterations. To see this, we rewrite the algorithm recursion (17b)–(17c) more compactly as

$$\begin{bmatrix} \mathbf{x}^{(k+1)} \\ \mathbf{g}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{W}^{(k)} & -\alpha \mathbf{W}^{(k)} \\ 0 & \mathbf{W}^{(k)} \end{bmatrix} \begin{bmatrix} \mathbf{x}^{(k)} \\ \mathbf{g}^{(k)} \end{bmatrix} + \begin{bmatrix} 0 \\ \nabla \mathbf{F}^{(k+1)} - \nabla \mathbf{F}^{(k)} \end{bmatrix},$$

where we define  $\nabla \mathbf{F}^{(k)} \triangleq \nabla \mathbf{F}(\mathbf{x}^{(k)}, \xi^{(k)})$  for the ease of presentation. Then, we have

$$\begin{bmatrix} \mathbf{W}^{(i)} & -\alpha \mathbf{W}^{(i)} \\ 0 & \mathbf{W}^{(i)} \end{bmatrix} \cdots \begin{bmatrix} \mathbf{W}^{(j)} & -\alpha \mathbf{W}^{(j)} \\ 0 & \mathbf{W}^{(j)} \end{bmatrix} = \begin{bmatrix} \prod_{k=j}^i \mathbf{W}^{(k)} & -\alpha(i-j+1) \prod_{k=j}^i \mathbf{W}^{(k)} \\ 0 & \prod_{k=j}^i \mathbf{W}^{(k)} \end{bmatrix}.$$

Consequently, the  $\mathbf{x}$ -update (17b) can be written as

$$\mathbf{x}^{(k)} = \left( \prod_{i=k-1}^0 \mathbf{W}^{(i)} \right) \mathbf{x}^{(0)} - \alpha \sum_{j=0}^{k-1} (k-j) \prod_{i=k-1}^j \mathbf{W}^{(i)} (\nabla \mathbf{F}^{(j)} - \nabla \mathbf{F}^{(j-1)}),$$

where for convenience we set  $\nabla \mathbf{F}^{(-1)} = 0$ . Recall the finite-time consensus parameter  $\tau \in \mathbb{N}_{\geq 1}$  from Definition 3.1, and assume  $k-1 \geq \tau$ . Thus, at least one pass through the topology sequence satisfying Definition 3.1 has been performed. Then, we have

$$\begin{aligned} \mathbf{x}^{(k)} &= \left( \prod_{i=k-1}^0 \mathbf{W}^{(i)} \right) \hat{\mathbf{x}}^{(0)} - \alpha \sum_{j=0}^{k-1-\tau} (k-j) \left( \prod_{i=k-1}^j \mathbf{W}^{(i)} \right) (\nabla \mathbf{F}^{(j)} - \nabla \mathbf{F}^{(j-1)}) \\ &\quad - \alpha \sum_{j=k-\tau}^{k-1} (k-j) \left( \prod_{i=k-1}^j \mathbf{W}^{(i)} \right) (\nabla \mathbf{F}^{(j)} - \nabla \mathbf{F}^{(j-1)}). \end{aligned}$$

Multiplying  $\hat{\mathbf{I}} \triangleq I_{dn} - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top \otimes I_d$  on both sides yields

$$\begin{aligned} \hat{\mathbf{x}}^{(k)} &\triangleq \mathbf{x}^{(k)} - \bar{\mathbf{x}}^{(k)} = (I_{dn} - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top \otimes I_d) \mathbf{x}^{(k)} \\ &= \left( \prod_{i=k-1}^0 \widehat{\mathbf{W}}^{(i)} \right) \hat{\mathbf{x}}^{(0)} - \alpha \sum_{j=0}^{k-1-\tau} (k-j) \left( \prod_{i=k-1}^j \widehat{\mathbf{W}}^{(i)} \right) (\nabla \mathbf{F}^{(j)} - \nabla \mathbf{F}^{(j-1)}) \\ &\quad - \alpha \sum_{j=k-\tau}^{k-1} (k-j) \left( \prod_{i=k-1}^j \widehat{\mathbf{W}}^{(i)} \right) (\nabla \mathbf{F}^{(j)} - \nabla \mathbf{F}^{(j-1)}), \end{aligned} \tag{18}$$

where we use the definition of  $\hat{\mathbf{x}}^{(k)}$  in (15) and denote  $\widehat{\mathbf{W}}^{(i)} \triangleq \mathbf{W}^{(i)} - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top \otimes I_d$  for all  $i \in \mathbb{N}$ . Recall that Algorithm 1 cycles through a topology sequence satisfying Definition 3.1 repeatedly in the same order. So the term  $\prod_{i=k-1}^j \widehat{\mathbf{W}}^{(i)}$ , for  $j = 0, 1, \dots, k-1-\tau$  include a products of  $\widehat{\mathbf{W}}^{(i)}$  that consists of a complete cycle of a topology sequence satisfying Definition 3.1. Hence, by (3), the first two terms in (18) are zero and  $\hat{\mathbf{x}}^{(k)}$  in (18) reduces to

$$\hat{\mathbf{x}}^{(k)} = -\alpha \sum_{j=k-\tau}^{k-1} (k-j) \prod_{i=k-1}^j \widehat{\mathbf{W}}^{(i)} (\nabla \mathbf{F}^{(j)} - \nabla \mathbf{F}^{(j-1)}), \tag{19}$$

since by assumption  $k-1 \geq \tau$ .

With the concise presentation of  $\hat{\mathbf{x}}^{(k)}$ , the rest of the analysis is typical in the literature and relies on two important inequalities. The *descent inequality* helps establish the convergence property of the averaged iterates  $\bar{\mathbf{x}}^{(k)}$ . By exploiting the new presentation (19), the *consensus inequality* reveals the per-iteration behavior of the consensus error  $\hat{\mathbf{x}}^{(k)}$ , and will be used to show that each agent's parameter  $x_i^{(k)}$  converges to the average  $\bar{x}^{(k)}$ .

**Lemma 5.1** (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  $\alpha$  satisfy  $\alpha \in (0, \frac{1}{2L}]$ . Then, the sequence  $\{\mathbf{x}^{(k)}\}$  generated by Algorithm 1 satisfies*

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

for all  $k \in \mathbb{N}$ , where  $\tilde{f} \triangleq f - f^*$ .



