Communication-Efficient Topologies for Decentralized Learning with $\mathcal{O}(1)$ Consensus Rate

Anonymous Author(s)
Affiliation
Address
email

Abstract

Decentralized optimization is an emerging paradigm in distributed learning in which agents achieve network-wide solutions by peer-to-peer communication without the central server. Since communication tends to be slower than computation, when each agent communicates with only a few neighboring agents per iteration, they can complete iterations faster than with more agents or a central server. However, the total number of iterations to reach a network-wide solution is affected by the speed at which the information of the agents is "mixed" by communication. We found that popular communication topologies either have large degrees (such as stars and complete graphs) or are ineffective at mixing information (such as rings and grids). To address this problem, we propose a new family of topologies, EquiTopo, which has an (almost) constant degree and a network-size-independent consensus rate, which is used to measure the mixing efficiency.

In the proposed family, EquiStatic has a degree of $\Theta(\ln(n))$, where n is the network size, and a series of time-dependent one-peer topologies, EquiDyn, has a constant degree of 1. We generate EquiDyn through a certain random sampling procedure. Both of them achieve an n-independent consensus rate. We apply them to decentralized SGD and decentralized gradient tracking and obtain faster communication and better convergence, both theoretically and empirically.

1 Introduction

Modern optimization and machine learning typically involve tremendous data samples and model parameters. The scale of these problems calls for efficient distributed algorithms across multiple computing nodes. Traditional distributed approaches usually follow a centralized setup, where each node needs to communicate with a (virtually) central server. This communication pattern incurs significant communication overheads and long latency.

Decentralized learning is an emerging paradigm to save communications in large-scale optimization and learning. In decentralized learning, all computing nodes are connected with some network topology (e.g., ring, grid, hypercube, etc.) in which each node averages/communicates locally with its immediate neighbors. This decentralized setup allows each node to communicate with fewer neighbors and hence has a much lower overhead in per-iteration communication. However, local averaging is less effective in "mixing" information, making decentralized algorithms converge slower than their centralized counterparts. Therefore, it is critical to seek a balance between communication efficiency and convergence rate in decentralized learning.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Table 1: Comparison between different commonly-used topologies. "Static Exp.": static exponential graph; "O.-P. Exp.": one-peer exponential graph; "E.-R. Rand": Erdos-Renyi random graph $G(n,p)$ with probability $p = (1+a)\ln(n)/n$ for some $a > 0$; "Geo. Rand": geometric random graph $G(n,r)$ with radius $r^2 = (1+a)\ln(n)/n$ for some $a > 0$. Undirected graphs can admit symmetric gossip matrices. If some graph has a dynamic pattern, its associated gossip matrix will vary at each iteration.

Topology	**Connection**	**Pattern**	**Degree**	**Consensus Rate**	**size** n
Ring [21]	undirect.	static	$\Theta(1)$	$1-\Theta(1/n^2)$	arbitrary
Grid [21]	undirect.	static	$\Theta(1)$	$1-\Theta(1/(n\ln(n)))$	arbitrary
Torus [21]	undirect.	static	$\Theta(1)$	$1-\Theta(1/n)$	arbitrary
Hypercube [30]	undirect.	static	$\Theta(\ln(n))$	$1-\Theta(1/\ln(n))$	power of 2
Static Exp.[36]	directed	static	$\Theta(\ln(n))$	$1-\Theta(1/\ln(n))$	arbitrary
O.-P. Exp.[36]	directed	dynamic	1	finite-time conv.†	power of 2
E.-R. Rand [3]	undirect.	static	$\Theta(\ln(n))$	$1-\Theta(1/\ln(n))$	arbitrary
Geo. Rand [5]	undirect.	static	$\Theta(\ln(n))$	$1-\Theta(\ln(n)/n)$	arbitrary
D-EquiStatic	directed	static	$\Theta(\ln(n))$	$\rho \in (0,1)^\ddagger$	arbitrary
U-EquiStatic	undirect.	static	$\Theta(\ln(n))$	$\rho \in (0,1)^\ddagger$	arbitrary
OD-EquiDyn	directed	dynamic	1	$\sqrt{(1+\rho)/2}$	arbitrary
OU-EquiDyn	undirect.	dynamic	1	$\sqrt{(2+\rho)/3}$	arbitrary

† One-peer exponential graph has finite-time exact convergence only when n is the power of 2.
‡ Constant $\rho = \Theta(1)$ is independent of network-size n.

The network topology (or graph) determines decentralized algorithms' per-iteration communication and convergence rate. The maximum graph degree controls the communication cost, whereas the connectivity influences the convergence rate. Intuitively speaking, a densely-connected topology enables decentralized methods to converge faster but results in less efficient communication since each node needs to average with more neighbors. Selecting an appropriate network topology is key to achieving both light communication and fast convergence in decentralized learning.

1.1 Prior arts in topology selections

Gossip matrix and consensus rate. Given a connected network of size n and its associated doubly-stochastic gossip matrix $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ (see the definition in § 2), its consensus rate β determines how effective the gossip operation $\boldsymbol{W}\boldsymbol{x}$ is to mix information (see more explanations in § 2). It is a long-standing topic in decentralized learning to seek topologies with both a small maximum degree and a fast consensus rate (i.e., a small β as close to 0 as possible).

Static graphs. Static topologies maintain the same graph connections throughout all iterations. The directedness, degree, and consensus rate of various common topologies are summarized in Table 1. The ring, grid, and torus graphs [21] are the simplest sparse topologies with $\Theta(1)$ maximum degree. However, their consensus rates quickly approach 1 as network size n increases, which leads to an inefficient local averaging. The hypercube graph [30] maintains a nice balance between degree and consensus rate since $\ln(n)$ varies slowly with n. However, this graph cannot be formed when size n is not the power of 2. The static exponential graph extends hypercubes to graphs with any size n, but its directed communications cannot enable symmetric gossip matrices required in well-known decentralized algorithms such as EXTRA [27], Exact-Diffusion [38], NIDS [16], D^2 [29]. Two widely-used random topologies, i.e., the Erdos-Renyi graph [19, 3] and the geometric random graph [4, 5], are also listed in Table 1. While they are more robust to node or link failures, their maximum degree and consensus rate are not better than deterministic hypercube or exponential graphs.

