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

Zhuoqing Song[1,*], **Weijian Li**[2*], **Kexin Jin**[3*], **Lei Shi**[1,7], **Ming Yan**[4,5], **Wotao Yin**[2], **Kun Yuan**[2,6]

[1]Fudan University [2]Alibaba DAMO Academy [3]Princeton University
[4]The Chinese University of Hong Kong, Shenzhen [5]Michigan State University
[6]Peking University [7]Shanghai Artificial Intelligence Laboratory

zqsong19@fudan.edu.cn, weijian.li@alibaba-inc.com, kexinj@math.princeton.edu
yanming@cuhk.edu.cn, leishi@fudan.edu.cn
wotao.yin@alibaba-inc.com, kun.yuan@alibaba-inc.com

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 agents' information is "mixed" by communication. We found that popular communication topologies either have large maximum 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 that 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, theoretically and empirically. Our code is implemented through BlueFog and available at `https://github.com/kexinjinnn/EquiTopo`.

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

[*]Equal Contribution. Corresponding Author: Kun Yuan

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

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 [22]	undirect.	static	$\Theta(1)$	$1-\Theta(1/n^2)$	arbitrary
Grid [22]	undirect.	static	$\Theta(1)$	$1-\Theta(1/(n\ln(n)))$	arbitrary
Torus [22]	undirect.	static	$\Theta(1)$	$1-\Theta(1/n)$	arbitrary
Hypercube [31]	undirect.	static	$\Theta(\ln(n))$	$1-\Theta(1/\ln(n))$	power of 2
Static Exp.[37]	directed	static	$\Theta(\ln(n))$	$1-\Theta(1/\ln(n))$	arbitrary
O.-P. Exp.[37]	directed	dynamic	1	finite-time conv.†	power of 2
E.-R. Rand [22]	undirect.	static	$\Theta(\ln(n))^\diamond$	$\Theta(1)$	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.
$^\diamond$ $\Theta(\ln(n))$ is the averaged degree; its maximum degree can be $O(n)$ with a non-zero probability.
‡ Constant $\rho = \Theta(1)$ is independent of network-size n.

than their centralized counterparts. Therefore, seeking a balance between communication efficiency and convergence rate in decentralized learning is critical.

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 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 [22] 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 inefficient local averaging. The hypercube graph [31] 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 [28], Exact-Diffusion [40], NIDS [16], D^2 [30]. Two widely-used random topologies, i.e., the Erdos-Renyi graph [20, 3] and the geometric random graph [4, 5], are also listed in Table 1. It is observed that the Erdos-Renyi graph achieves a network-size-independent consensus rate with a $\Theta(\ln(n))$ *averaged* degree in expectation. However, it is worth noting that the communication overhead in network topology is determined by the *maximum* degree. Since some nodes may have much more neighbors than others in a random realization, the maximum degree in the Erdos-Renyi graph can be $O(n)$ with a non-zero probability. Moreover, the random graphs listed in Table 1 are undirected. The may not be used in scenarios where directed graphs are preferred.

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, 37]. 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 [34] 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 (except for the Erdos-Renyi graph) discussed above are either unknown or dependent on size n. Their efficiency in mixing 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 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 a 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) [23, 36, 7, 35], 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 maximum 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 [24, 6, 26, 39] 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 [28, 40, 16], gradient tracking [23, 7, 25, 36], and dual acceleration [27, 33]. Decentralize SGD is extensively studied in [6, 17, 2] to solve stochastic problems. It has been extended to directed [2, 37] or time-varying topologies [12, 21, 34, 37], asynchronous settings [18], and data-heterogeneous scenarios [30, 35, 1, 11, 19] to achieve better performances.

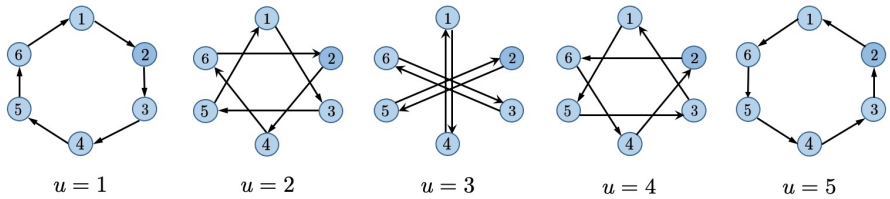

Figure 1: The set of the basis graphs $\{\mathcal{G}(\boldsymbol{A}^{(u)})\}_{u=1}^{5}$ for $n = 6$.

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 $[n] = \{1, \cdots, n\}$. Throughout the paper, we define a mod operation that returns a 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 one-peer graph 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 an undirected graph is that it can be associated with a symmetric matrix. Given a nonnegative 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 $u_i \in [n-1]$ and $M > 0$ is the number of basis graphs we will sample. 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 constant $\rho \in (0,1)$, 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 $\boldsymbol{W}$ (which can be tuned freely as in Theorem 1).

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 $\rho = 0$.*

Remark 4 *Although the Erdos-Renyi random graph also enjoys $\Theta(1)$ consensus rate and $\Theta(\ln(n))$ average degree (see Table 1), its maximum degree could be as large as $\Theta(n)$ which implies that*

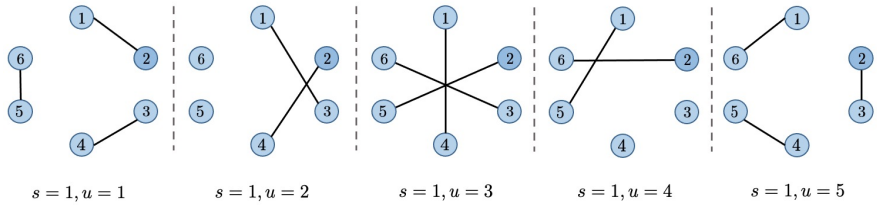

Figure 2: A few realizations of the OU-EquiDyn graphs for $n = 6$, $s = 1$, and $u \in \{1, 2, 3, 4, 5\}$.

Erdos-Renyi random graphs could be highly unbalanced. Moreover, Erdos-Renyi random graphs are undirected graphs, while EquiStatic graphs can be both directed (Section 3) and undirected (Section 4). In addition, the structure of EquiStatic allows simple construction of one-peer random graphs which preserve $\Theta(1)$ consensus, while it is still an open problem on whether Erdos-Renyi random graphs admit one-peer variants with $\Theta(1)$ consensus rate.

4 Undirected EquiTopo Graphs

The implementation of many important algorithms such as EXTRA [28], Exact-Diffusion [40], NIDS [16], decentralized ADMM [29], and the dual-based optimal algorithms [27, 33, 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.

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 a 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 neighbor. For instance, 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*

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

Remark 5 *Theorem 4 implies that the OU-EquiDyn graph can achieve a size-independent consensus rate with a degree at most* 1*. When $\eta = 1/2$, it holds that $\mathbb{E}\|\boldsymbol{\Pi}\,\boldsymbol{W}^{(t)}\boldsymbol{x}\|^2 \leq [(2+\rho)/3]\|\boldsymbol{\Pi}\boldsymbol{x}\|^2$.*

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 $\mathbb{E}\|\boldsymbol{\Pi}\,\widetilde{\boldsymbol{W}}^{(t)}\boldsymbol{x}\|^2 \leq (2/3)\|\boldsymbol{\Pi}\boldsymbol{x}\|^2$.

Remark 6 *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:

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

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*

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

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.

5.2 Decentralized stochastic gradient tracking algorithm

The decentralized stochastic gradient tracking algorithm (DSGT) [23, 7, 25, 36, 35] is given by

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

The following result of DSGT does not appear in the 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 [35].

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 and all implementation details are referred to Appendix D.

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 Theorems 1 - 4.

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	$\mathcal{O}\left(\frac{\sigma}{\sqrt{nT}} + \left(\frac{\sigma}{T}\right)^{\frac{2}{3}} + \frac{1}{T}\right)$	$\mathcal{O}(n^3)$

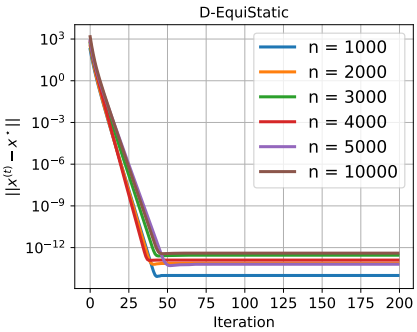

Figure 3: The D-EquiStatic topology can achieve network-size independent consensus rate.

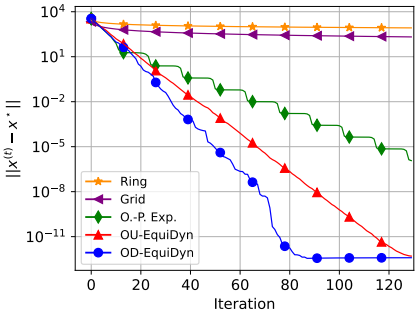

Figure 4: OD/OU-EquiDyn is faster than other topologies (i.e., ring, grid, and one-peer exponential graph) with $\Theta(1)$ degree in consensus rate.

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 is 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.

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 a static exponential graph, especially in the initial stages when the learning rate is large. The U-EquiStatic performs slightly better 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 [38] to support decentralized communication and topology setting in a cluster of 17 Tesla P100 GPUs. Fig. 6 illustrates how D/U-EquiStatic compares with static exp., ring, and centralized SGD in training loss and test accuracy. It is observed that D/U-EquiStatic has strong performance. They achieve competitive training losses to centralized SGD but slightly better test accuracy. Meanwhile, they also outperform static exponential graphs 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.