The left-hand side of the inequality in [Lemma 5.1](#) is the (expected) gradient norm, which aligns with our main convergence result (see [Theorem 5.3](#)). Such a convergence result is common in stochastic unconstrained optimization; see, *e.g.*, [6], 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 5.2** (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  $\alpha \in (0, \frac{1}{4\sqrt{6}\tau^2 L}]$ . Then, for all  $T \in \mathbb{N}_{\geq \tau}$ , the sequence  $\{\mathbf{x}^{(k)}\}$  generated by [Algorithm 1](#) satisfies*

$$\begin{aligned} \frac{1}{T+1} \sum_{k=0}^T \mathbb{E} \|\hat{\mathbf{x}}^{(k)}\|^2 &\leq \frac{2}{T+1} \sum_{k=0}^{\tau} \mathbb{E} \|\hat{\mathbf{x}}^{(k)}\|^2 + \frac{48\alpha^4 \tau^4 n L^2}{T+1} \sum_{k=0}^T \mathbb{E} \|\nabla f(\bar{\mathbf{x}}^{(k)})\|^2 \\ &\quad + (12\alpha^4 \tau^4 L^2 + 16\alpha^2 \tau^3 n) \sigma^2. \end{aligned} \quad (20)$$

Note that the first summation term on the right-hand side of (20) relies on the first  $\tau + 1$  iterates of  $\hat{\mathbf{x}}^{(k)}$ , and recall that  $\tau$  is a prescribed constant for [Algorithm 1](#). Hence, the term  $\sum_{k=0}^{\tau} \mathbb{E} \|\hat{\mathbf{x}}^{(k)}\|^2$  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 to show that the consensus error in the result of [Lemma 5.1](#) vanishes asymptotically. Hence, not only does the averaged parameter asymptotically reach a stationary point of (1) biased by stochastic noise, but all the agents' parameters also converge (because they reach a consensus). [Theorem 5.3](#) formally presents this result.

**Theorem 5.3.** *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 satisfy  $\alpha \in (0, \frac{1}{4\sqrt{6}\tau^2 L}]$ . Then, for all  $T \in \mathbb{N}_{\geq \tau}$ , the sequence  $\{\mathbf{x}^{(k)}\}$  generated by [Algorithm 1](#) satisfies*

$$\frac{1}{T+1} \sum_{k=0}^T \left( \mathbb{E} \|\nabla f(\mathbf{x}^{(k)})\|^2 + \mathbb{E} \|\nabla f(\bar{\mathbf{x}}^{(k)})\|^2 \right) \leq \frac{\gamma_1}{\alpha T} + \frac{\gamma_2 L^2}{n T} + \frac{\gamma_3 \alpha^4 \tau^4 L^4 \sigma^2}{n} + \gamma_4 \alpha^2 \tau^3 L^2 \sigma^2 + \frac{\gamma_5 \alpha L \sigma^2}{n}$$

with some constants  $(\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5) \in \mathbb{R}_{>0}^5$ .

The numeric constants  $(\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5)$ , together with the proof of [Theorem 5.3](#), are provided in [Appendix B](#).

Then, we tune the stepsize  $\alpha$  using the technique described in [15, 25]. Also, similarly to [29], we weaken the rate's dependence on  $L$  by imposing a warm-up strategy (*e.g.*, AllReduce [4]) to force all agents' parameters in the first  $\tau + 1$  iterations to be the same. This leads us to the following corollary.

**Corollary 5.4.** *Let [Assumptions 5.1](#) and [5.2](#) hold, let  $\{W^{(l)}\}_{l=0}^{\tau-1} \subset \mathbb{R}^{n \times n}$  satisfy [Definition 3.1](#), and let*

$$\alpha = \min \left\{ \left( \frac{1}{c_1 T} \right)^{\frac{1}{2}}, \left( \frac{1}{c_2 T} \right)^{\frac{1}{3}}, \frac{1}{4\sqrt{6}\tau^2 L} \right\}, \quad \text{where } c_1 \triangleq \frac{L\sigma^2}{n}, c_2 \triangleq \tau^3 L^2 \sigma^2.$$

*Suppose a warm-up strategy (*e.g.*, AllReduce) is applied so that all agents' parameters and tracking variables in the first period are the same:  $\sum_{k=0}^{\tau} \mathbb{E} \|\hat{\mathbf{x}}^{(k)}\|^2 = 0$ . Then, for all  $T \in \mathbb{N}_{\geq \tau}$ , the sequence*

$\{\mathbf{x}^{(k)}\}$  generated by [Algorithm 1](#) satisfies

$$\frac{1}{T+1} \sum_{k=0}^T \left( \mathbb{E} \|\nabla f(\mathbf{x}^{(k)})\|^2 + \mathbb{E} \|\nabla f(\bar{\mathbf{x}}^{(k)})\|^2 \right) \leq \frac{\gamma_6 \tau^2 L}{T} + \gamma_7 \tau \left( \frac{L\sigma}{T} \right)^{\frac{2}{3}} + \gamma_8 \left( \frac{L\sigma^2}{nT} \right)^{\frac{1}{2}},$$

for some constants  $(\gamma_6, \gamma_7, \gamma_8) \in \mathbb{R}_{>0}^3$ .

### 5.3 Discussion and comparison with other analyses for TV-GT

The derived convergence rate in [Corollary 5.4](#) depends on the finite-time consensus parameter  $\tau$ , and, remarkably, is independent of the connectivity of any of the individual elements of the graph sequence. This contrasts, for example, the rate derived in [\[24\]](#), 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 counterparts of the finite-time consensus graph sequences. Given a sequence of graphs  $\{\mathcal{G}^{(l)} \equiv (\mathcal{V}, W^{(l)}, \mathcal{E}^{(l)})\}_{l=0}^{\tau-1}$ , its static counterpart  $\mathcal{G}^{(\text{static})} = (\mathcal{V}, W^{(\text{static})}, \mathcal{E}^{(\text{static})})$  is defined by  $\mathcal{E}^{(\text{static})} = \mathcal{E}^{(0)} \cup \mathcal{E}^{(1)} \cup \dots \cup \mathcal{E}^{(\tau-1)}$ , 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 [\[9, 13, 22, 29\]](#).) 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  $\tau$  one-peer exponential graphs (satisfying [Definition 3.1](#)) are used in GT-FT. In this case, the best existing rate [\[24\]](#) (to our knowledge) of TV-GT depends on the spectral gap  $1 - \rho = 2/(1 + \tau)$  of the static exponential graph [\[29\]](#), and reads as