Dynamic graphs. Dynamic graphs allow time-varying topologies between iterations. When an exponential graph allows each node to cycle through all its neighbors and communicate only to a single node per iteration, we achieve the time-varying one-peer exponential graph [2, 36]. When the network size n is the power of 2, a sequence of one-peer exponential graphs can together achieve periodic global averaging. However, its consensus rate is unknown for other values of n. A closely related work Matcha [33] proposed a disjoint matching decomposition sampling strategy when

training learning models. While it decomposes a static dense graph into a series of sparse graphs with small degrees, the consensus rates of these dynamic graphs are not established.

Finally, it is worth noting that the consensus rates of all graphs discussed above are either unknown or dependent on size n. Their efficiency to mix information gets less effective as n goes large.

1.2 Main results

Motivation. Since existing network topologies suffer from several limitations, we ask the following questions. *Can we develop topologies that have (almost) constant degrees and network-size-independent consensus rates that admit both symmetric and asymmetric matrices of any size? Can these topologies allow one-peer dynamic variants?* This paper provides affirmative answers.

Main results and contributions. This paper develops several novel graphs that are built upon a set of basis graphs in which the label difference between any pair of connected nodes are *equivalent*. With a general name EquiTopo, these new graphs can achieve network-size-independent consensus rates while maintaining (almost) constant graph degrees. Our contributions are:

- We construct a directed graph named D-EquiStatic that has a network-size-independent consensus rate ρ with degree $\Theta(\ln(n))$. Furthermore, we develop an one-peer time-varying variant named OD-EquiDyn to achieve a network-size-independent consensus rate with degree 1.
- We construct a undirected graph U-EquiStatic, which has a network-size-independent consensus rate ρ with degree $\Theta(\ln(n))$. It admits symmetric gossip matrices that are required by various important algorithms. We also develop an one-peer time-varying and undirected variant named OU-EquiDyn to achieve a network-size-independent consensus rate with degree 1.
- We apply the EquiTopo graphs to two well-known decentralized algorithms, i.e., decentralized stochastic gradient descent (SGD) [6, 17, 12] and stochastic gradient tracking (SGT) [22, 35, 7, 34], to achieve the state-of-the-art convergence rate while maintaining $\Theta(\ln(n))$ (with D/U-EquiStatic) or 1 (with OD/OU-EquiDyn) degree in per-iteration communication.

The comparison between EquiTopo and other common topologies in Table 1 shows that the EquiTopo family (especially the one-peer variants) has achieved the best balance between graph degree and consensus rate. The comparison between EquiTopo and other common topologies when applying to DSGD and DSGT are listed in Tables 3 and 4 in Appendix C.2 and Table 2.

Note. This paper considers scenarios in which any two nodes can be connected when necessary. The high-performance data-center cluster is one such scenario, in which all GPUs are connected with high-bandwidth channels and the network topology can be fully controlled. EquiTopo may not be applied to wireless network settings where two remote nodes cannot be connected directly.

1.3 Other related works

In decentralized optimization, decentralized gradient descent [23, 6, 25] and dual averaging [8] are well-known approaches. While simple and widely used, their solutions are sensitive to heterogeneous data distributions. Advanced algorithms that can overcome this drawback include explicit bias-correction [27, 38, 16], gradient tracking [22, 7, 24, 35], and dual acceleration [26, 32]. Decentralize SGD is extensively studied in [6, 17, 2] to solve stochastic problems. It has been extended to directed [2, 36] or time-varying topologies [12, 20, 33], asynchronous settings [18], and data-heterogeneous scenarios [29, 34, 1, 11] to achieve better performances.

2 Notations and Preliminaries

Notations. We let $\mathbb{1}_n \in \mathbb{R}^n$ be the all-ones vector and $\boldsymbol{I} \in \mathbb{R}^{n\times n}$ be the identity matrix. Furthermore, we define $\boldsymbol{J} = \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T$ and $\boldsymbol{\Pi} = \boldsymbol{I} - \boldsymbol{J}$. A matrix $\boldsymbol{A} = [a_{ij}] \in \mathbb{R}^{n\times n}$ is nonnegative if $a_{ij} \geq 0$ for all $1 \leq i, j \leq n$. A nonnegative matrix $\boldsymbol{A}$ is doubly stochastic if $\boldsymbol{A}\mathbb{1}_n = \boldsymbol{A}^T\mathbb{1}_n = \mathbb{1}_n$. Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\|\boldsymbol{A}\|_2$ is its spectral norm. For $\boldsymbol{x} \in \mathbb{R}^n$, $\|\boldsymbol{x}\|$ is its Euclidean norm. We let

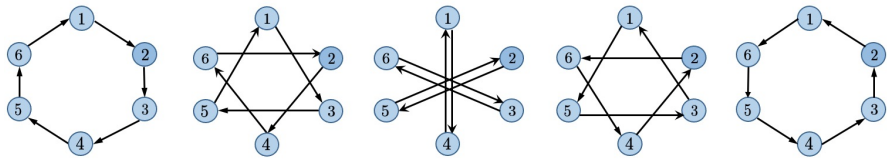

Figure 1: The set of the basis graphs $\{\mathcal{G}(\boldsymbol{A}^{(u)})\}_{u=1}^{5}$ for $n = 6$.