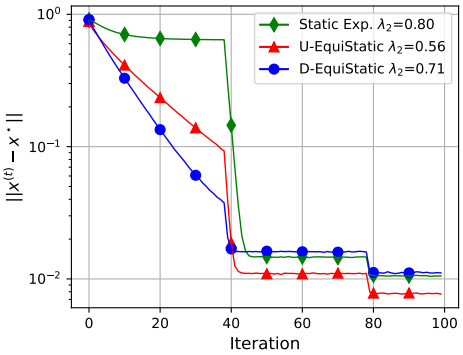

Figure 5: D/U-EquiStatic in DSGD. λ_2 is the second largest eigenvalue.

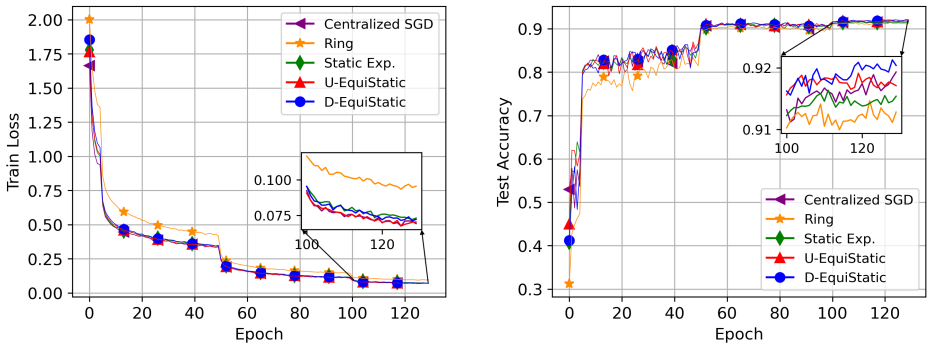

Figure 6: Train loss and test accuracy comparisons among different topologies for ResNet-20 on CIFAR-10.

7 Conclusion

This paper proposes EquiTopo graphs that achieve a state-of-the-art balance between the 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 a degree at most 1. EquiTopo enables decentralized learning with light communication and fast convergence. To the best of our knowledge, we are not aware of any **negative social impacts** in our results.

Acknowledgement

We thank the anonymous reviewers for suggesting updated literature on E.-R. graphs. Ming Yan was partially supported by the NSF award DMS-2012439. Lei Shi was partially supported by Shanghai Science and Technology Program under Project No. 21JC1400600 and No. 20JC1412700 and National Natural Science Foundation of China (NSFC) under Grant No. 12171093. Kexin Jin was supported by Alibaba Research Internship Program.

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] Songtao Lu, Xinwei Zhang, Haoran Sun, and Mingyi Hong. Gnsd: A gradient-tracking based nonconvex stochastic algorithm for decentralized optimization. In *2019 IEEE Data Science Workshop (DSW)*, pages 315–321. IEEE, 2019.

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

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

[22] 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.

[23] 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.

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

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

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

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

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

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

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

[40] 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]

A Directed EquiTopo Graphs

A.1 Construction of a D-EquiStatic graph

A practical method to construct a D-EquiStatic weight matrix $\boldsymbol{W}$ is provided in Alg. 3. We should mention that the "while" loop in the algorithm is adopted to guarantee $\|\boldsymbol{\Pi}\,\boldsymbol{W}\|_2 \leq \rho$.

Algorithm 3: A practical method for D-EquiStatic weight matrix generation

Input: Network size n; desired consensus rate $\rho \in (0,1)$; probability p

Set $M = \left\lceil \frac{8}{3\rho^2}\ln(2n/p)\right\rceil$ and initialize $\boldsymbol{W} = \boldsymbol{I}$;

while $\|\boldsymbol{\Pi}\,\boldsymbol{W}\|_2 > \rho$ **do**

  Sample M i.i.d random variables $u_1, u_2, \ldots, u_M$ uniformly from $[n-1]$;

  Generate basis weight matrices $\{A^{(u_i)}\}_{i=1}^M$ according to (2);

  Construct $\boldsymbol{W}$ by (3);

end

Output: The D-EquiStatic weight matrix $\boldsymbol{W}$ and its associated basis indices $\{u_t\}_{t=1}^M$

A.2 Proof of Theorem 1

Before showing properties of $\boldsymbol{W}$ defined by (3), we provide two lemmas as follows.

Referring to Theorem 1.6 of [32], we have the following result for a sequence of random matrices.

Lemma 1 (Matrix Bernstein) *Consider a sequence of K independent random $n \times n$ matrices $\{\boldsymbol{M}_i\}_{i=1}^K$. Assume that each random matrix satisfies*

$$\mathbb{E}[\boldsymbol{M}_i] = 0, \quad \text{and} \quad \|\boldsymbol{M}_i\|_2 \leq R \ \text{ almost surely.}$$

Define

$$\sigma^2 := \max\Big\{\Big\|\sum\nolimits_{i=1}^K \mathbb{E}[\boldsymbol{M}_i\boldsymbol{M}_i^T]\Big\|_2, \Big\|\sum\nolimits_{i=1}^K \mathbb{E}[\boldsymbol{M}_i^T\boldsymbol{M}_i]\Big\|_2\Big\}.$$

It holds that

$$\mathbb{P}\Big(\Big\|\sum\nolimits_{i=1}^K \boldsymbol{M}_i\Big\|_2 \geq \delta\Big) \leq 2n\exp\Big(-\frac{\delta^2/2}{\sigma^2 + R\delta/3}\Big), \quad \forall \delta \geq 0.$$

Lemma 2 *For any matrix $\boldsymbol{B} = [b_{ij}] \in \mathbb{R}^{n\times n}$, it holds that*

$$\|\boldsymbol{B}\|_2 \leq \max\{\|\boldsymbol{B}\|_1, \|\boldsymbol{B}\|_\infty\}.$$

Proof. By definition,

$$\|\boldsymbol{B}\|_2 = \sup_{\|\boldsymbol{x}\|\leq 1, \|\boldsymbol{y}\|\leq 1} \boldsymbol{x}^T\boldsymbol{B}\boldsymbol{y}.$$

Moverover, for all $\|\boldsymbol{x}\| \leq 1,\ \|\boldsymbol{y}\| \leq 1$, we have

$$(\boldsymbol{x}^T\boldsymbol{B}\boldsymbol{y})^2 \leq \Big(\sum_{i,j}|b_{ij}|x_i^2\Big)\Big(\sum_{i,j}|b_{ij}|y_j^2\Big) \leq \|\boldsymbol{B}\|_1\|\boldsymbol{B}\|_\infty.$$

Thus, Lemma 2 holds. □

Theorem 1 *(Formal restatement of Theorem 1) Let $\boldsymbol{A}^{(u)}$ be defined by (2) for any $u \in [n-1]$ and the D-EquiStatic weight matrix $\boldsymbol{W}$ be constructed by (3) with $\{u_i\}_{i=1}^M$ following an independent and identical uniform distribution from $[n-1]$. For any size-independent consensus rate $\rho \in (0,1)$ and probability $p \in (0,1)$, if $M \geq \frac{8}{3\rho^2}\ln\frac{2n}{p}$ it holds with probability at least $1-p$ that*

$$\|\boldsymbol{\Pi}\,\boldsymbol{W}\boldsymbol{x}\| \leq \rho\,\|\boldsymbol{\Pi}\boldsymbol{x}\|, \ \forall \boldsymbol{x} \in \mathbb{R}^n. \tag{10}$$

Proof. Notice that

$$\mathbb{E}[\boldsymbol{A}^{(u_i)}] = \boldsymbol{J}, \ \ \forall i \in \{1, 2, \ldots, M\}.$$

Since each $\boldsymbol{A}^{(u_i)}$ is doubly stochastic, it follows from Lemma 2 that

$$\left\|\boldsymbol{A}^{(u_i)} - \boldsymbol{J}\right\|_2 = \left\|\boldsymbol{\Pi}\boldsymbol{A}^{(u_i)}\right\|_2 \leq \|\boldsymbol{\Pi}\|_2 \left\|\boldsymbol{A}^{(u_i)}\right\|_2 \leq 1.$$

Consequently,

$$\sum_{i=1}^{M} \mathbb{E}\Big[\left\|(\boldsymbol{A}^{(u_i)} - \boldsymbol{J})(\boldsymbol{A}^{(u_i)} - \boldsymbol{J})^T\right\|_2\Big] \leq \sum_{i=1}^{M} \mathbb{E}\Big[\left\|\boldsymbol{\Pi}\boldsymbol{A}^{(u_i)}\right\|_2^2\Big] \leq M.$$

Analogously, $\sum_{i=1}^{M} \mathbb{E}\big[\left\|(\boldsymbol{A}^{(u_i)} - \boldsymbol{J})^T(\boldsymbol{A}^{(u_i)} - \boldsymbol{J})\right\|_2\big] \leq M$. By Lemma 1,

$$\begin{aligned}\mathbb{P}\big(\|\boldsymbol{W} - \boldsymbol{J}\|_2 \geq \rho\big) =& \mathbb{P}\Big(\left\|\textstyle\sum_{i=1}^{M}(\boldsymbol{A}^{(u_i)} - \boldsymbol{J})\right\|_2 \geq M\rho\Big) \\ \leq& 2n\exp\Big(-\frac{M^2\rho^2/2}{M + M\rho/3}\Big) \leq 2n\exp\Big(-\frac{M^2\rho^2/2}{M + M/3}\Big) \leq p,\end{aligned}$$

i.e.,

$$\mathbb{P}(\|\boldsymbol{\Pi}\,\boldsymbol{W}\|_2 \leq \rho) \geq 1 - p.$$

Note that $\boldsymbol{\Pi}\,\boldsymbol{W} = \boldsymbol{\Pi}\,\boldsymbol{W}\boldsymbol{\Pi}$. If $\|\boldsymbol{\Pi}\,\boldsymbol{W}\|_2 \leq \rho$, then

$$\|\boldsymbol{\Pi}\,\boldsymbol{W}\boldsymbol{x}\|^2 = \|\boldsymbol{\Pi}\,\boldsymbol{W}\boldsymbol{\Pi}\boldsymbol{x}\|^2 \leq \|\boldsymbol{\Pi}\,\boldsymbol{W}\|_2^2 \|\boldsymbol{\Pi}\boldsymbol{x}\|^2 \leq \rho^2 \|\boldsymbol{\Pi}\boldsymbol{x}\|^2.$$