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

where the Lipschitz constant  $L$  is omitted in [\[24\]](#). In comparison, it follows from [Corollary 5.4](#) that the iteration complexity of GT-FT using a sequence of  $\tau$  one-peer exponential graphs is given by

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

Ignoring the Lipschitz constants as in [\[24\]](#), we find that this implementation of GT-FT has a similar iteration complexity to 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  $\Theta(\log_2 n)$  while that of a single one-peer exponential graph is  $\Theta(1)$ . Similar comparisons 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

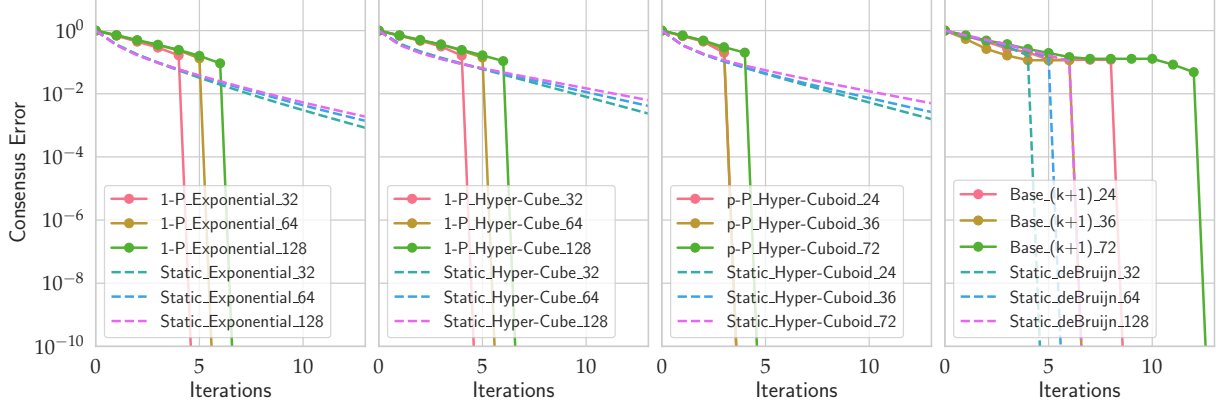


Figure 3: Consensus error versus the number of iterations. The legend is composed of three parts. The first part is either “Static”, “1-P”, or “ $p$ -P”, standing for static graphs, one-peer time-varying graphs, and  $p$ -peer time-varying graphs, respectively. The second part of the legend describes 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 solid lines, while others are plotted with dashed lines. The static variants do not achieve finite-time consensus. The 1-P time-varying graphs only achieve finite-time consensus when the number of agents is a power of two. The  $p$ -P time-varying graphs always achieve finite-time consensus. The sizes of split network components used in Base- $(k+1)$  graphs [27] are  $(16, 8)$  for 24 nodes size,  $(32, 4)$  for 36 nodes, and  $(64, 8)$  for 72 nodes, respectively.

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

We verify in numerical experiments that the presented graph sequences satisfy the finite-time consensus property. 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  $i = 1, \dots, n$ , and the consensus error at each iteration is defined as

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

In the simulation, we compare the presented time-varying graphs with their corresponding static variant, which is defined in Section 5.3.

The simulation results are presented in Figure 3, of which the setting and style closely follow [29, Figure 4]. In Figure 3, the graph sequences satisfying Definition 3.1 have a steep drop in the consensus error (see dashed lines), indicating the vanishing of the consensus error. In comparison, the consensus error for other graphs decreases asymptotically (at an exponential rate).

## 6.2 Gradient tracking with finite-time consensus topologies

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. More specifically, 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  $j$ th component of  $x$ , and the data  $\{A_i, b_i\}$  is held exclusively by agent  $i$ . This problem instance has been used extensively in the literature (see, *e.g.*, [\[2, 28\]](#)), and we follow existing conventions to construct the problem data. 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 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.

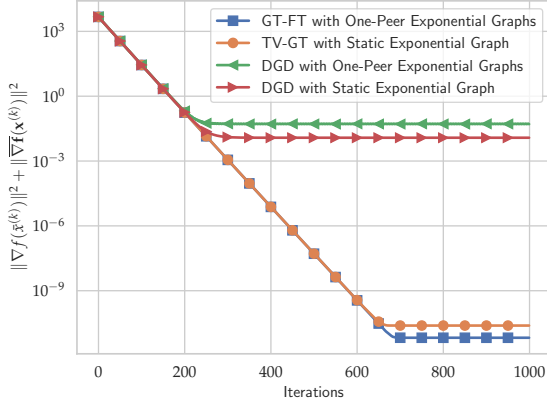
In addition, we consider the case where agents have access to the true gradients and the case where agents only have access to the stochastic gradient estimates. The stochastic gradient is formed by adding Gaussian noise to the true gradient, *i.e.*,  $\widehat{\nabla} f_i(x) = \nabla f_i(x) + s_i$  with  $s_i \sim \mathcal{N}(0, \sigma^2 I_d)$ . The magnitude of the gradient noise can be controlled by the constant  $\sigma^2$ , and we set  $\sigma^2 = 10^{-4}$ . Regarding optimization algorithms, we consider GT-FT ([Algorithm 1](#)) for graph sequences satisfying [Definition 3.1](#), TV-GT for the static counterparts, and DGD for both types of graphs. In all the experiments, the stepsize is set to  $\alpha = 10^{-4}$ .