$[n] = \{1, \cdots, n\}$. Throughout the paper, we define a mod operation that returns value in $[n]$ as

$$i \bmod n = \begin{cases} \ell & \text{if } i = kn + \ell \text{ for some } k \in \mathbb{Z} \text{ and } \ell \in [n-1], \\ n & \text{if } i = kn \text{ for some } k \in \mathbb{Z}. \end{cases} \tag{1}$$

Network. Given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with a set of n nodes $\mathcal{V}$ and a set of directed edges $\mathcal{E}$. An edge $(j, i) \in \mathcal{E}$ means node j can directly send information to node i. For undirected graphs, $(j, i) \in \mathcal{E}$ if and only if $(i, j) \in \mathcal{E}$. Node i's degree is the number of its in-neighbors $|\{j|(j, i) \in \mathcal{E}\}|$. A graph is one-peer means that the degree for each node is at most 1.

Weight matrices. To facilitate the local averaging step in decentralized algorithms, each graph is associated with a nonnegative weight matrix $\boldsymbol{W} = [w_{ij}] \in \mathbb{R}^{n \times n}$, whose element w_{ij} is non-zero only if $(j, i) \in \mathcal{E}$ or $i = j$. One benefit of a undirected graph is that it can be associated with a symmetric matrix. Given a nonegative weight matrix $\boldsymbol{W} \in \mathbb{R}^{n \times n}$, we let $\mathcal{G}(\boldsymbol{W})$ be its associated graph such that $(j, i) \in \mathcal{E}$ if $w_{ij} > 0$ and $i \neq j$.

Consensus rate. For weight matrices $\{\boldsymbol{W}^{(t)}\}_{t \geq 0} \subseteq \mathbb{R}^{n \times n}$, the consensus rate β is the minimum nonnegative number such that for any $t \geq 0$ and vector $\boldsymbol{x} \in \mathbb{R}^n$ with the average $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$,

$$\mathbb{E}\left[\left\| \boldsymbol{W}^{(t)} \boldsymbol{x} - \bar{x} \cdot \mathbb{1}_n \right\|^2\right] \leq \beta^2 \left\| \boldsymbol{x} - \bar{x} \cdot \mathbb{1}_n \right\|^2,$$

or equivalently, $\mathbb{E}\left[\|\boldsymbol{\Pi}\, \boldsymbol{W}^{(t)} \boldsymbol{x}\|^2\right] \leq \beta^2 \|\boldsymbol{\Pi} \boldsymbol{x}\|^2$. For $\boldsymbol{W}^{(t)} \equiv \boldsymbol{W}$, β essentially equals $\|\boldsymbol{\Pi}\, \boldsymbol{W}\|_2$.

3 Directed EquiTopo Graphs

3.1 Basis weight matrices and basis graphs

Given a graph of size n, we introduce a set of doubly stochastic *basis matrices* $\{\boldsymbol{A}^{(u,n)}\}_{u=1}^{n-1}$, where $\boldsymbol{A}^{(u,n)} = [a_{ij}^{(u,n)}] \in \mathbb{R}^{n \times n}$ with

$$a_{ij}^{(u,n)} = \begin{cases} \frac{n-1}{n}, & \text{if } i = (j + u) \mod n, \\ \frac{1}{n}, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Their associated graphs $\{\mathcal{G}(\boldsymbol{A}^{(u,n)})\}_{u=1}^{n-1}$ are called *basis graphs*. A basis graph $\mathcal{G}(\boldsymbol{A}^{(u,n)})$ has degree one and the same *label difference* $(i - j) \bmod n$ for all edges (j, i). The set of five basis graphs $\{\mathcal{G}(\boldsymbol{A}^{(u,6)})\}_{u=1}^{5}$ for $n = 6$ is shown in Fig. 1. When n is clear from the context, we omit it and write $\boldsymbol{A}^{(u)}$ instead.

3.2 Directed static EquiTopo graphs (D-EquiStatic)

Our directed graphs are built on the above basis graphs, and a weight matrix has the form

$$\boldsymbol{W} = \frac{1}{M} \sum_{i=1}^{M} \boldsymbol{A}^{(u_i)}, \tag{3}$$

where $u_i \in [n-1]$. Throughout this paper, the multiset $\{u_i\}_{i=1}^{M}$ are called *basis index*. It is possible that $u_i = u_j$ for some $i \neq j$. Since each $\boldsymbol{A}^{(u)}$ has the form (2), the matrix $\boldsymbol{W}$ is doubly stochastic, and all nodes of the directed graph $\mathcal{G}(\boldsymbol{W})$ have the same degree that is no more than M.

Since $\mathcal{G}(\boldsymbol{W})$ is a directed static graph and built with M basis graphs, we name it D-EquiStatic. The following theorem shows that we can construct a weight matrix $\boldsymbol{W}$ such that its consensus rate is independent of the network size n by setting M properly. The proofs of all theorems are in the Appendix.

Theorem 1 *Let $\boldsymbol{A}^{(u)}$ be defined by* (2) *for any $u \in [n-1]$. For any $\rho \in (0,1)$, we can choose a sequence of $u_1, \cdots, u_M$ from $[n-1]$ with $M = \Theta\left(\ln(n)/\rho^2\right)$ and construct the D-EquiStatic weight matrix $\boldsymbol{W}$ as in* (3) *such that the consensus rate of $\boldsymbol{W}$ is ρ, i.e.,*

$$\|\boldsymbol{\Pi}\,\boldsymbol{W}\boldsymbol{x}\| \leq \rho\,\|\boldsymbol{\Pi}\boldsymbol{x}\|\,,\ \forall \boldsymbol{x} \in \mathbb{R}^n \tag{4}$$

The graph $\mathcal{G}(\boldsymbol{W})$ has degree at most M. In the following, we will just say that the degree is $\Theta(\ln(n))$. As ρ is tunable, we choose ρ as a constant, e.g., $\rho = 0.5$. A method of constructing D-EquiStatic weight matrix $\boldsymbol{W}$ can be found in Appendix A.1.