Therefore, the conclusion holds. □

The relation $M \geq \frac{8}{3\rho^2}\ln(2n/p)$ is required for theoretical analysis, and it is very conservative. In practice, we can set M to be far less than $\frac{8}{3\rho^2}\ln(2n/p)$ and repeat the process described in the formal version of Theorem 1 until we find a desirable $\boldsymbol{W}$ (see Alg. 3). In addition, the verification condition $\|\boldsymbol{\Pi}\,\boldsymbol{W}\|_2 \leq \rho$ in Alg. 3 can also be dropped in implementations so that we only conduct the "while" loop once. We find that these relaxations can still achieve $\boldsymbol{W}$ with an empirically fast consensus rate (see the illustration in the experiments).

Remark 7 *We have much flexibility in the choice of p, such as $p = 1/2$ or $p = 1/n$. If $p \in (1/\mathrm{poly}\,(\mathrm{n}), 1)$, the corresponding value of M will only differ in constants.*

A.3 Proof of Theorem 2

Due to $\boldsymbol{\Pi}\boldsymbol{A}^{(v_t)} = \boldsymbol{A}^{(v_t)}\boldsymbol{\Pi} = \boldsymbol{\Pi}\boldsymbol{A}^{(v_t)}\boldsymbol{\Pi}$, it follows from Alg. 1 that

$$\begin{aligned}\mathbb{E}\Big[\left\|\boldsymbol{\Pi}\,\boldsymbol{W}^{(t)}\boldsymbol{x}\right\|^2\Big] &= \mathbb{E}\Big[\left\|\boldsymbol{\Pi}\Big((1-\eta)\boldsymbol{I} + \eta\boldsymbol{A}^{(v_t)}\Big)\boldsymbol{x}\right\|^2\Big] \\ &\leq (1-\eta)^2\|\boldsymbol{\Pi}\boldsymbol{x}\|^2 + 2\eta(1-\eta)(\boldsymbol{\Pi}\boldsymbol{x})^T\mathbb{E}\big[\boldsymbol{\Pi}\boldsymbol{A}^{(v_t)}\big]\boldsymbol{\Pi}\boldsymbol{x} + \eta^2\mathbb{E}\Big[\left\|\boldsymbol{A}^{(v_t)}\right\|_2^2\Big]\|\boldsymbol{\Pi}\boldsymbol{x}\|^2\end{aligned}$$

Notice that $\mathbb{E}\big[\boldsymbol{A}^{(v_t)}\big] = \boldsymbol{W}$ and $\|\boldsymbol{\Pi}\,\boldsymbol{W}\boldsymbol{y}\| \leq \rho\|\boldsymbol{\Pi}\boldsymbol{y}\|, \forall \boldsymbol{y} \in \mathbb{R}^n$. Therefore,

$$\begin{aligned}\mathbb{E}\Big[\left\|\boldsymbol{\Pi}\,\boldsymbol{W}^{(t)}\boldsymbol{x}\right\|^2\Big] &\leq \big((1-\eta)^2 + 2\eta\rho(1-\eta) + \eta^2\big)\|\boldsymbol{\Pi}\boldsymbol{x}\|^2 \\ &= \big(1 - 2\eta(1-\eta)(1-\rho)\big)\|\boldsymbol{\Pi}\boldsymbol{x}\|^2.\end{aligned}$$

The proof is completed. □

B One-Peer Undirected EquiTopo Graphs (OU-EquiDyn)

B.1 Illustration for basis graphs of OU-EquiDyn

The associated graph of $\widehat{\boldsymbol{A}}^{(u)} = \frac{1}{2}(\boldsymbol{A}^{(u)} + [\boldsymbol{A}^{(u)}]^T)$ with $n = 6$ are given in Fig. 7. Clearly, there exist non-one-peer graphs.

Consider $n = 6$. The OU-EquiDyn graphs generated by Alg. 2 (when $s = 3$) are presented in Fig. 8.

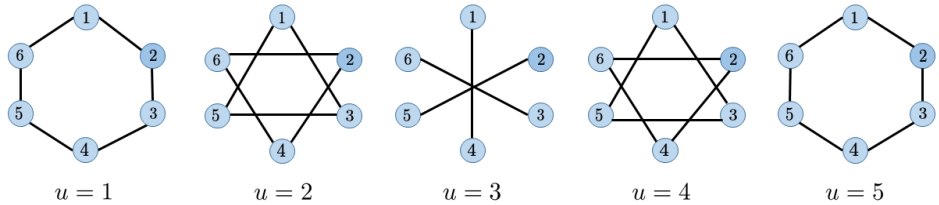

Figure 7: Undirected graphs generated by $\widehat{\boldsymbol{A}}^{(u)} = \frac{1}{2}(\boldsymbol{A}^{(u)} + [\boldsymbol{A}^{(u)}]^T)$.

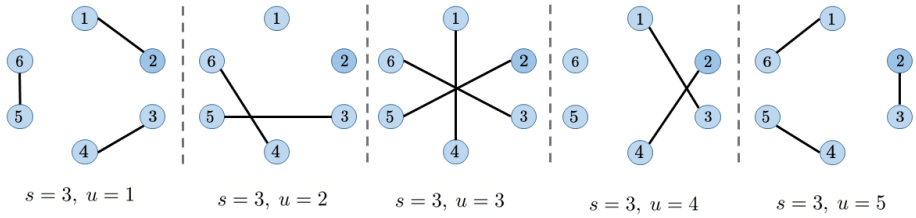

Figure 8: One-peer undirected graphs generated in Alg. 2 with $s = 3$ and $u = 1, \ldots, 5$.

B.2 Node version of Alg. 2

From the node's perspective, an equivalent version of Alg. 2 is presented in Alg. 4. In the remainder of Appendix B, we denote $\lfloor x \rfloor$ as the largest integer no greater than x. We also denote the traditional mod operator as

$$\mathrm{mod}(a, b) = a - b \cdot \left\lfloor \frac{a}{b} \right\rfloor.$$

If $v_t = n/2$, every node i is connected to node $(i + n/2) \bmod n$ for Alg. 2, which is the same as Alg. 4.

If $v_t < \frac{n}{2}$, then $q = v_t$. In Alg. 2, nodes $\{s_t, s_t + 1, \cdots, s_t + q - 1\} \bmod n$ are connected to $\{s_t + q, s_t + q + 1, \cdots, s_t + 2q - 1\} \bmod n$, respectively. If $j \in \{s_t + q, s_t + q + 1, \cdots, s_t + 2q - 1\} \bmod n$, then new edges cannot be added because they have been connected. Similar process starts from connecting node $(s_t + 2q) \bmod n$ with node $(s_t + 3q) \bmod n$, and as a result, for $v_t < \frac{n}{2}$, Alg. 2 can be interpreted as follows: divide $[n]$ into q disjoint subsets:

$$C_\ell = \{i \in [n] : i = (s_t + \ell + d \cdot q) \bmod n,\ d \in \mathbb{Z}\},\ 0 \le \ell < q.$$

Equivalently,

$$C_\ell = \left\{ i \in [n] : i = (s_t + \ell + d \cdot q) \bmod n,\ 0 \le d \le \left\lfloor \frac{n - 1 - \ell}{q} \right\rfloor \right\}.$$

For node i, we define $k(i) = \mathrm{mod}(i - s_t, n)$, $d(i) = \lfloor k(i)/q \rfloor$ and $r(i) = \mathrm{mod}(k(i), q)$. Then, $i \in C_{r(i)}$ and $i = (s_t + r(i) + d(i) \cdot q) \bmod n$.

In each C_ℓ, $s_t + \ell$ is connected with $s_t + \ell + q$, $s_t + \ell + 2q$ is connected with $s_t + \ell + 3q, \ldots$. Thus, if $|C_\ell|$ is even, every node in C_ℓ has a neighbor. If $|C_\ell|$ is odd (equivalently, $\left\lfloor \frac{n-1-\ell}{q} \right\rfloor$ is odd), the node $\ell + q \cdot \left\lfloor \frac{n-1-\ell}{q} \right\rfloor$ is idle and the others has neighbors.

Define

$$C'_\ell = \begin{cases} C_\ell, & \text{if } \left\lfloor \frac{n-1-\ell}{q} \right\rfloor \text{ is odd} \\ \left\{ i \in [n] : i = (s_t + \ell + d \cdot q) \bmod n,\ 0 \le d < \left\lfloor \frac{n - 1 - \ell}{q} \right\rfloor \right\}, & \text{otherwise.} \end{cases}$$

Then, node i has a neighbor if and only if it is in the set $C'_{r(i)}$. In addition, for node i in $C'_{r(i)}$, it is connected to $(i + q) \bmod n$ if $d(i)$ is even; and connected to $(i - q) \bmod n$ if $d(i)$ is odd.