### 6.2.1 Numerical results

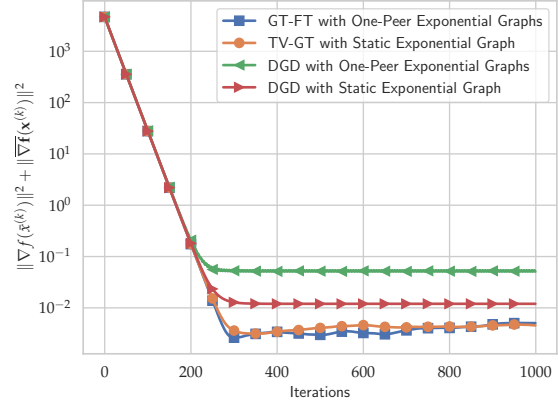
[Figure 4](#) presents the numerical results for one-peer exponential graphs and their static variant, and [Figure 5](#) presents those for  $p$ -peer hyper-cuboids and their static variant. Regardless of whether true gradients or their stochastic estimates are used, the convergence rate of decentralized algorithms using graph sequences with finite-time consensus is similar to that using their static counterparts. In view of this similar rate of convergence, using graph sequences satisfying [Definition 3.1](#) would significantly reduce the communication costs compared with using the static counterpart. For example, the maximum degree of a static exponential graph is  $\Theta(\log n)$  while that of a single one-peer exponential graph is  $\Theta(1)$ .

### 6.2.2 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 has been thoroughly studied in the literature [\[1, 2, 15, 30\]](#). With a constant stepsize, DGD converges to a sub-optimal solution biased proportionally to the magnitude of heterogeneity between agents [\[15, 30\]](#). In comparison, as a gradient tracking method, GT-FT is able to correct this bias caused by heterogeneity, and converges to a better solution [\[1, 2\]](#). However, when agents can only use stochastic gradients, our experiments indicate that the difference in the quality of the solutions returned by

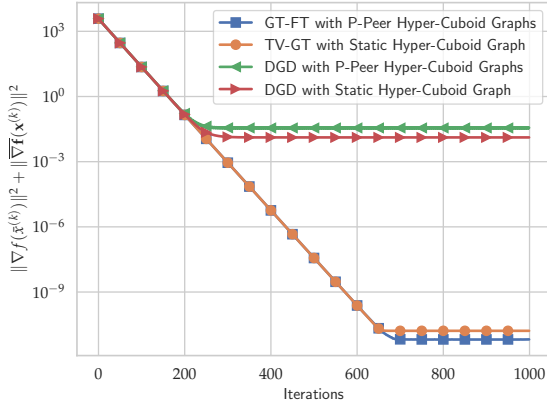


(a) Agents use true gradients.

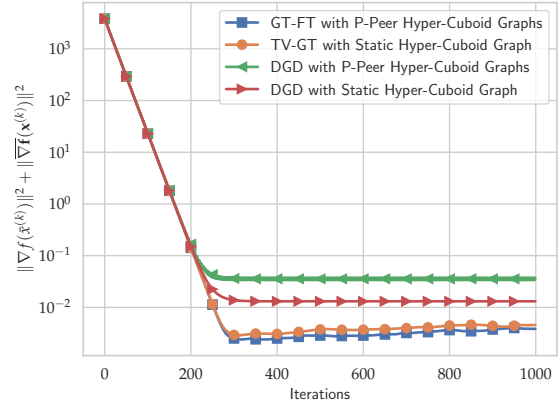


(b) Agents use stochastic gradients.

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



(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.

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 [2].

## 7 Conclusions and future work

We study several graph sequences satisfying 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 one of the few classes 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.

Despite the success of incorporating graph sequences with finite-time consensus in decentralized optimization algorithms, several open questions remain. For example, the incorporation of the proposed graph sequences into other decentralized methods, such as EXTRA [23] and Exact Diffusion [31], is not straightforward. Furthermore, optimal communication costs and convergence rates for distributed algorithms with potentially disconnected time-varying graphs have yet to be established.

## References

- [1] 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.
- [2] 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.
- [3] Sulaiman A. Alghunaim and Kun Yuan. An enhanced gradient-tracking bound for distributed online stochastic convex optimization. *Signal Processing*, 217:109345, 2024.
- [4] 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.
- [5] 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.
- [6] Dimitri P. Bertsekas and John N. Tsitsiklis. Gradient convergence in gradient methods with errors. *SIAM Journal on Optimization*, 10(3):627–642, 2000.

- [7] Federico S. Cattivelli and Ali H. Sayed. Diffusion LMS strategies for distributed estimation. *IEEE Transactions on Signal Processing*, 58(3):1035, 2010.
- [8] Harold Scott Macdonald Coxeter. *Regular Polytopes*. Courier Corporation, 1973.
- [9] 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.
- [10] Jean-Charles Delvenne, Ruggero Carli, and Sandro Zampieri. Optimal strategies in the average consensus problem. *Systems & Control Letters*, 58(10-11):759–765, 2009.
- [11] 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.
- [12] Lisang Ding, Kexin Jin, Bicheng Ying, Kun Yuan, and Wotao Yin. DSGD-CECA: decentralized SGD with communication-optimal exact consensus algorithm. In *International Conference on Machine Learning*, pages 8067–8089. PMLR, 2023.
- [13] 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.
- [14] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2013.
- [15] 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. PMLR, 2020.
- [16] Anastasiia Koloskova, Tao Lin, and Sebastian U. Stich. An improved analysis of gradient tracking for decentralized machine learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 11422–11435, 2021.
- [17] Daniel Krenn, Dimbinaina Ralaivaosona, and Stephan Wagner. Multi-base representations of integers: asymptotic enumeration and central limit theorems. *Applicable Analysis and Discrete Mathematics*, 9(2):285–312, 2015.
- [18] Guanghui Lan, Soomin Lee, and Yi Zhou. Communication-efficient algorithms for decentralized and stochastic optimization. *Mathematical Programming*, 2018.
- [19] 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.
- [20] 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.
- [21] Gesualdo Scutari and Ying Sun. Distributed nonconvex constrained optimization over time-varying digraphs. *Mathematical Programming*, 176(1-2):497–544, 2019.
- [22] 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.



- [23] 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.
- [24] 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. In *Advances in Neural Information Processing Systems*, volume 35, pages 1073–1085, 2022.
- [25] Sebastian U. Stich. Unified optimal analysis of the (stochastic) gradient method. *arXiv preprint arXiv:1907.04232*, 2019.
- [26] 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.
- [27] Yuki Takezawa, Ryoma Sato, Han Bao, Kenta Niwa, and Makoto Yamada. Beyond exponential graph: Communication-efficient topologies for decentralized learning via finite-time convergence. In *Advances in Neural Information Processing Systems*, volume 36, 2024.
- [28] Ran Xin, Usman A. Khan, and Soumya Kar. An improved convergence analysis for decentralized online stochastic non-convex optimization. *IEEE Transactions on Signal Processing*, 69:1842–1858, 2021.
- [29] Bicheng Ying, Kun Yuan, Yiming Chen, Hanbin Hu, Pan Pan, and Wotao Yin. Exponential graph is provably efficient for decentralized deep training. In *Advances in Neural Information Processing Systems*, volume 34, pages 13975–13987, 2021.
- [30] Kun Yuan, Qing Ling, and Wotao Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.
- [31] 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.