Remark 1 *Compared to all common topologies listed in Table 1, D-EquiStatic achieves a better balance between degree and consensus rate. Moreover, D-EquiStatic works for any size $n \geq 2$. Different from the Erdos-Renyi random graph and the geometric random graph, whose degree cannot be predefined before the implementation, we can easily specify the degree M for D-EquiStatic.*

3.3 One-peer directed EquiTopo graphs (OD-EquiDyn)

While D-EquiStatic achieves a size-independent consensus rate with $\Theta(\ln(n))$ degree, we develop a one-peer dynamic variant to further reduce its degree. Given a weight matrix $\boldsymbol{W}$ of form (3) and its associated basis matrix $\{\boldsymbol{A}^{(u_i)}\}_{i=1}^{M}$, the one-peer directed variant, or OD-EquiDyn for short, samples a random $\boldsymbol{A}^{(u)}$ per iteration and utilizes it as the one-peer weight matrix, see Alg. 1. Since each node in $\mathcal{G}(\boldsymbol{A}^{(u)})$ has exactly one neighbor, the graph $\mathcal{G}(\boldsymbol{W}^{(t)})$ has degree one for every iteration t. Note that $\boldsymbol{W}^{(t)}$ is a random time-varying weight matrix. Its consensus rate (in expectation) can be characterized as below.

Algorithm 1: OD-EquiDyn weight matrix generation at iteration t

Input: constant $\eta \in (0,1)$; basis index $\{u_1, u_2, \ldots, u_M\}$ from a weight matrix $\boldsymbol{W}$ of form (3);
Pick v_t from uniform distribution over the basis index $\{u_1, u_2, \ldots, u_M\}$;
Produce basis matrix $\boldsymbol{A}^{(v_t)}$ according to (2);
Output: $\boldsymbol{W}^{(t)} = (1-\eta)\boldsymbol{I} + \eta\boldsymbol{A}^{(v_t)}$

Theorem 2 *Let the one-peer directed weight matrix $\boldsymbol{W}^{(t)}$ be generated by Alg. 1. It holds that*

$$\mathbb{E}\Big[\left\|\boldsymbol{\Pi}\,\boldsymbol{W}^{(t)}\boldsymbol{x}\right\|^2\Big] \leq \big(1 - 2\eta(1-\eta)(1-\rho)\big)\,\|\boldsymbol{\Pi}\boldsymbol{x}\|^2\,,\ \ \forall \boldsymbol{x} \in \mathbb{R}^n$$

where ρ is the consensus rate of the weight matrix $\boldsymbol{W}$.

Remark 2 *The OD-EquiDyn graph has a degree of* 1 *no matter how dense the input matrix $\boldsymbol{W}$ is. When $\eta = 1/2$, which is used in our implementations, it holds that $\mathbb{E}\|\boldsymbol{\Pi}\,\boldsymbol{W}^{(t)}\boldsymbol{x}\|^2 \leq (1+\rho)/2\,\|\boldsymbol{\Pi}\boldsymbol{x}\|^2$ for any $\boldsymbol{x} \in \mathbb{R}^n$. Thus, the OD-EquiDyn graph maintains the same $\Theta(1)$ degree as ring, grid, and the one-peer exponential graph but with a faster size-independent consensus rate.*

Remark 3 *For basis index $\{1, \cdots, n-1\}$ ($\boldsymbol{W} = \boldsymbol{J}$), Alg. 1 returns a sequence of OD-EquiDyn graphs with $\mathbb{E}\|\boldsymbol{\Pi}\,\boldsymbol{W}^{(t)}\boldsymbol{x}\|^2 \leq \frac{1}{2}\,\|\boldsymbol{\Pi}\boldsymbol{x}\|^2$ when $\eta = \frac{1}{2}$ because $\|\boldsymbol{\Pi}\boldsymbol{J}\boldsymbol{x}\| = 0$ implies $\rho = 0$.*

4 Undirected EquiTopo Graphs

The implementation of many important algorithms such as EXTRA [27], Exact-Diffusion [38], NIDS [16], decentralized ADMM [28], and the dual-based optimal algorithms [26, 32, 13] rely on symmetric weight matrices. Moreover, devices in full-duplex communication systems can communicate with one another in both directions, and undirected networks are natural to be utilized. These motivate us to study undirected graphs.

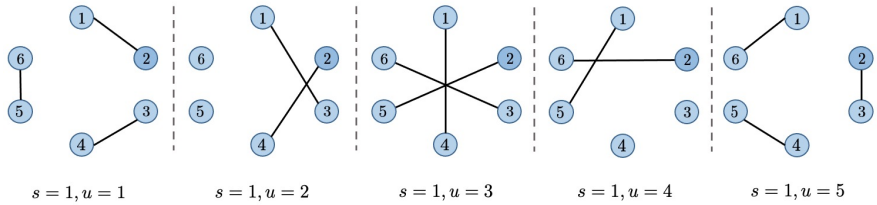

Figure 2: A few realizations of the OU-EquiDyn graphs for $n = 6$, $s = 1$, and $u \in \{1, 2, 3, 4, 5\}$.

4.1 Undirected static EquiTopo graphs (U-EquiStatic)

Given a D-EquiStatic weight matrix $\boldsymbol{W}$ and its associated basis matrices $\{\boldsymbol{A}^{(u_i)}\}_{i=1}^M$, we directly construct an undirected weight matrix name U-EquiStatic by

$$\widetilde{\boldsymbol{W}} = \tfrac{1}{2}(\boldsymbol{W} + \boldsymbol{W}^T) = \tfrac{1}{2M}\sum_{i=1}^M (\boldsymbol{A}^{(u_i)} + [\boldsymbol{A}^{(u_i)}]^T), \tag{5}$$

whose basis index are $\{u_i, -u_i\}_{i=1}^M$ because $\boldsymbol{A}^{(-u)} = [\boldsymbol{A}^{(u)}]^T$.