Algorithm 4: OU-EquiDyn weight matrix generation at iteration t (from nodes' perspective)

Input: $\eta \in (0,1)$; basis index $\{u_i, -u_i\}_{i=1}^M$ from a weight matrix $\widetilde{\boldsymbol{W}} \in \mathbb{R}^{n\times n}$ of form (5);
Initialize $\boldsymbol{A} = [a_{ij}] = \boldsymbol{I}$;
for *node* $i = 1$ *to* n *(in parallel)* **do**
  Pick v_t from $\{u_i, -u_i\}_{i=1}^M$ and $s_t \in [n]$ uniformly at random using the common random seed;
  if $v_t \le n/2$ **then**
    $q = v_t$;
    $k(i) = \mathrm{mod}(i - s_t, n)$;
  end
  else
    $q = n - v_t$;
    $k(i) = \mathrm{mod}(i - s_t + q, n)$;
  end
  $r(i) = \mathrm{mod}(k(i), q)$;
  $d(i) = \lfloor k(i)/q \rfloor$;
  if $\lfloor (n-1-r(i))/q \rfloor$ *is odd or* $d(i) < \lfloor (n-1-r(i))/q \rfloor$ **then**
    if $d(i)$ *is even* **then**
      $j = (i+q) \bmod n$;
    end
    else
      $j = (i-q) \bmod n$;
    end
    $a_{ij} = (n-1)/n$;
    $a_{ii} = 1/n$;
  end
end
Output: $\widetilde{\boldsymbol{W}}^{(t)} = (1-\eta)\boldsymbol{I} + \eta \boldsymbol{A}$

In Alg. 4, we compute $r(i)$ and $d(i)$ firstly, and then check whether node i is in $C'_{r(i)}$. If $i \in C'_{r(i)}$ and $d(i)$ is even (odd) , then it is connected to $(i+q) \bmod n$ ($(i-q) \bmod n$). Otherwise, node i is idle, i.e., $a_{ii} = 1$. This yields the equivalence between Alg.2 and Alg. 4 for the case of $v_t < n/2$.

If $v_t > n/2$, let $q = n - v_t < n/2$. Then the nodes in $\{s_t, s_t+1, \cdots, s_t+q-1\} \mod n$ are connected to the nodes $\{s_t+v_t, s_t+1+v_t, \cdots, s_t+q-1+v_t\} \mod n$ respectively. Note that $(i+v_t) \bmod n$ is equivalent to $(i-q) \bmod n$. Then, equivalently, nodes $\{s_t, s_t+1, \cdots, s_t+q-1\} \mod n$ are connected with $\{s_t-q, s_t-q+1, \cdots, s_t-1\} \mod n$, respectively. Similar process starts from connecting node $(s_t+2q) \bmod n$ with node $(s_t+q) \bmod n$. Then, the undirected graph generated with the starting point s_t and the label difference $v_t > n/2$ is equivalent to the undirected graph generated with the starting point $(s_t - q) \bmod n$ and the label difference q. So by setting $k(i) = \mathrm{mod}(i - (s_t - q), n)$, the proof follows by similar arguments for the case $v_t < n/2$.

B.3 Proof of Theorem 4

We first provide the following three lemmas. In the remainder of Appendix B, for any matrix $\boldsymbol{A} = [a_{ij}] \in \mathbb{R}^{n\times n}$, we denote its edge set as

$$\mathcal{E}(\boldsymbol{A}) = \{(i,j) \in [n] \times [n] : \ a_{ij} > 0, \ i \neq j\}.$$

Denote the matrix $\boldsymbol{A}$ generated at the t-th iteration of Alg. 2 by $\widetilde{\boldsymbol{A}}^{(v_t)} = [\widetilde{a}_{ij}^{(v_t)}] \in \mathbb{R}^{n\times n}$. Note that $\widetilde{\boldsymbol{A}}^{(v_t)}$ is also stochastic even when v_t is given since s_t is randomly chosen from $[n]$.

Lemma 3 *For any $n \geq 2$, it holds for $\widetilde{\boldsymbol{A}}^{(v_t)}$ defined by Alg. 2 that it holds that*

$$\mathbb{E}\big[|\mathcal{E}(\widetilde{\boldsymbol{A}}^{(v_t)})|\big] \geq \frac{2n}{3}.$$

Proof. Because $|\mathcal{E}(\widetilde{\boldsymbol{A}}^{(v_t)})|$ is invariant with respect to s_t, it suffices to prove $|\mathcal{E}(\widetilde{\boldsymbol{A}}^{(v_t)})| \geq 2n/3$ for $s_t = 1$.

For $v_t \leq n/2$, we define $m = \lfloor n/(2v_t) \rfloor$ and $r = \mathrm{mod}(n, 2v_t)$, then, $m \geq 1$. Notice that node i is connected with $i + v_t$ for any $i \in \{\ell + 2dv_t : \ 1 \leq \ell \leq v_t, \ 0 \leq d < m\}$.

If $r \leq v_t$, then the last r nodes are idle. As $m \geq 1$, we have $n = 2v_t m + r \geq 2v_t + r \geq 3r$. Thus,

$$|\mathcal{E}(\widetilde{\boldsymbol{A}}^{(v_t)})| \geq n - r \geq n - \frac{1}{3}n = \frac{2}{3}n.$$

If $r > v_t$, then node i in $\{2mv_t + \ell : \ 1 \leq \ell \leq r - v_t\}$ is connected to $i + v_t$. As a result, only the nodes in $\{2mv_t + \ell : \ r - v_t + 1 \leq \ell \leq v_t\}$ are idle, i.e., $2v_t - r$ nodes are idle. We have $n = 2mv_t + r > 3v_t$ from $m \geq 1$ and $r > v_t$. Consequently,

$$|\mathcal{E}(\widetilde{\boldsymbol{A}}^{(v_t)})| \geq n - (2v_t - r) \geq n - v_t \geq \frac{2}{3}n.$$

We have shown $|\mathcal{E}(\widetilde{\boldsymbol{A}}^{(v_t)})| \geq \frac{2}{3}n$ for $v_t \leq n/2$.

For $v_t > n/2$, recall that we have shown in Appendix B.2 that the undirected graph generated with label difference v_t and starting point s_t equals the undirected graph generated with label difference $q = n - v_t < n/2$ and starting point $(s_t - q) \bmod n$. Since the number of edges is invariant with the starting point, the case $v_t > n/2$ has been reduced to $v_t < n/2$. This completes the proof. □

Lemma 4 *For any symmetric matrix $\boldsymbol{B} = [b_{ij}] \in \mathbb{R}^{n \times n}$, if $\boldsymbol{B}\mathbb{1}_n = \boldsymbol{0}_n$, then*

$$\boldsymbol{x}^T \boldsymbol{B} \boldsymbol{x} = -\frac{1}{2} \sum_{i,j} b_{ij}(x_i - x_j)^2, \quad \forall \boldsymbol{x} \in \mathbb{R}^n.$$

Proof. The i-th entry of $\boldsymbol{B}\boldsymbol{x}$ is

$$[\boldsymbol{B}\boldsymbol{x}]_i = \sum_j b_{ij} x_j = b_{ii} x_i + \sum_{j: j \neq i} b_{ij} x_j = \sum_j b_{ij}(x_j - x_i).$$

Hence,

$$\boldsymbol{x}^T \boldsymbol{B} \boldsymbol{x} = \sum_{i,j} b_{ij} x_i (x_j - x_i).$$

Due to $\boldsymbol{B}^T = \boldsymbol{B}$, we have

$$\boldsymbol{x}^T \boldsymbol{B} \boldsymbol{x} = \sum_{i,j} b_{ij} x_j (x_i - x_j).$$

Averaging the above equations yields the result. □

Lemma 5 *For any $n \geq 2$, it holds for $\widetilde{\boldsymbol{A}}^{(v_t)}$ that*

$$\mathbb{E}[\widetilde{\boldsymbol{A}}^{(v_t)}] \preceq \frac{1}{3}\boldsymbol{I} + \frac{1}{3}\Big(\boldsymbol{A}^{(v_t)} + [\boldsymbol{A}^{(v_t)}]^T)\Big).$$

Proof. If $v_t = n/2$, then $\widetilde{\boldsymbol{A}}^{(v_t)} = \boldsymbol{A}^{(v_t)} = [\boldsymbol{A}^{(v_t)}]^T$. By Lemma 4, $\boldsymbol{x}^T \left(\boldsymbol{I} - \boldsymbol{A}^{(v_t)}\right) \boldsymbol{x} \geq 0$, i.e., $\boldsymbol{A}^{(v_t)} \preceq \boldsymbol{I}$. Thus,

$$\widetilde{\boldsymbol{A}}^{(v_t)} = \boldsymbol{A}^{(v_t)} \preceq \frac{1}{3}\boldsymbol{I} + \frac{2}{3}\left(\boldsymbol{A}^{(v_t)} + [\boldsymbol{A}^{(v_t)}]^T\right).$$

Consider $v_t \neq n/2$. Notice that $\mathcal{E}(\widetilde{\boldsymbol{A}}^{(v_t)}) \subset \mathcal{E}\big(\boldsymbol{A}^{(v_t)} + [\boldsymbol{A}^{(v_t)}]^T\big)$ and $|\mathcal{E}(\boldsymbol{A}^{(v_t)} + [\boldsymbol{A}^{(v_t)}]^T)| \leq 2n$. By Lemma 3 and the fact that s_t is from the uniform distribution over $[n]$, it holds for any $(i,j) \in \mathcal{E}(\boldsymbol{A}^{(v_t)} + [\boldsymbol{A}^{(v_t)}]^T)$ that

$$\mathbb{P}[(i,j) \in \mathcal{E}(\widetilde{\boldsymbol{A}}^{(v_t)})] \geq \frac{1}{2n}\mathbb{E}[|\mathcal{E}(\widetilde{\boldsymbol{A}}^{(v_t)})|] \geq \frac{1}{3}.$$