## A Supplementary materials for Section 3

We now present the proof of [Proposition 3.5](#), which establishes the connection between de Bruijn graphs and  $p$ -peer hyper-cuboids. The proof relies on the so-called *perfect shuffle matrix* [[14](#), §12.3] and the Kronecker representation of de Bruijn graphs, which, to the best of our knowledge, has not been presented in the literature.

**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  $P_s \in \mathbb{R}^{n \times n}$  such that*

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

Moreover, it holds that  $P_s^\tau = I_n$ .

*Proof.* The first result follows directly from [[14](#), §12.3]. The second result follows by repeatedly applying the first one:

$$P_s^\tau(a_{\tau-1} \otimes \cdots \otimes a_0) = P_s^{\tau-1}(a_0 \otimes a_{\tau-1} \otimes \cdots \otimes a_1) = \cdots = a_{\tau-1} \otimes a_{\tau-2} \otimes \cdots \otimes a_0. \quad \square$$

This permutation matrix  $P_s$  is called the *perfect shuffle matrix* [14, §12.3], hence the subscript “s” in  $P_s$ . Following the same steps as in (11a)–(11c), the matrix form of de Bruijn graphs can be written concisely as

$$W_{\text{db}} = (J_p \otimes I \otimes \cdots \otimes I \otimes I) P_s^\top, \quad \text{where } J_p \triangleq \frac{1}{p} \mathbf{1}_p \mathbf{1}_p^\top. \quad (21)$$

We now present the proof of [Proposition 3.5](#), which states the permutation equivalence between de Bruijn graphs and  $p$ -peer hyper-cuboids. More specifically, with the perfect shuffle matrix  $P_s$ , the connection (13) stated in [Proposition 3.5](#) can be written more explicitly as

$$W_{\text{hc}}^{(\tau-l)} = (P_s^\top)^{l+1} W_{\text{db}} P_s^l, \quad \text{for all } l = 0, 1, \dots, \tau - 1,$$

where  $W_{\text{hc}}$  is the weight matrix of the  $p$ -peer hyper-cuboid (8).

*Proof of [Proposition 3.5](#).* It follows from (9) that

$$W_{\text{hc}}^{(\tau-l)} = \underbrace{I \otimes \cdots \otimes I}_{l-1 \text{ times}} \otimes J_p \otimes I \cdots \otimes I \quad (22a)$$

$$= (P_s^\top)^l (J_p \otimes I \otimes \cdots \otimes I \otimes I) P_s^l \quad (22b)$$

$$= (P_s^\top)^l P_s^\top P_s (J_p \otimes I \otimes \cdots \otimes I \otimes I) P_s^l \quad (22c)$$

$$= P_s^\top (P_s^\top)^l W_{\text{db}} P_s^l. \quad (22d)$$

Step (22a) uses the Kronecker representation (11c). Step (22b) applies the following property of the perfect shuffle matrix  $l$  times to shift  $J_p$  to the beginning:

$$P_s(A_{\tau-1} \otimes A_{\tau-2} \cdots \otimes A_1 \otimes A_0) P_s^\top = A_0 \otimes A_{\tau-1} \otimes \cdots \otimes A_2 \otimes A_1,$$

where  $\{A_i\}_{i=0}^{\tau-1}$  are any arbitrary  $p \times p$  matrices. Step (22c) uses the fact that  $P_s^\top P_s = I$ . Step (22d) uses the definition of de Bruijn graphs in (21).  $\square$

## B Supplementary materials for [Section 5](#)

This section includes the missing proofs from [Section 5](#). First, we introduce the abbreviations

$$\overline{\nabla f}^{(k)} \triangleq \overline{\nabla f}(\mathbf{x}^{(k)}), \quad \nabla \mathbf{f}^{(k)} \triangleq \nabla \mathbf{f}(\mathbf{x}^{(k)}), \quad \nabla \mathbf{F}^{(k)} \triangleq \nabla \mathbf{F}(\mathbf{x}^{(k)}, \boldsymbol{\xi}^{(k)})$$

for all  $k \in \mathbb{N}$ , as well as a list of notations

$$\begin{aligned} \bar{g}^{(k)} &\triangleq \frac{1}{n} \sum_{i=1}^n g_i^{(k)}, & \bar{s}^{(k)} &\triangleq \frac{1}{n} \sum_{i=1}^n (\nabla F_i(\mathbf{x}_i^{(k)}; \boldsymbol{\xi}_i^{(k)}) - \nabla f_i(\mathbf{x}_i^{(k)})), \\ \bar{\mathbf{g}}^{(k)} &\triangleq \mathbf{1}_n \otimes \bar{g}^{(k)}, & \overline{\nabla \mathbf{f}}^{(k)} &\triangleq \overline{\nabla f}^{(k)} \otimes I_d, & \mathbf{s}^{(k)} &\triangleq \nabla \mathbf{F}^{(k)} - \nabla \mathbf{f}^{(k)}. \end{aligned} \quad (23)$$

We also introduce the notation

$$\prod_{k=i}^j \mathbf{W}^{(k)} \triangleq \mathbf{W}^{(i)} \mathbf{W}^{(i-1)} \cdots \mathbf{W}^{(j)}, \quad \text{for all } (i, j) \in \mathbb{N} \times \mathbb{N} \text{ with } i \geq j.$$

## B.1 Proof of Lemma 5.1

The proof of the descent inequality is standard in the literature on GT methods. In particular, it follows from [2, Eq. (73)] that

$$\mathbb{E}f(\bar{x}^{(k+1)}) \leq f(\bar{x}^{(k)}) - \frac{\alpha(1-\alpha L)}{2}\mathbb{E}\|\bar{\nabla}f^{(k)}\|^2 - \frac{\alpha}{2}\mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2 \leq +\frac{\alpha}{2}\|\bar{\nabla}f^{(k)} - \nabla f(\bar{x}^{(k)})\|^2 + \frac{\alpha^2 L \sigma^2}{2n},$$

and from [2, Eq. (66)] that  $\|\bar{\nabla}f^{(k)} - \nabla f(\bar{x}^{(k)})\|^2 \leq \frac{L^2}{n}\|\mathbf{x}^{(k)} - \bar{\mathbf{x}}^{(k)}\|^2 = \frac{L^2}{n}\|\hat{\mathbf{x}}^{(k)}\|^2$ . Combining the above two inequalities yields the desired result.