Since $\widetilde{\boldsymbol{W}}$ is built upon $\boldsymbol{W}$ and $\boldsymbol{\Pi}\,\boldsymbol{W} = \boldsymbol{W}\boldsymbol{\Pi}$, the following theorem follows directly from

$$\left\|\boldsymbol{\Pi}\,\widetilde{\boldsymbol{W}}\right\|_2 = \frac{1}{2}\left\|\boldsymbol{\Pi}\,\boldsymbol{W} + (\boldsymbol{\Pi}\,\boldsymbol{W})^T\right\|_2 \leq \frac{1}{2}\left(\|\boldsymbol{\Pi}\,\boldsymbol{W}\|_2 + \left\|(\boldsymbol{\Pi}\,\boldsymbol{W})^T\right\|_2\right) = \|\boldsymbol{\Pi}\,\boldsymbol{W}\|_2\,.$$

Theorem 3 *Let $\boldsymbol{W}$ be a D-EquiStatic matrix with consensus rate ρ and $\widetilde{\boldsymbol{W}}$ be the U-EquiStatic matrix defined by* (5)*. It holds that*

$$\left\|\boldsymbol{\Pi}\,\widetilde{\boldsymbol{W}}\boldsymbol{x}\right\| \leq \rho\,\|\boldsymbol{\Pi}\boldsymbol{x}\|\,, \quad \forall \boldsymbol{x} \in \mathbb{R}^n. \tag{6}$$

4.2 One-peer undirected EquiTopo graphs (OU-EquiDyn)

Constructing an one-peer undirected graph OU-EquiDyn is not as direct as U-EquiStatic because $\frac{1}{2}(\boldsymbol{A}^{(u)} + [\boldsymbol{A}^{(u)}]^T)$ admits a graph with degree 2, see Appendix B.1 for an illustration.

Alg. 2 shows a method to construct a series of OU-EquiDyn matrices with degree 1. Starting from node s, we connect a node with the uth node after it, as long as both of them have not been connected to any other nodes. Fig. 2 illustrates the process when $s = 1$ and $u = 1, \cdots, 5$ for a network of size 6. Some nodes have no neighbors at realizations $u = 2$ and $u = 4$. This phenomenon is caused by the restriction that each node has no more than one neighbors. E.g., when $u = 2$, node 5 wants to connect with node 1 but node 1 has already been connected to node 3. Thus, there exist node pairs that are never connected when $s = 1$. To resolve this issue, we let the starting index s be sampled randomly from $[n]$. Fig. 8 in Appendix B.1 illustrates the scenarios when $s = 3$. It is observed that the pairs $\{3, 5\}$ and $\{4, 6\}$ are now connected to each other. The node version of OU-EquiDyn is illustrated in Alg. 4 of Appendix. B.2.

Algorithm 2: OU-EquiDyn weight matrix generation at iteration t

Input: $\eta \in (0, 1)$; basis index $\{u_i, -u_i\}_{i=1}^M$ from a symmetric weight matrix $\widetilde{\boldsymbol{W}} \in \mathbb{R}^{n\times n}$ of form (5);

Pick $v_t \in \{u_i, -u_i\}_{i=1}^M$ and $s_t \in [n]$ uniformly at random;

Initialize $\boldsymbol{A} = [a_{ij}] = \boldsymbol{I}$ and $b_i = 0, \forall i \in [n]$;

for $j = (s_t : s_t + n - 1 \mod n)$ **do**

  $i = (j + v_t) \mod n$;

  if $b_i = 0$ *and* $b_j = 0$ **then**

    $a_{ij} = a_{ji} = (n-1)/n$;

    $a_{ii} = a_{jj} = 1/n$;

    $b_i = 1, b_j = 1$;

  end

end

Output: $\widetilde{\boldsymbol{W}}^{(t)} = (1-\eta)I + \eta\boldsymbol{A}$

Theorem 4 *Let $\widetilde{\boldsymbol{W}}$ be a U-EquiStatic matrix with consensus rate ρ, and $\widetilde{\boldsymbol{W}}^{(t)}$ be an OU-EquiDyn matrix generated by Alg. 2, it holds that*

$$\mathbb{E}\left[\left\|\boldsymbol{\Pi}\,\widetilde{\boldsymbol{W}}^{(t)}\boldsymbol{x}\right\|^2\right] \leq \left(1 - \frac{4}{3}\eta(1-\eta)(1-\rho)\right)\|\boldsymbol{\Pi}\boldsymbol{x}\|^2\,, \quad \forall \boldsymbol{x} \in \mathbb{R}^n.$$

Remark 4 *Theorem 4 implies that the OU-EquiDyn graph can achieve a size-independent consensus rate with degree at most* 1*. When* $\eta = 1/2$*, it holds that* $\mathbb{E}\|\mathbf{\Pi}\, \boldsymbol{W}^{(t)}\boldsymbol{x}\|^2 \leq [(2+\rho)/3]\|\mathbf{\Pi}\boldsymbol{x}\|^2$.

When $\widetilde{\boldsymbol{W}} = \boldsymbol{J}$ and the basis index $\{1, \cdots, n-1\}$ are input to Alg. 2, we obtain an OU-EquiDyn sequence $\widetilde{\boldsymbol{W}}^{(t)}$ such that $\mathbb{E}\|\mathbf{\Pi}\, \widetilde{\boldsymbol{W}}^{(t)}\boldsymbol{x}\|^2 \leq (2/3)\|\mathbf{\Pi}\boldsymbol{x}\|^2$.

Remark 5 *An alternative OU-EquiDyn matrix construction that relies on the Euclidean algorithm is in Appendix B.4.*

5 Applying EquiTopo Matrices to Decentralized Learning

We consider the following distributed problem over a network of n computing nodes:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \quad f(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}) \tag{7}$$

where $f_i(\boldsymbol{x}) := \mathbb{E}_{\xi_i\sim\mathcal{D}_i}\big[F(\boldsymbol{x};\xi_i)\big]$. The function $f_i(\boldsymbol{x})$ is kept at node i, and ξ_i denotes the local data that follows the local distribution $\mathcal{D}_i$. Data heterogeneity exists if local distributions $\{\mathcal{D}_i\}_{i=1}^n$ are not identical. Throughout this section, we let $\boldsymbol{x}_i^{(t)}$ be node i's local model at iteration t, and $\bar{\boldsymbol{x}}^{(t)} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^{(t)}$.