By the construction of $\widetilde{\boldsymbol{A}}^{(v_t)}$ in Alg. 2 and $\boldsymbol{A}^{(v_t)}$ in (2), the non-diagonal and non-zero entries are $\frac{n-1}{n}$. It follows from Lemma 4 that for any $\boldsymbol{x} \in \mathbb{R}^n$, we have

$$\begin{aligned}
\boldsymbol{x}^T\mathbb{E}[\mathbf{I} - \widetilde{\boldsymbol{A}}^{(v_t)}]\boldsymbol{x} &= \frac{n-1}{2n}\mathbb{E}\left[\sum_{(i,j)\in\mathcal{E}(\widetilde{\boldsymbol{A}}^{(v_t)})}(x_i - x_j)^2\right] \\
&= \frac{n-1}{2n}\sum_{(i,j)\in\mathcal{E}(\boldsymbol{A}^{(v_t)}+[\boldsymbol{A}^{(v_t)}]^T)}\mathbb{P}[(i,j) \in \mathcal{E}(\widetilde{\boldsymbol{A}}^{(v_t)})](x_i - x_j)^2 \\
&\geq \frac{n-1}{6n}\sum_{(i,j)\in\mathcal{E}(\boldsymbol{A}^{(v_t)}+[\boldsymbol{A}^{(v_t)}]^T)}(x_i - x_j)^2 \\
&= \frac{1}{3}\boldsymbol{x}^T\big(2\mathbf{I} - \boldsymbol{A}^{(v_t)} - [\boldsymbol{A}^{(v_t)}]^T\big)\boldsymbol{x}.
\end{aligned}$$

Rearranging the terms, we derive

$$\mathbb{E}\left[\widetilde{\boldsymbol{A}}^{(v_t)}\right] \preceq \frac{1}{3}\boldsymbol{I} + \frac{1}{3}\left(\boldsymbol{A}^{(v_t)} + [\boldsymbol{A}^{(v_t)}]^T\right).$$

This completes the proof. □

Proof of Theorem 4 It follows from Lemma 5 that

$$\mathbb{E}\Big[\widetilde{\boldsymbol{A}}^{(v_t)}\Big] \preceq \frac{1}{3}\boldsymbol{I} + \frac{1}{3}\mathbb{E}\Big[\boldsymbol{A}^{(v_t)} + [\boldsymbol{A}^{(v_t)}]^T\Big] = \frac{1}{3}\boldsymbol{I} + \frac{2}{3}\widetilde{\boldsymbol{W}}.$$

Consequently,

$$\begin{aligned}
&\mathbb{E}\Big[\Big\|\boldsymbol{\Pi}\,\widetilde{\boldsymbol{W}}^{(t)}\boldsymbol{x}\Big\|^2\Big] = \mathbb{E}\Big[\Big\|\boldsymbol{\Pi}\Big((1-\eta)\boldsymbol{I} + \eta\widetilde{\boldsymbol{A}}^{(v_t)}\Big)\boldsymbol{x}\Big\|^2\Big] \\
&\quad\leq (1-\eta)^2\left\|\boldsymbol{\Pi}\boldsymbol{x}\right\|^2 + 2\eta(1-\eta)(\boldsymbol{\Pi}\boldsymbol{x})^T\mathbb{E}\big[\widetilde{\boldsymbol{A}}^{(v_t)}\big]\boldsymbol{\Pi}\boldsymbol{x} + \eta^2\mathbb{E}\Big[\Big\|\widetilde{\boldsymbol{A}}^{(v_t)}\Big\|_2^2\Big]\left\|\boldsymbol{\Pi}\boldsymbol{x}\right\|^2.
\end{aligned}$$

Combining the above two inequalities, we derive

$$\begin{aligned}
\mathbb{E}\Big[\Big\|\boldsymbol{\Pi}\,\widetilde{\boldsymbol{W}}^{(t)}\boldsymbol{x}\Big\|^2\Big] &\leq \big((1-\eta)^2 + \frac{2}{3}\eta(1-\eta) + \frac{4}{3}\eta\rho(1-\eta) + \eta^2\big)\left\|\boldsymbol{\Pi}\boldsymbol{x}\right\|^2 \\
&= \big(1 - \frac{4}{3}\eta(1-\eta)(1-\rho)\big)\left\|\boldsymbol{\Pi}\boldsymbol{x}\right\|^2.
\end{aligned}$$

Thus, the proof is completed. □

B.4 An alternative construction of OU-EquiDyn

Alg. 5 provides a different way to construct one-peer undirected graphs which achieve a similar consensus rate as the graphs generated by Alg. 2 but with a different structure.

We denote the matrix $\boldsymbol{A}$ generated at the t-th iteration of Alg. 5 by $\overline{\boldsymbol{A}}^{(v_t)} = [\overline{a}_{ij}^{(v_t)}] \in \mathbb{R}^{n\times n}$. Next, we explain the motivation of Alg. 5.

Denote $\gcd(a,b)$ as the greastest common divisor of a and b. Let $d = \gcd(v_t, n)$ and $\tilde{v}_t = v_t/d$, $\tilde{n} = n/d$. Then, $\tilde{v}_t$ and $\tilde{n}$ are coprime.

Firstly, we divide $[n]$ into d disjoint subsets:

$$\overline{C}_\ell = \{i \in [n] :\ \mathrm{mod}(i - s_t, d) = \ell\},\ 0 \leq \ell < d.$$

Algorithm 5: Alternative OU-EquiDyn weight matrix generation at iteration t (from nodes' perspective)

Input: $\eta \in (0,1)$; basis index $\{u_i, -u_i\}_{i=1}^M$ from a weight matrix $\widetilde{\boldsymbol{W}} \in \mathbb{R}^{n\times n}$ of form 5;
Initialize $\boldsymbol{A} = [a_{ij}] = \boldsymbol{I}$;
for *node* $i = 1$ *to* n *(in parallel)* **do**
  Pick v_t from $\{u_i, -u_i\}_{i=1}^M$ and $s_t \in [n]$ uniformly at random using the common random seed;
  Compute $d = \gcd(v_t, n)$ and find $1 \le b \le n/d - 1$ such that $\mathrm{mod}(b \cdot v_t, n) = d$ by Euclidean algorithm;
  Set $\tilde{n} = n/d$, and $m(i) = \mathrm{mod}(\lfloor (i - s_t)/d \rfloor \cdot b, \tilde{n})$;
  if $\tilde{n}$ *is even or* $m(i) < \tilde{n} - 1$ **then**
    if m *is even* **then**
      $j = (i + v_t) \bmod n$;
    end
    else
      $j = (i - v_t) \bmod n$;
    end
    $a_{ij} = (n-1)/n$;
    $a_{ii} = 1/n$;
  end
end
Output: $\widetilde{\boldsymbol{W}}^{(t)} = (1-\eta)\boldsymbol{I} + \eta \boldsymbol{A}$

Clearly, $\left|\overline{C}_\ell\right| = n/d = \tilde{n}$, $\forall 0 \le \ell < d$.

We claim that

$$\overline{C}_\ell = \{s_t + \ell + m v_t \bmod n : \ 0 \le m < \tilde{n}\}, \ 0 \le \ell < d. \tag{11}$$

To proof the claim, we denote the RHS of (11) by $\widetilde{C}_\ell$. Since v_t can be divided evenly by d, for any $i \in \widetilde{C}_\ell$, it satisfies $\mathrm{mod}(i, d) = \mathrm{mod}((s_t + \ell), d)$. Combining with the fact that $\widetilde{C}_\ell \subset [n]$, we have $\widetilde{C}_\ell \subset \overline{C}_\ell$.

Then, since $\tilde{n}$ and $\tilde{v}$ are coprime, for any $0 \le m_1 < m_2 < \tilde{n}$, $\mathrm{mod}(m_1 \tilde{v}_t, \tilde{n}) \ne \mathrm{mod}(m_2 \tilde{v}_t, \tilde{n})$. Then, $\mathrm{mod}(m_1 v_t, n) \ne \mathrm{mod}(m_2 v_t, n)$. Thus, $\left|\widetilde{C}_\ell\right| = \tilde{n} = \left|\overline{C}_\ell\right|$. Combining with $\widetilde{C}_\ell \subset \overline{C}_\ell$, we have $\overline{C}_\ell = \widetilde{C}_\ell$.

The above analysis also implies that for each $i \in \overline{C}_\ell$, there exists a unique $0 \le m(i) < d$ such that $i = (s_t + \ell + m(i) v_t) \bmod n$.

By (11), a natural way to construct one-peer undirected graphs is: in each $\overline{C}_\ell$, connect $s_t + \ell$ with $s_t + \ell + v_t$, connect $s_t + \ell + 2v_t$ with $s_t + \ell + 3v_t, \ldots$. Equivalently, i is connected with $(i + v_t) \bmod n$ if $m(i)$ is even and connected with $(i - v_t) \bmod n$ if $m(i)$ is odd.