## B.2 Proof of Lemma 5.2

The proof of the consensus inequality leverages the new presentation of  $\hat{\mathbf{x}}^{(k)}$  (19). More specifically, splitting (19) via the definition (23) gives

$$\begin{aligned} \hat{\mathbf{x}}^{(k)} = & -\alpha \sum_{j=k-\tau}^{k-1} (k-j) \prod_{i=k-1}^j \widehat{\mathbf{W}}^{(i)} (\nabla \mathbf{f}^{(j)} - \nabla \mathbf{f}^{(j-1)}) - \alpha \widehat{\mathbf{W}}^{(k-1)} \mathbf{s}^{(k-1)} \\ & - \alpha \sum_{j=k-\tau}^{k-2} \left( (k-j) \prod_{i=k-1}^j \widehat{\mathbf{W}}^{(i)} - (k-j-1) \prod_{i=k-1}^{j+1} \widehat{\mathbf{W}}^{(i)} \right) \mathbf{s}^{(j)}. \end{aligned} \quad (24)$$

To provide an upper bound on  $\mathbb{E}_k[\|\hat{\mathbf{x}}^{(k)}\|^2]$ , we bound each of the three terms on the right-hand side of (24) one by one. First, it holds that

$$\begin{aligned} & \mathbb{E}_k \left[ \left\| \sum_{j=k-\tau}^{k-1} (k-j) \left( \prod_{i=k-1}^j \widehat{\mathbf{W}}^{(i)} \right) (\nabla \mathbf{f}^{(j)} - \nabla \mathbf{f}^{(j-1)}) \right\|^2 \right] \\ & \leq \tau^3 \sum_{j=k-\tau}^{k-1} \mathbb{E}_k \left[ \left\| \left( \prod_{i=k-1}^j \widehat{\mathbf{W}}^{(i)} \right) (\nabla \mathbf{f}^{(j)} - \nabla \mathbf{f}^{(j-1)}) \right\|^2 \right] \end{aligned} \quad (25a)$$

$$\leq \tau^3 \sum_{j=k-\tau}^{k-1} \mathbb{E}_k [\|\nabla \mathbf{f}^{(j)} - \nabla \mathbf{f}^{(j-1)}\|^2] \quad (25b)$$

$$\leq \tau^3 L^2 \sum_{j=k-\tau}^{k-1} \left( 2\alpha^2 n \mathbb{E}_k [\|\nabla f(\bar{x}^{(j-1)})\|^2] + 9\mathbb{E}_k [\|\hat{\mathbf{x}}^{(j-1)}\|^2] + 3\mathbb{E}_k [\|\hat{\mathbf{x}}^{(j)}\|^2] + 3\alpha^2 \sigma^2 \right). \quad (25c)$$

Note that (25a) uses Jensen's inequality, and (25b) uses the submultiplicative property of matrix norms and the fact that  $\|\widehat{\mathbf{W}}^{(j)}\|_2 \leq 1$ . The last step is standard in the analysis of GT methods (see, e.g., [24, Lemma 11]), and its derivation is omitted due to space limit.

Second, it follows from the fact that  $\|\widehat{\mathbf{W}}^{(k)}\|_2 \leq 1$ , the definition of  $\mathbf{s}^{(k)}$  (23) and Assumption 5.2 that

$$\mathbb{E}_k [\|\widehat{\mathbf{W}}^{(k-1)} \mathbf{s}^{(k-1)}\|^2] \leq n\sigma^2, \quad \text{for all } k \in \mathbb{N}. \quad (26)$$

Third, it holds that

$$\mathbb{E}_k \left[ \left\| \sum_{j=k-\tau}^{k-2} \left( (k-j) \prod_{i=k-1}^j \widehat{\mathbf{W}}^{(i)} - (k-j-1) \prod_{i=k-1}^{j+1} \widehat{\mathbf{W}}^{(i)} \right) \mathbf{s}^{(j)} \right\|^2 \right]$$

$$= \sum_{j=k-\tau}^{k-2} \mathbb{E}_k \left[ \left\| \left( (k-j) \prod_{i=k-1}^j \widehat{\mathbf{W}}^{(i)} - (k-j-1) \prod_{i=k-1}^{j+1} \widehat{\mathbf{W}}^{(i)} \right) \mathbf{s}^{(j)} \right\|^2 \right] \quad (27a)$$

$$\leq 2 \sum_{j=k-\tau}^{k-2} \left( (k-j)^2 \left\| \prod_{i=k-1}^j \widehat{\mathbf{W}}^{(i)} \right\|_2^2 + (k-j-1)^2 \left\| \prod_{i=k-1}^{j+1} \widehat{\mathbf{W}}^{(i)} \right\|_2^2 \right) \mathbb{E}_k [\|\mathbf{s}^{(j)}\|^2] \quad (27b)$$

$$\leq 4\tau^2 \sum_{j=k-\tau}^{k-2} \mathbb{E}_k [\|\mathbf{s}^{(j)}\|^2] \quad (27c)$$

$$\leq 4\tau^2(\tau-1)n\sigma^2. \quad (27d)$$

In (27a), we use the independence of the gradient noise, and (27b) uses the properties of matrix norms. Then, in (27c) we use the fact  $k-j-1 < k-j \leq \tau$  and that  $\|\widehat{\mathbf{W}}^{(i)}\|^2 \leq 1$ , and (27d) applies [Assumption 5.2](#).