Assumptions. We make the following standard assumptions to facilitate analysis.

A.1 *Each local cost function* $f_i(x)$ *is differentiable, and there exists a constant* $L > 0$ *such that* $\|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})\| \leq L\,\|\boldsymbol{x}-\boldsymbol{y}\|$ *for all* $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$.

A.2 *Let* $\boldsymbol{g}_i^{(t)} = \nabla F(\boldsymbol{x}_i^{(t)};\xi_i^{(t)})$*. There exists* $\sigma^2 > 0$ *such that for any* t *and* i

$$\mathbb{E}_{\xi_i^{(t)}\sim\mathcal{D}_i}\boldsymbol{g}_i^{(t)} = \nabla f_i(\boldsymbol{x}_i^{(t)}), \;\text{ and }\; \mathbb{E}_{\xi_i^{(t)}\sim\mathcal{D}_i}\Big[\Big\|\boldsymbol{g}_i^{(t)} - \nabla f_i(\boldsymbol{x}_i^{(t)})\Big\|^2\Big] \leq \sigma^2.$$

A.3 *(For DSGD only) There exists* b^2 *such that* $\frac{1}{n}\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\| \leq b^2$ *for all* $\boldsymbol{x} \in \mathbb{R}^d$.

5.1 Decentralized stochastic gradient descent

The decentralized stochastic gradient descent (DSGD) [6, 17, 12] is given by

$$\boldsymbol{x}_i^{(t+1)} = \textstyle\sum_{j=1}^n w_{ij}^{(t)}\big(\boldsymbol{x}_j^{(t)} - \gamma\boldsymbol{g}_j^{(t)}\big), \tag{8}$$

where the weight matrix $\boldsymbol{W}^{(t)} = [w_{ij}^{(t)}]$ can be time-varying and random. Applying the EquiTopo matrices discussed in § 3-4, we achieve the following convergence results, whose proof follows Theorem 2 in [12] directly, and is omitted here. More results for DSGD are given in Appendix C.1.

Theorem 5 *Consider the DSGD algorithm* (8)*. Under Assumptions A.1-A.3, it holds that*

$$\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}\Big[\Big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\Big\|^2\Big] = \mathcal{O}\Big(\frac{\sigma}{\sqrt{nT}} + \frac{\beta^{\frac{2}{3}}\sigma^{\frac{2}{3}}}{T^{\frac{2}{3}}(1-\beta)^{\frac{1}{3}}} + \frac{\beta^{\frac{2}{3}}b^{\frac{2}{3}}}{T^{\frac{2}{3}}(1-\beta)^{\frac{2}{3}}} + \frac{\beta}{T(1-\beta)}\Big),$$

- *where* $\beta = \rho$ *with D-EquiStatic* $\boldsymbol{W}$ *or U-EquiStatic* $\widetilde{\boldsymbol{W}}$*;*
- *where* $\beta = \sqrt{(1+\rho)/2}$ *for OD-EquiDyn* $\boldsymbol{W}^{(t)}$ *(Alg. 1 with* $\eta = 1/2$*), and* $\beta = \sqrt{(2+\rho)/3}$ *for OU-EquiDyn* $\widetilde{\boldsymbol{W}}^{(t)}$ *(Alg. 2 with* $\eta = 1/2$*).*

For a sufficiently large T, the term $\mathcal{O}(1/\sqrt{nT})$ dominates the rate, and we say the algorithm reaches the linear speedup stage. The transient iterations are referred to as those iterations before an algorithm reaches the linear-speedup stage. We compare the per-iteration communication, convergence rate, and transient iterations of DSGD over various topologies in Tables 3 and 4 of the Appendix. It is observed that OD/OU-EquiDyn endows DSGD with the lightest communication, fastest convergence rate, and smallest transient iteration complexity.

Table 2: Per-iteration communication and computation complexity of the DSGT under different topologies.

Topology	Per-iter Comm.	Convergence Rate	Trans. Iters.
Ring	$\Theta(1)$	$\mathcal{O}\left(\frac{\sigma}{\sqrt{nT}} + \frac{n^2\sigma^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{n^4}{T}\right)$	$\mathcal{O}(n^{15})$
Torus	$\Theta(1)$	$\mathcal{O}\left(\frac{\sigma}{\sqrt{nT}} + \frac{n\sigma^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{n^2}{T}\right)$	$\mathcal{O}(n^9)$
Static Exp.	$\Theta(\ln(n))$	$\mathcal{O}\left(\frac{\sigma}{\sqrt{nT}} + \frac{\ln(n)\sigma^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{\ln^2(n)}{T}\right)$	$\mathcal{O}(n^3\ln^6(n))$
O.-P. Exp.	1	$\mathcal{O}\left(\frac{\sigma}{\sqrt{nT}} + \frac{\ln(n)\sigma^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{\ln^2(n)}{T}\right)$	$\mathcal{O}(n^3\ln^6(n))$
D(U)-EquiStatic	$\Theta(\ln(n))$	$\mathcal{O}\left(\frac{\sigma}{\sqrt{nT}} + \left(\frac{\sigma}{T}\right)^{\frac{2}{3}} + \frac{1}{T}\right)$	$\mathcal{O}(n^3)$
OD (OU)-EquiDyn	1	$\mathcal{O}\left(\frac{\sigma}{\sqrt{nT}} + \left(\frac{\sigma}{T}\right)^{\frac{2}{3}} + \frac{1}{T}\right)$	$\mathcal{O}(n^3)$

5.2 Decentralized stochastic gradient tracking algorithm

The decentralized stochastic gradient tracking algorithm (DSGT) [22, 7, 24, 35, 34] is given by

$$\begin{aligned} \boldsymbol{x}_i^{(t+1)} &= \textstyle\sum_{j=1}^n w_{ij}^{(t)}\big(\boldsymbol{x}_j^{(t)} - \gamma \boldsymbol{y}_j^{(t)}\big); \\ \boldsymbol{y}_i^{(t+1)} &= \textstyle\sum_{j=1}^n w_{ij}^{(t)}\boldsymbol{y}_j^{(t)} + \boldsymbol{g}_i^{(t+1)} - \boldsymbol{g}_i^{(t)}, \quad \boldsymbol{y}_i^{(0)} = \boldsymbol{g}_i^{(0)}. \end{aligned} \tag{9}$$

The following result of DSGT does not appear in literature since it admits an improved convergence rate for stochastic decentralized optimization over asymmetric or time-varying weight matrices. Existing works on DSGT assume weight matrix to be either symmetric [11, 1] or static [34].