In this way, if $\tilde{n}$ is even, every node in C_ℓ has a neighbor, $\forall 0 \le \ell < d$, i.e., every node in $[n]$ has a neighbor. Equivalently, a node $i \in \overline{C}_\ell$ has a neighbor if it is in the set

$$\overline{C}'_\ell = \begin{cases} C_\ell, & \text{if } \tilde{n} \text{ is even} \\ \{(s_t + \ell + m v_t) \bmod n : \ 0 \le m < \tilde{n} - 1\}, & \text{if } \tilde{n} \text{ is odd} \end{cases} \tag{12}$$

To give a practical way of the process described above from each node's perspective, for each node $i \in \overline{C}_\ell$, we provide a more efficient way to compute the unique $0 \le m(i) < \tilde{n}$ such that $i = (s_t + \ell + m(i) v_t) \bmod n$. Firstly, for each $i \in \overline{C}_\ell$, since $\ell = \mathrm{mod}(i - s_t, d)$, we have

$$i = (s_t + \ell + \lfloor (i - s_t)/d \rfloor \cdot d) \bmod n.$$

Then, by Bézout's theorem, since $\tilde{n}$ and $\tilde{v}_t$ are coprime, there exist integers $1 \leq b \leq \tilde{n}-1$ and b' such that $b \cdot v_t + b' \cdot n = d$. The pair (b, b') can be computed by the Euclidean algorithm in $\mathcal{O}(\ln(n))$ time. Then,

$$\lfloor (i-s_t)/d \rfloor \cdot d = \lfloor (i-s_t)/d \rfloor \cdot (bv_t + b'n).$$

Define $m(i) = \mathrm{mod}(\lfloor (i-s_t)/d \rfloor \cdot b, \tilde{n})$, then, we have $(\lfloor (i-s_t)/d \rfloor \cdot d) \bmod n = m(i)v_t \bmod n$, Thus, $i = (s_t + \ell + m(i)v_t) \bmod n$.

In Alg. 5, each node i computes $\tilde{n}$ and its $m(i)$ firstly. If $\tilde{n}$ is even, every node has a neighbor. If $\tilde{n}$ is odd but $m(i) < \tilde{n}-1$, then $i \in \overline{C}'_\ell$ where $\ell = \mod (i-s_t, d)$, i.e., i also has a neighbor. In this way, each node can determine whether it has a neighbor in this iteration. If node i has a neighbor, as we have described above, it is connected with $(i+v_t) \bmod n$ if $m(i)$ is even and $(i-v_t) \bmod n$ otherwise.

The following lemma is used to prove Lemma 7.

Lemma 6 *Let $d = \gcd(v_t, n)$, then*

$$\mathbb{E}\big[|\mathcal{E}(\overline{\boldsymbol{A}}^{(v_t)})|\big] = (2d) \cdot \left\lfloor \frac{n}{2d} \right\rfloor \geq \frac{2n}{3}.$$

Proof. Define $\overline{C}_\ell$ and $\overline{C}'_\ell$ as in (11) and (12). If n can be divided by $2d$ evenly, i.e., $\left|\overline{C}_\ell\right| = \tilde{n}$ is even for any $0 \leq \ell < d$. Then, every node has a neighbor, i.e., $\left|\mathcal{E}(\overline{\boldsymbol{A}}^{(v_t)})\right| = n = (2d) \cdot \left\lfloor \frac{n}{2d} \right\rfloor$.

If n cannot be divided by $2d$ evenly, by (12), in each $\overline{C}_\ell$, there is one node idle in this iteration. Then, we also have $\left|\mathcal{E}(\overline{\boldsymbol{A}}^{(v_t)})\right| = n - d = (2d) \cdot \left\lfloor \frac{n}{2d} \right\rfloor$. As $d = \gcd(v_t, n)$, we have $n = (2k+1)d$, with $k \in \mathbb{Z}$. Since v_t can be divided by d evenly and $v_t \leq n-1$, we have $d < n$. Thus, $k \geq 1$, i.e., $n \geq 3d$. Then, $\left|\mathcal{E}(\overline{\boldsymbol{A}}^{(v_t)})\right| = n - d \geq n - \frac{n}{3} = \frac{2n}{3}$.

Since the above analysis holds for arbitrary $s_t \in [n]$, the lemma is proved. □

The following lemma follows by similar arguments with Lemma 5.

Lemma 7 *For any $n \geq 2$, the output matrix $\overline{\boldsymbol{A}}^{(v_t)}$ of Algorithm 2 satisfies*

$$\mathbb{E}[\overline{\boldsymbol{A}}^{(v_t)}] \preceq \frac{1}{3}\boldsymbol{I} + \frac{1}{3}\Big(\boldsymbol{A}^{(v_t)} + [\boldsymbol{A}^{(v_t)}]^T\Big).$$

The following consensus rate for Alg. 5 is proved similarly to Theorem 4.

Theorem 7 *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. 5, it holds that*

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

C Applying EquiTopo Matrices to Decentralized Learning

C.1 Convergence of DSGD for strongly convex cost functions

We assume that $f_i(\boldsymbol{x})$ is μ-strongly convex for any i, i.e., there exists a constant $\mu > 0$ such that

$$f_i(\boldsymbol{y}) \geq f_i(\boldsymbol{x}) + \langle \nabla f_i(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\mu}{2}\left\|\boldsymbol{y} - \boldsymbol{x}\right\|^2, \ \ \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d.$$

As we have tested the performance of DSGD with EquiTopo matrices for strongly convex cost functions, we attach the following convergence result of the algorithm (8). The proof follows by [12, Theorem 2] (or Appendix A.4 therein) and is omitted here.

Theorem 8 *Consider the DSGD algorithm* (8) *utilizing the EquiTopo matrices, and f_i being μ-strongly convex for all i. Under Assumptions A.1-A.3, it holds that*

$$\frac{1}{H_T}\sum_{t=0}^{T} h^{(t)}\mathbb{E}\Big[f(\bar{\boldsymbol{x}}^{(t)}) - f(\boldsymbol{x}^*)\Big] = \tilde{\mathcal{O}}\Big(\frac{\sigma^2}{nT} + \frac{\kappa\beta\sigma^2}{(1-\beta)T^2} + \frac{\kappa\beta b^2}{(1-\beta)^2T^2} + \frac{1}{1-\beta}\exp\big(-\frac{(1-\beta)T}{\kappa}\big)\Big)$$

where $\kappa = L/\mu$, $\tilde{\mathcal{O}}(\cdot)$ hides constants and polylogarithmic factors, positive weights $h^{(t)} = \big(1-\frac{\mu\gamma}{2}\big)^t$, and $H_T = \sum_{t=0}^{T} h^{(t)}$. Furthermore,

- *$\beta = \rho$ with D-EquiStatic $\boldsymbol{W}$ or U-EquiStatic $\widetilde{\boldsymbol{W}}$;*
- *$\beta = \sqrt{(1+\rho)/2}$ 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 $\eta = 1/2$).*

C.2 Transient iteration

The computation of transient iteration.

For nonconvex cost functions, the convergence rate of (8) is given by

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

To reach the linear speedup stage, the iteration T has to be sufficiently large so that the $\sqrt{nT}$-term dominates, i.e., $\frac{\sigma}{\sqrt{nT}} \geq \frac{\beta^{\frac{2}{3}}\sigma^{\frac{2}{3}}}{T^{\frac{2}{3}}(1-\beta)^{\frac{1}{3}}}$, $\frac{\sigma}{\sqrt{nT}} \geq \frac{\beta^{\frac{2}{3}}b^{\frac{2}{3}}}{T^{\frac{2}{3}}(1-\beta)^{\frac{2}{3}}}$, and moreover, $\frac{\sigma}{\sqrt{nT}} \geq \frac{\beta}{T(1-\beta)}$. Then $T \geq \frac{\beta^4 n^3}{(1-\beta)^2\sigma^2}$, $T \geq \frac{\beta^4 b^4 n^3}{(1-\beta)^4\sigma^6}$, and $T \geq \frac{\beta^2 n}{(1-\beta)^2\sigma^2}$. Substituting β into the inequalities, transient iterations under different networks can be computed. Similar methods can be adopted for the transient iterations of the distributed gradient tracking algorithm.

Under different network topologies, for non-convex and strongly convex cost functions, convergence results and transient iterations are shown in Table 3 and 4. The results indicate that the proposed networks are at faster rates.

Table 3: For non-convex cost functions, per-iteration communication and convergence rate comparison between DSGD over different topologies. The smaller the transient iteration complexity is, the faster the algorithm converges.

Topology	Per-iter Comm.	Convergence Rate	Trans. Iters.
Ring	$\Theta(1)$	$\mathcal{O}\Big(\frac{\sigma}{\sqrt{nT}} + \frac{n^{\frac{2}{3}}\sigma^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{n^{\frac{4}{3}}b^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{n^2}{T}\Big)$	$\mathcal{O}(n^{11})$
Torus	$\Theta(1)$	$\mathcal{O}\Big(\frac{\sigma}{\sqrt{nT}} + \frac{n^{\frac{1}{3}}\sigma^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{n^{\frac{2}{3}}b^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{n}{T}\Big)$	$\mathcal{O}(n^7)$
Static Exp.	$\Theta(\ln(n))$	$\mathcal{O}\Big(\frac{\sigma}{\sqrt{nT}} + \frac{\ln^{\frac{1}{3}}(n)\sigma^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{\ln^{\frac{2}{3}}(n)b^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{\ln(n)}{T}\Big)$	$\mathcal{O}(n^3\ln^4(n))$
O.-P. Exp.	1	$\mathcal{O}\Big(\frac{\sigma}{\sqrt{nT}} + \frac{\ln^{\frac{1}{3}}(n)\sigma^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{\ln^{\frac{2}{3}}(n)b^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{\ln(n)}{T}\Big)$	$\mathcal{O}(n^3\ln^4(n))$
D(U)-EquiStatic	$\Theta(\ln(n))$	$\mathcal{O}\Big(\frac{\sigma}{\sqrt{nT}} + \frac{\sigma^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{b^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{1}{T}\Big)$	$\mathcal{O}(n^3)$
OD (OU)-EquiDyn	1	$\mathcal{O}\Big(\frac{\sigma}{\sqrt{nT}} + \frac{\sigma^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{b^{\frac{2}{3}}}{T^{\frac{2}{3}}} + \frac{1}{T}\Big)$	$\mathcal{O}(n^3)$

C.3 Decentralized stochastic gradient tracking algorithm

We write local variables compactly into matrix form, for instance

$$\boldsymbol{X}^{(t)} = \Big[\boldsymbol{x}_1^{(t)}, \cdots, \boldsymbol{x}_n^{(t)}\Big]^T \in \mathbb{R}^{n\times d},\ \nabla\boldsymbol{F}\big(\boldsymbol{X}^{(t)}\big) = \Big[\nabla f_1\Big(\boldsymbol{x}_1^{(t)}\Big), \cdots, \nabla f_n\Big(\boldsymbol{x}^{(t)}\Big)\Big]^T \in \mathbb{R}^{n\times d}.$$

Table 4: For strongly convex cost functions, per-iteration communication and convergence rate comparison between DSGD over different topologies. The smaller the transient iteration complexity is, the faster the algorithm converges.