Combining (24), (25c), (26), and (27d) yields

$$\begin{aligned} \mathbb{E}_k [\|\hat{\mathbf{x}}^{(k)}\|^2] &\leq 12\alpha^4\tau^3nL^2 \sum_{j=k-\tau}^{k-1} \mathbb{E}_k [\|\nabla f(\bar{x}^{(j-1)})\|^2] + (6\alpha^4\tau^4L^2 + 8\alpha^2\tau^3n)\sigma^2 \\ &\quad + 18\alpha^2\tau^3L^2 \sum_{j=k-\tau}^{k-1} \mathbb{E}_k [\|\hat{\mathbf{x}}^{(j-1)}\|^2] + 6\alpha^2\tau^3L^2 \sum_{j=k-\tau}^{k-1} \mathbb{E}_k [\|\hat{\mathbf{x}}^{(j)}\|^2]. \end{aligned} \quad (28)$$

where we also use Jensen's inequality and the fact that  $\tau \geq 1$ .

Recall our assumption at the beginning of [Appendix B.2](#) that  $k-1 \geq \tau$ . So we sum up (28) over iteration  $k$  from  $\tau+1$  to  $T$  ( $T \geq \tau+1$ ), add  $\sum_{k=0}^{\tau} \mathbb{E}_k [\|\hat{\mathbf{x}}^{(k)}\|^2]$  and divide  $(T+1)$  on both sides:

$$\begin{aligned} &\frac{1}{T+1} \sum_{k=0}^T \mathbb{E}_k [\|\hat{\mathbf{x}}^{(k)}\|^2] \\ &\leq \frac{12\alpha^4\tau^3nL^2}{T+1} \sum_{k=\tau+1}^T \sum_{j=k-\tau-1}^{k-2} \mathbb{E}_j [\|\nabla f(\bar{x}^{(j)})\|^2] + \frac{1}{T+1} \sum_{k=0}^{\tau} \mathbb{E}_k [\|\hat{\mathbf{x}}^{(k)}\|^2] \\ &\quad + \frac{18\alpha^2\tau^3L^2}{T+1} \sum_{k=\tau+1}^T \sum_{j=k-\tau-1}^{k-2} \mathbb{E}_j [\|\hat{\mathbf{x}}^{(j)}\|^2] + \frac{6\alpha^2\tau^3L^2}{T+1} \sum_{k=\tau+1}^T \sum_{j=k-\tau}^{k-1} \mathbb{E}_j [\|\hat{\mathbf{x}}^{(j)}\|^2] \\ &\quad + \frac{(T-\tau+1)}{T+1} (6\alpha^4\tau^4L^2 + 8\alpha^2\tau^3n)\sigma^2. \end{aligned} \quad (29)$$

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  $j = 0, 1, \dots, T$  and

$$\sum_{k=\tau+1}^T \sum_{j=k-\tau-1}^{k-2} \psi_j = \sum_{k=0}^{T-1} \beta_k \psi_k \leq 2\tau \sum_{k=0}^T \psi_k, \quad \sum_{k=\tau+1}^T \sum_{j=k-\tau}^{k-1} \psi_j = \sum_{k=1}^T \beta_k \psi_k \leq 2\tau \sum_{k=0}^T \psi_k. \quad (30)$$

Thus, (29) can be further bounded as

$$\frac{1}{T+1} \sum_{k=0}^T \mathbb{E}_k [\|\hat{\mathbf{x}}^{(k)}\|^2]$$

$$\begin{aligned}
&\leq \frac{24\alpha^4\tau^4nL^2}{T+1} \sum_{k=0}^T \mathbb{E}_k[\|\nabla f(\bar{x}^{(k)})\|^2] + \frac{1}{T+1} \sum_{k=0}^{\tau} \mathbb{E}_k[\|\hat{\mathbf{x}}^{(k)}\|^2] \\
&\quad + \frac{48\alpha^2\tau^4L^2}{T+1} \sum_{k=0}^T \mathbb{E}_k[\|\hat{\mathbf{x}}^{(k)}\|^2] + \frac{(T-\tau+1)}{T+1} (6\alpha^4\tau^4L^2 + 8\alpha^2\tau^3n)\sigma^2, \tag{31}
\end{aligned}$$

where we apply the first inequality in (30) twice (with  $\psi_j \leftarrow \mathbb{E}_j[\|\nabla f(\bar{x}^{(j)})\|^2]$  and with  $\psi_j \leftarrow \mathbb{E}_j[\|\hat{x}^{(j)}\|^2]$ ), and the second inequality in (30) with  $\psi_j \leftarrow \mathbb{E}_j[\|\hat{x}^{(j)}\|^2]$ . Then, the inequality (31) can be written equivalently as

$$\begin{aligned}
\frac{1-48\alpha^2\tau^4L^2}{T+1} \sum_{k=0}^T \mathbb{E}_k[\|\hat{\mathbf{x}}^{(k)}\|^2] &\leq \frac{24\alpha^4\tau^4nL^2}{T+1} \sum_{k=0}^T \mathbb{E}_k[\|\nabla f(\bar{x}^{(k)})\|^2] + \frac{1}{T+1} \sum_{k=0}^{\tau} \mathbb{E}_k[\|\hat{\mathbf{x}}^{(k)}\|^2] \\
&\quad + \frac{T-\tau+1}{T+1} (6\alpha^4\tau^4L^2 + 8\alpha^2\tau^3n)\sigma^2. \tag{32}
\end{aligned}$$

The stepsize condition  $\alpha \leq \frac{1}{4\sqrt{6}\tau^2L}$  implies that  $1-48\alpha^2\tau^4L^2 \geq \frac{1}{2}$ . Substituting this into (32) gives

$$\begin{aligned}
\frac{1}{T+1} \sum_{k=0}^T \mathbb{E}_k[\|\hat{\mathbf{x}}^{(k)}\|^2] &\leq \frac{48\alpha^4\tau^4nL^2}{T+1} \sum_{k=0}^T \mathbb{E}_k[\|\nabla f(\bar{x}^{(k)})\|^2] + \frac{2}{T+1} \sum_{k=0}^{\tau} \mathbb{E}_k[\|\hat{\mathbf{x}}^{(k)}\|^2] \\
&\quad + (1 - \frac{\tau}{T+1})(12\alpha^4\tau^4L^2 + 16\alpha^2\tau^3n)\sigma^2.
\end{aligned}$$

Finally, the desired result (20) follows by taking the total expectation and relaxing  $(1 - \frac{\tau}{T+1})$  to 1.