Theorem 6 *Consider the DSGT algorithm in* (9)*. If* $\{\boldsymbol{W}^{(t)}\}_{t\geq 0}$ *have consensus rate* β*, then under Assumptions A.1-A.2, it holds for* $T \geq \frac{1}{1-\beta}$ *that*

$$\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}\left[\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\|^2\right] = \mathcal{O}\Big(\frac{\sigma}{\sqrt{nT}} + \frac{\sigma^{\frac{2}{3}}}{(1-\beta)T^{\frac{2}{3}}} + \frac{1}{(1-\beta)^2 T}\Big).$$

When utilizing the EquiTopo matrices, the corresponding β is specified in Theorem 5. Note that DSGT achieves linear speedup for large T. The per-iteration communication and convergence rate comparison of DSGT over different topologies is in Table 2. OD/OU-EquiDyn endows DSGT with the lightest communication, fastest convergence rate, and smallest transient iteration complexity.

6 Numerical Experiments

This section presents experimental results to validate EquiTopo's network-size-independent consensus rate and its comparison with other commonly-used topologies in DSGD on both strongly-convex problems and non-convex deep learning tasks. More experiments for EquiTopo in DSGT, as well as all implementation details, are referred to Appendix D.

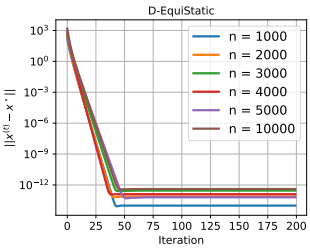

Figure 3: The D-EquiStatic topology can achieve network-size independent consensus rate.

Network-size independent consensus rate. This simulation examines the consensus rates of all four EquiTopo graphs. We recursively run the gossip averaging $\boldsymbol{x}^{(t+1)} = \boldsymbol{W}^{(t)}\boldsymbol{x}^{(t)}$ with $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ initialized arbitrarily and $\boldsymbol{W}^{(t)} \in \mathbb{R}^{n\times n}$ generated as D-EquiStatic (Eq. (3)), OD-EquiDyn (Alg. 1), U-EquiStatic (Eq. (5)), and OU-EquiDyn (Alg. 2), respectively. Fig. 3 depicts how the quantity $\|\boldsymbol{x}^{(t)} - \boldsymbol{J}\boldsymbol{x}^{(0)}\|$ evolves when n ranges from 1000 to 10,000 with D-EquiStatic topology. See Appendix D for all

other EquiTopo graphs which also achieve network-size independent consensus rates when n varies. These results are consistent with Thms. 1 - 4.

Comparison with other topologies. We now compare EquiTopo's consensus rate with other commonly-used topologies. Fig. 4 illustrates the performance of several graphs with $\Theta(1)$ degree when running gossip averaging. We set $n = 4900$ so that the grid graph can be organized as 70×70. OD/OU-EquiDyn are much faster than other topologies. Note that each node in OD-EquiDyn, OU-EquiDyn, and O.-P. Exp. has exactly one neighbor per iteration. More experiments on graphs with $\Theta(\ln(n))$ degrees and on scenarios with smaller network sizes are in Appendix D.

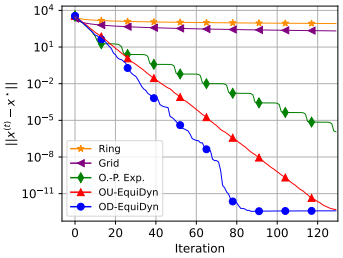

Figure 4: OD/OU-EquiDyn is faster than other topologies with $\Theta(1)$ degree in consensus rate.

DSGD with EquiTopo: least-square. We next apply D/U-EquiStatic graphs to DSGD when solving the distributed least square problems. In the experiment, we let $n = 300$ and set $M = 9$ so that D/U-EquiStatic has the same degree as the exponential graph. Fig. 5 depicts that D/U-EquiStatic converges much faster than static exponential graph especially in the initial stages when learning rate is large. The U-EquiStatic has a slightly better performance than D-EquiStatic since its bi-directional communication enables the graph with better connectivity.

DSGD with EquiTopo: deep learning. We consider the image classification task with ResNet-20 model [9] over the CIFAR-10 dataset [14]. We utilize BlueFog [37] to support decentralized communication and topology setting in a cluster of 17 Tesla P100 GPUs. Fig. 6 illustrates how D/U-EquiStatic compare with static exp., ring, and centralized SGD in training loss and test accuracy. It is observed that D/U-EquiStatic have strong performance. They achieve competitive training losses to centralized SGD but with a slightly better test accuracy. Meanwhile, they also outperform static exponential graph in test accuracy by a visible margin (D-EquiStatic: 92%, U-EquiStatic: 91.7%, Static Exp.: 91.5%). Experiments with EquiDyn topologies and results on MNIST dataset [15] are in Appendix D.

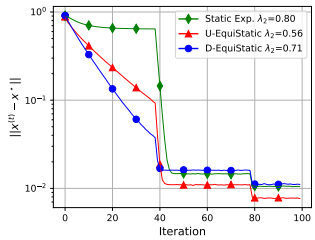

Figure 5: D/U-EquiStatic in DSGD. λ_2 is the second largest eigenvalue.