Topology	Per-iter Comm.	Convergence Rate	Trans. Iters.
Ring	$\Theta(1)$	$\tilde{\mathcal{O}}\left(\frac{\sigma^2}{nT} + \frac{\kappa n^2 \sigma^2}{T^2} + \frac{\kappa n^4 b^2}{T^2}\right)$	$\tilde{\mathcal{O}}(\kappa n^5)$
Torus	$\Theta(1)$	$\tilde{\mathcal{O}}\left(\frac{\sigma^2}{nT} + \frac{\kappa n \sigma^2}{T^2} + \frac{\kappa n^2 b^2}{T^2}\right)$	$\tilde{\mathcal{O}}(\kappa n^3)$
Static Exp.	$\Theta(\ln(n))$	$\tilde{\mathcal{O}}\left(\frac{\sigma^2}{nT} + \frac{\kappa \ln(n) \sigma^2}{T^2} + \frac{\kappa \ln^2(n) b^2}{T^2}\right)$	$\tilde{\mathcal{O}}(\kappa n \ln^2(n))$
O.-P. Exp.	1	$\tilde{\mathcal{O}}\left(\frac{\sigma^2}{nT} + \frac{\kappa \ln(n) \sigma^2}{T^2} + \frac{\kappa \ln^2(n) b^2}{T^2}\right)$	$\tilde{\mathcal{O}}(\kappa n \ln^2(n))$
D(U)-EquiStatic	$\Theta(\ln(n))$	$\tilde{\mathcal{O}}\left(\frac{\sigma^2}{nT} + \frac{\kappa \sigma^2}{T^2} + \frac{\kappa b^2}{T^2}\right)$	$\tilde{\mathcal{O}}(\kappa n)$
OD (OU)-EquiDyn	1	$\tilde{\mathcal{O}}\left(\frac{\sigma^2}{nT} + \frac{\kappa \sigma^2}{T^2} + \frac{\kappa b^2}{T^2}\right)$	$\tilde{\mathcal{O}}(\kappa n)$

The matrices $\boldsymbol{Y}^{(t)}, \boldsymbol{G}^{(t)} \in \mathbb{R}^{n \times d}$ are defined analogously. We also denote $\nabla \boldsymbol{F}\left(\boldsymbol{X}^{(-1)}\right) = \boldsymbol{G}^{(-1)} = \mathbf{0}$ for notational simplicity.

Clearly, the DSGT algorithm can be simplified as

$$\begin{pmatrix} \boldsymbol{X}^{(t+1)} \\ \boldsymbol{Y}^{(t+1)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{W}^{(t)} & -\gamma \boldsymbol{W}^{(t)} \\ \mathbf{0} & \boldsymbol{W}^{(t)} \end{pmatrix} \begin{pmatrix} \boldsymbol{X}^{(t)} \\ \boldsymbol{Y}^{(t)} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \boldsymbol{G}^{(t+1)} - \boldsymbol{G}^{(t)} \end{pmatrix}. \tag{13}$$

For simplicity, we define

$$\boldsymbol{W}^{(j:k)} = \boldsymbol{W}^{(k)} \cdots \boldsymbol{W}^{(j)}, \ \forall k \geq j \geq 0,$$

and moreover, $\boldsymbol{W}^{(j:k)} = \boldsymbol{I}$ for $j > k$.

Notice that

$$\begin{pmatrix} \boldsymbol{W}^{(k)} & -\gamma \boldsymbol{W}^{(k)} \\ \mathbf{0} & \boldsymbol{W}^{(k)} \end{pmatrix} \cdots \begin{pmatrix} \boldsymbol{W}^{(j)} & -\gamma \boldsymbol{W}^{(j)} \\ \mathbf{0} & \boldsymbol{W}^{(j)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{W}^{(j:k)} & -\gamma\,(k-j+1)\,\boldsymbol{W}^{(j:k)} \\ \mathbf{0} & \boldsymbol{W}^{(j:k)} \end{pmatrix}.$$

Consequently, it holds for all $t \geq 1$ that

$$\boldsymbol{X}^{(t)} = \boldsymbol{W}^{(0:(t-1))} \boldsymbol{X}^{(0)} - \gamma \sum_{j=0}^{t-1} (t-j)\, \boldsymbol{W}^{(j:(t-1))} \left(\boldsymbol{G}^{(j)} - \boldsymbol{G}^{(j-1)}\right). \tag{14}$$

Moreover, we have two inequalities as follows.

Lemma 8 *For any $t \geq 1$ and $\beta \in (0,1)$, we have*

$$t^2 \beta^{2(t-1)} \leq \frac{c_1}{(1-\beta)^2} \beta^{t-1}, \quad t^2 \beta^{t-1} \leq \frac{c_2}{(1-\beta)^2} \left(\frac{1+\beta}{2}\right)^{t-1},$$

where $c_1 = 4$, $c_2 = 16$.

Proof. Define $r(x) = x^2 \beta^{x-1}$, where $x \geq 1$. Then, for the first inequality, it suffices to show that $r(x) \leq c_1/(1-\beta)^2$ for $x \geq 1$. Due to $r'(x) = x\beta^{x-1}\,(2 + x \ln \beta)$, $r(x)$ attains its maximum at $x_0 = \max\left\{1, -\frac{2}{\ln \beta}\right\}$.

If $-\frac{2}{\ln \beta} > 1$, by combining with the fact that $\ln \beta \leq \beta - 1 < 0$ for $\beta \in (0,1)$, we have

$$x^2 \beta^{x-1} \leq r\left(-\frac{2}{\ln \beta}\right) = \frac{4}{(\ln \beta)^2} \beta^{-\frac{2}{\ln \beta} - 1} \leq \frac{4}{(\ln \beta)^2} \leq \frac{4}{(1-\beta)^2}.$$

If $-\frac{2}{\ln\beta} \leq 1$, then

$$x^2\beta^{x-1} = r(1) = 1 \leq \frac{4}{(1-\beta)^2}.$$

The second inequality follows by similar arguments and the fact that $\sqrt{\beta} \leq \frac{1+\beta}{2}$. The proof is completed. □

The following lemma is a generalization of Cauchy-Schwartz inequality. Its proof follows by using $\|\boldsymbol{A}+\boldsymbol{B}\|_{\mathrm{F}}^2 \leq \frac{1}{\alpha}\|\boldsymbol{A}\|_{\mathrm{F}}^2 + \frac{1}{1-\alpha}\|\boldsymbol{B}\|_{\mathrm{F}}^2$ $(\alpha \in (0,1))$ repeatedly.

Lemma 9 *Consider a sequence of matrices $\{\boldsymbol{B}_i\}_{i=1}^m$. If $\alpha_1, \alpha_2, \cdots, \alpha_m > 0$ and $\sum_{i=1}^m \alpha_i \leq 1$, then*

$$\left\|\sum_{i=1}^m \boldsymbol{B}_i\right\|_{\mathrm{F}}^2 \leq \sum_{i=1}^m \frac{1}{\alpha_i}\|\boldsymbol{B}_i\|_{\mathrm{F}}^2.$$

We define a potential function as

$$\begin{aligned}\Phi^{(t)} =& \frac{4\beta^t}{(1-\beta)^2}\mathbb{E}\left[\left\|\boldsymbol{\Pi}\boldsymbol{X}^{(0)}\right\|_{\mathrm{F}}^2\right] + \frac{16c_1 n\gamma^2}{(1-\beta)^4}\sigma^2 \\ &+ \frac{4c_2\gamma^2}{(1-\beta)^4}\sum_{j=0}^{t-1}\left(\frac{1+\beta}{2}\right)^{t-j-1}\mathbb{E}\left[\left\|\nabla\boldsymbol{F}\left(\boldsymbol{X}^{(j)}\right) - \nabla\boldsymbol{F}\left(\boldsymbol{X}^{(j-1)}\right)\right\|_{\mathrm{F}}^2\right], \ \forall t \geq 1,\end{aligned} \tag{15}$$

and moreover,

$$\Phi^{(0)} = \frac{4}{(1-\beta)^2}\left\|\boldsymbol{\Pi}\boldsymbol{X}^{(0)}\right\|_{\mathrm{F}}^2 + \frac{16c_1 n\gamma^2}{(1-\beta)^4}\sigma^2. \tag{16}$$

The following Lemma 10 and Lemma 11 are used to prove Lemma 12. Theorem 6 follows by combining Lemma 12 with the descent lemma (Lemma 14).