### B.3 Proof of Theorem 5.3

Lemma 5.1 implies that for all  $k \in \mathbb{N}$ ,

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

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

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

where we use the fact  $\tilde{f}(x) \geq 0$  for any  $x \in \text{dom } f$ . Substituting (20) yields

$$\begin{aligned}
&\frac{1-96\alpha^4\tau^4L^4}{T+1} \sum_{k=0}^T (\mathbb{E}\|\bar{\nabla}f^{(k)}\|^2 + \mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2) \\
&\leq \frac{4}{\alpha(T+1)} \mathbb{E}\tilde{f}(\bar{x}^{(0)}) + \frac{4L^2}{n(T+1)} \sum_{k=0}^{\tau} \mathbb{E}\|\hat{\mathbf{x}}^{(k)}\|^2 + \left( \frac{24\alpha^4\tau^4L^4}{n} + 32\alpha^2\tau^3L^2 + \frac{2\alpha L}{n} \right) \sigma^2.
\end{aligned}$$

The stepsize condition  $\alpha \leq \frac{1}{4\sqrt{6}\tau^2L}$  implies  $96\alpha^4L^4\tau^4 \leq \frac{1}{2}$ , and thus we have

$$\frac{1}{T+1} \sum_{k=0}^T (\mathbb{E}\|\bar{\nabla}f^{(k)}\|^2 + \mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2)$$

$$\leq \frac{8}{\alpha(T+1)} \mathbb{E} \tilde{f}(\bar{x}^{(0)}) + \frac{8L^2}{n(T+1)} \sum_{k=0}^{\tau} \mathbb{E} \|\hat{\mathbf{x}}^{(k)}\|^2 + \left( \frac{48\alpha^4\tau^4L^4}{n} + 64\alpha^2\tau^3L^2 + \frac{4\alpha L}{n} \right) \sigma^2. \quad (33)$$

The desired result follows after further relaxing  $\frac{1}{T+1}$  to  $\frac{1}{T}$  and defining the constants  $\gamma_1 = 8, \gamma_2 = 8 \sum_{k=0}^{\tau} \mathbb{E} \|\hat{\mathbf{x}}^{(k)}\|^2, \gamma_3 = 48, \gamma_4 = 64, \gamma_5 = 4$ . (Note that  $\gamma_2$  consists of only the first  $\tau$  iterates, which can be safely viewed as a constant in an asymptotic analysis.)

#### B.4 Proof of Corollary 5.4

To simplify the notation, from now on, we use the notation  $\lesssim$  to hide irrelevant constants. The notation  $a \lesssim b$  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  $\alpha, n, L$ , and  $\sigma$ . For the statement of the corollary, we define the constants  $\gamma_6 = 8, \gamma_7 = 64, \gamma_8 = 4$ . Using this notation and the AllReduce warm-up strategy that guarantees  $\sum_{k=0}^{\tau} (\mathbb{E} \|\hat{\mathbf{x}}^{(k)}\|^2) = 0$ , we obtain from (33) and the stepsize condition  $\alpha \lesssim \frac{1}{\tau^2 L}$  that

$$\begin{aligned} & \frac{1}{T+1} \sum_{k=0}^T (\mathbb{E} \|\bar{\nabla} f^{(k)}\|^2 + \mathbb{E} \|\nabla f(\bar{x}^{(k)})\|^2) \\ & \lesssim \frac{1}{\alpha T} + \frac{\alpha^4 \tau^4 L^4 \sigma^2}{n} + \alpha^2 \tau^3 L^2 \sigma^2 + \frac{\alpha L \sigma^2}{n} \\ & \lesssim \frac{1}{\alpha T} + \frac{\alpha^2 L^2 \sigma^2}{n} + \alpha^2 \tau^3 L^2 \sigma^2 + \frac{\alpha L \sigma^2}{n} \\ & \lesssim \frac{1}{\alpha T} + \alpha^2 \tau^3 L^2 \sigma^2 + \frac{\alpha L \sigma^2}{n} \\ & \lesssim \frac{1}{\alpha T} + c_1 \alpha + c_2 \alpha^2, \end{aligned}$$

where  $c_1 \triangleq \frac{L\sigma^2}{n}$  and  $c_2 \triangleq \tau^3 L^2 \sigma^2$ . Now we set the stepsize  $\alpha$  as

$$\alpha = \min \left\{ \left( \frac{c_0}{c_1 T} \right)^{\frac{1}{2}}, \left( \frac{c_0}{c_2 T} \right)^{\frac{1}{3}}, \frac{1}{4\sqrt{6}\tau^2 L} \right\}.$$

By definition, this choice of  $\alpha$  satisfies the stepsize condition in Theorem 5.3. We then discuss the following three cases.

- (a) If  $\alpha = \frac{1}{4\sqrt{6}\tau^2 L} \leq \min\left\{\left(\frac{1}{c_1 T}\right)^{\frac{1}{2}}, \left(\frac{1}{c_2 T}\right)^{\frac{1}{3}}\right\}$ , then  $\frac{1}{\alpha T} + \alpha c_1 + c_2 \alpha^2 \lesssim \frac{1}{\alpha T} + \left(\frac{c_1}{T}\right)^{\frac{1}{2}} + \frac{c_2^{\frac{1}{3}}}{T^{\frac{1}{3}}}$ .
- (b) If  $\alpha = \left(\frac{1}{c_1 T}\right)^{\frac{1}{2}} \leq \left(\frac{1}{c_2 T}\right)^{\frac{1}{3}}$ , then  $\frac{1}{\alpha T} + c_1 \alpha + c_2 \alpha^2 \lesssim \frac{c_1^{\frac{1}{2}}}{T^{\frac{1}{2}}} + \frac{c_2^{\frac{1}{3}}}{T^{\frac{1}{3}}}$ .
- (c) If  $\alpha = \left(\frac{1}{c_2 T}\right)^{\frac{1}{3}} \leq \left(\frac{1}{c_1 T}\right)^{\frac{1}{2}}$ , then  $\frac{1}{\alpha T} + c_1 \alpha + c_2 \alpha^2 \lesssim \frac{c_2^{\frac{1}{3}}}{T^{\frac{1}{3}}} + \left(\frac{c_1}{T}\right)^{\frac{1}{2}}$ .

Combining all three cases yields

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