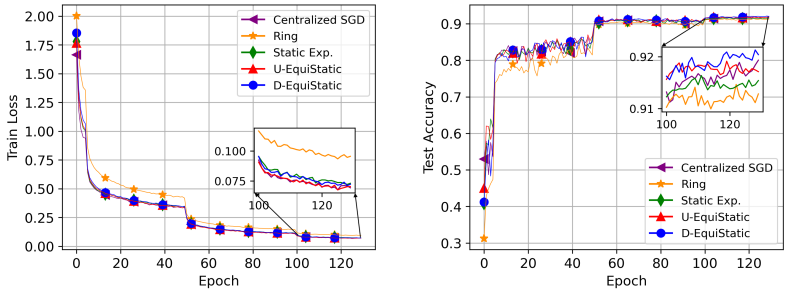

Figure 6: Train loss and test accuracy comparisons among different topologies for ResNet-20 on CIFAR-10.

7 Conclusion

This paper proposes EquiTopo graphs that achieve a state-of-the-art balance between maximum degree and consensus rate. The EquiStatic graphs are with $\Theta(\ln(n))$ degrees and n-independent consensus rates, while their one-peer variants, EquiDyn, maintain roughly the same consensus rates with degree at most 1. EquiTopo enables decentralized learning with light communication and fast convergence. We are not aware of any **negative social impacts** in our results to our best knowledge.

References

[1] Sulaiman A Alghunaim and Kun Yuan. A unified and refined convergence analysis for non-convex decentralized learning. *arXiv preprint arXiv:2110.09993*, 2021.

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

[3] Itai Benjamini, Gady Kozma, and Nicholas Wormald. The mixing time of the giant component of a random graph. *Random Structures & Algorithms*, 45(3):383–407, 2014.

[4] Andrew Beveridge and Jeanmarie Youngblood. The best mixing time for random walks on trees. *Graphs and Combinatorics*, 32(6):2211–2239, 2016.

[5] Stephen P Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Mixing times for random walks on geometric random graphs. In *ALENEX/ANALCO*, pages 240–249, 2005.

[6] Jianshu Chen and Ali H Sayed. Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Transactions on Signal Processing*, 60(8):4289–4305, 2012.

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

[8] John C Duchi, Alekh Agarwal, and Martin J Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3):592–606, 2011.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[10] Yerlan Idelbayev. Proper ResNet implementation for CIFAR10/CIFAR100 in PyTorch. `https://github.com/akamaster/pytorch_resnet_cifar10`. Accessed: 2022-05.

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

[12] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning (ICML)*, pages 1–12, 2020.

[13] Dmitry Kovalev, Egor Shulgin, Peter Richtárik, Alexander V Rogozin, and Alexander Gasnikov. Adom: accelerated decentralized optimization method for time-varying networks. In *International Conference on Machine Learning*, pages 5784–5793. PMLR, 2021.

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[15] Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[16] Z. Li, W. Shi, and M. Yan. A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *IEEE Transactions on Signal Processing*, July 2019. early acces. Also available on arXiv:1704.07807.

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

[18] Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. In *International Conference on Machine Learning (ICML)*, pages 3043–3052, 2018.

[19] Asaf Nachmias and Yuval Peres. Critical random graphs: diameter and mixing time. *The Annals of Probability*, 36(4):1267–1286, 2008.

[20] Angelia Nedić and Alex Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2014.

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

[22] A. Nedic, A. Olshevsky, and W. Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017.

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

[24] G. Qu and N. Li. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3):1245–1260, 2018.

[25] Ali H Sayed. Adaptive networks. *Proceedings of the IEEE*, 102(4):460–497, 2014.

[26] Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *International Conference on Machine Learning (ICML)*, pages 3027–3036, 2017.

[27] 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.

[28] Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. On the linear convergence of the admm in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 62(7):1750–1761, 2014.

[29] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. d^2: Decentralized training over decentralized data. In *International Conference on Machine Learning*, pages 4848–4856, 2018.

[30] Luca Trevisan. Lecture notes on graph partitioning, expanders and spectral methods. *University of California, Berkeley, https://people. eecs. berkeley. edu/˜ luca/books/expanders-2016. pdf*, 2017.

[31] Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.

[32] César A Uribe, Soomin Lee, Alexander Gasnikov, and Angelia Nedić. A dual approach for optimal algorithms in distributed optimization over networks. *Optimization Methods and Software*, pages 1–40, 2020.

[33] Jianyu Wang, Anit Kumar Sahu, Zhouyi Yang, Gauri Joshi, and Soummya Kar. MATCHA: Speeding up decentralized SGD via matching decomposition sampling. *arXiv preprint arXiv:1905.09435*, 2019.

[34] Ran Xin, Usman A Khan, and Soummya Kar. An improved convergence analysis for decentralized online stochastic non-convex optimization. *IEEE Transactions on Signal Processing*, 2020.

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

[36] Bicheng Ying, Kun Yuan, Yiming Chen, Hanbin Hu, Pan Pan, and Wotao Yin. Exponential graph is provably efficient for decentralized deep training. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

[37] Bicheng Ying, Kun Yuan, Hanbin Hu, Yiming Chen, and Wotao Yin. Bluefog: Make decentralized algorithms practical for optimization and deep learning. *arXiv preprint arXiv:2111.04287*, 2021.

[38] Kun Yuan, Bicheng Ying, Xiaochuan Zhao, and Ali H. Sayed. Exact dffusion for distributed optimization and learning – Part I: Algorithm development. *IEEE Transactions on Signal Processing*, 67(3):708 – 723, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] At the end of Section 1.2
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] They are in the Appendix (Supplemental material).
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] They are included in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] The details are in the Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We have repeated some experiments with different seeds and reported the averaged performance. See the Appendix for details.
   (d) Did you include the total amount of computer and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]