Lemma 10 *Consider the DSGT* (9). *Let Assumptions A.1 and A.2 hold. If $\left\{\boldsymbol{W}^{(t)}\right\}_{t\geq 0}$ have convergence rate β, i.e., $\mathbb{E}\left[\left\|\boldsymbol{\Pi}\boldsymbol{W}^{(t)}\boldsymbol{y}\right\|^2\right] \leq \beta^2\|\boldsymbol{\Pi}\boldsymbol{y}\|^2$ for any $\boldsymbol{y} \in \mathbb{R}^n$, then*

$$\mathbb{E}\left[\left\|\boldsymbol{\Pi}\boldsymbol{X}^{(t)}\right\|_{\mathrm{F}}^2\right] \leq \frac{1-\beta}{2}\Phi^{(t)}, \ \forall t \geq 0.$$

Proof. The case $t = 0$ follows by definition directly.

For $t \geq 1$, we define

$$\boldsymbol{Q}^{(t,1)} = \boldsymbol{W}^{(0:(t-1))}\boldsymbol{X}^{(0)} - \gamma\sum_{j=0}^{t-1}(t-j)\ \boldsymbol{W}^{(j:(t-1))}\left(\nabla\boldsymbol{F}\left(\boldsymbol{X}^{(j)}\right) - \nabla\boldsymbol{F}\left(\boldsymbol{X}^{(j-1)}\right)\right),$$

and

$$\boldsymbol{Q}^{(t,2)} = -\gamma\sum_{j=0}^{t-1}(t-j)\ \boldsymbol{W}^{(j:(t-1))}\left(\boldsymbol{G}^{(j)} - \nabla\boldsymbol{F}\left(\boldsymbol{X}^{(j)}\right) - \boldsymbol{G}^{(j-1)} + \nabla\boldsymbol{F}\left(\boldsymbol{X}^{(j-1)}\right)\right).$$

Recalling (14) gives

$$\boldsymbol{X}^{(t)} = \boldsymbol{Q}^{(t,1)} + \boldsymbol{Q}^{(t,2)}.$$

Then

$$\mathbb{E}\left[\left\|\boldsymbol{\Pi}\boldsymbol{X}^{(t)}\right\|_{\mathrm{F}}^2\right] \leq 2\mathbb{E}\left[\left\|\boldsymbol{\Pi}\boldsymbol{Q}^{(t,1)}\right\|_{\mathrm{F}}^2\right] + 2\mathbb{E}\left[\left\|\boldsymbol{\Pi}\boldsymbol{Q}^{(t,2)}\right\|_{\mathrm{F}}^2\right].$$

Rearranging $\boldsymbol{Q}^{(t,2)}$ yields

$$\begin{aligned}\boldsymbol{Q}^{(t,2)} =& -\gamma\, \boldsymbol{W}^{(t-1)} \left(\boldsymbol{G}^{(t-1)} - \nabla \boldsymbol{F}\big(\boldsymbol{X}^{(t-1)}\big)\right) \\ & -\gamma \sum_{j=0}^{t-2} \left((t-j)\ \boldsymbol{W}^{(j:(t-1))} - (t-j-1)\ \boldsymbol{W}^{((j+1):(t-1))}\right) \left(\boldsymbol{G}^{(j)} - \nabla \boldsymbol{F}\big(\boldsymbol{X}^{(j)}\big)\right).\end{aligned}$$

By Assumption A.2 and the assumption on consensus rate, we have

$$\begin{aligned}&\mathbb{E}\left[\left\|\boldsymbol{\Pi}\, \boldsymbol{Q}^{(t,2)}\right\|_{\mathrm{F}}^2\right] \\ =&\gamma^2 \mathbb{E}\left[\left\|\boldsymbol{\Pi}\, \boldsymbol{W}^{(t-1)} \left(\boldsymbol{G}^{(t-1)} - \nabla \boldsymbol{F}\big(\boldsymbol{X}^{(t-1)}\big)\right)\right\|_{\mathrm{F}}^2\right] \\ &+ \gamma^2 \sum_{j=0}^{t-2} \mathbb{E}\left[\left\|\left((t-j)\, \boldsymbol{\Pi}\, \boldsymbol{W}^{(j:(t-1))} - (t-j-1)\, \boldsymbol{\Pi}\, \boldsymbol{W}^{((j+1):(t-1))}\right) \left(\boldsymbol{G}^{(j)} - \nabla \boldsymbol{F}\big(\boldsymbol{X}^{(j)}\big)\right)\right\|_{\mathrm{F}}^2\right] \\ \leq& \gamma^2 \beta^2 \mathbb{E}\left[\left\|\boldsymbol{G}^{(t-1)} - \nabla \boldsymbol{F}\big(\boldsymbol{X}^{(t-1)}\big)\right\|_{\mathrm{F}}^2\right] + \gamma^2 \sum_{j=0}^{t-2} 4\,(t-j)^2\, \beta^{2(t-j-1)} \mathbb{E}\left[\left\|\boldsymbol{G}^{(j)} - \nabla \boldsymbol{F}\big(\boldsymbol{X}^{(j)}\big)\right\|_{\mathrm{F}}^2\right] \\ \leq& 4n\gamma^2 \sum_{j=1}^{t} j^2 \beta^{2(j-1)} \sigma^2.\end{aligned}$$

By Lemma 8, we have

$$n\gamma^2 \sum_{j=1}^{t} j^2 \beta^{2(j-1)} \sigma^2 \leq \frac{nc_1\gamma^2}{(1-\beta)^2} \sum_{j=1}^{t} \beta^{j-1} \sigma^2 \leq \frac{nc_1\gamma^2}{(1-\beta)^3} \sigma^2.$$

As a result,

$$\mathbb{E}\left[\left\|\boldsymbol{\Pi}\, \boldsymbol{Q}^{(t,2)}\right\|_{\mathrm{F}}^2\right] \leq \frac{4nc_1\gamma^2}{(1-\beta)^3} \sigma^2.$$

Moreover,

$$\begin{aligned}&\mathbb{E}\left[\left\|\boldsymbol{\Pi}\, \boldsymbol{Q}^{(t,1)}\right\|_{\mathrm{F}}^2\right] \\ \leq& \frac{1}{(1-\beta)\, \beta^t} \mathbb{E}\left[\left\|\boldsymbol{\Pi}\, \boldsymbol{W}^{(0:(t-1))} \boldsymbol{X}^{(0)}\right\|_{\mathrm{F}}^2\right] \\ &+ \sum_{j=0}^{t-1} \frac{\gamma^2\, (t-j)^2}{(1-\beta)\, \beta^{t-j-1}} \mathbb{E}\left[\left\|\boldsymbol{\Pi}\, \boldsymbol{W}^{(j:(t-1))} \left(\nabla \boldsymbol{F}\big(\boldsymbol{X}^{(j)}\big) - \nabla \boldsymbol{F}\big(\boldsymbol{X}^{(j-1)}\big)\right)\right\|_{\mathrm{F}}^2\right] \\ \leq& \frac{\beta^t}{(1-\beta)} \mathbb{E}\left[\left\|\boldsymbol{\Pi} \boldsymbol{X}^{(0)}\right\|_{\mathrm{F}}^2\right] + \frac{\gamma^2}{1-\beta} \sum_{j=0}^{t-1} (t-j)^2\, \beta^{t-j-1} \mathbb{E}\left[\left\|\nabla \boldsymbol{F}\big(\boldsymbol{X}^{(j)}\big) - \nabla \boldsymbol{F}\big(\boldsymbol{X}^{(j-1)}\big)\right\|_{\mathrm{F}}^2\right] \\ \leq& \frac{\beta^t}{1-\beta} \mathbb{E}\left[\left\|\boldsymbol{\Pi} \boldsymbol{X}^{(0)}\right\|_{\mathrm{F}}^2\right] + \frac{c_2\gamma^2}{(1-\beta)^3} \sum_{j=0}^{t-1} \left(\frac{1+\beta}{2}\right)^{t-j-1} \mathbb{E}\left[\left\|\nabla \boldsymbol{F}\big(\boldsymbol{X}^{(j)}\big) - \nabla \boldsymbol{F}\big(\boldsymbol{X}^{(j-1)}\big)\right\|_{\mathrm{F}}^2\right],\end{aligned}$$

where the first inequality follows by Lemma 9 and the fact that $(1-\beta)\sum_{j=0}^{t} \beta^j < 1$, the second inequality is by the assumption on consensus rate, and the third inequality is by Lemma 8.

Therefore, the conclusion holds by the definition of $\Phi^{(t)}$. □

Lemma 11 *Consider the DSGT* (9)*. Let Assumptions A.1 and A.2 hold. If $\gamma \leq \frac{1}{L}$, it holds for $t \geq 0$ that*

$$\begin{aligned}&\mathbb{E}\left[\left\|\nabla \boldsymbol{F}\left(\boldsymbol{X}^{(t+1)}\right)-\nabla \boldsymbol{F}\left(\boldsymbol{X}^{(t)}\right)\right\|_{\mathrm{F}}^{2}\right]\\ \leq & 6 n \gamma^{2} L^{2} \mathbb{E}\left[\left\|\nabla f\left(\bar{\boldsymbol{x}}^{(t)}\right)\right\|^{2}\right]+9 L^{2} \mathbb{E}\left[\left\|\boldsymbol{\Pi} \boldsymbol{X}^{(t)}\right\|_{\mathrm{F}}^{2}\right]+3 L^{2} \mathbb{E}\left[\left\|\boldsymbol{\Pi} \boldsymbol{X}^{(t+1)}\right\|_{\mathrm{F}}^{2}\right]+3 \gamma^{2} L^{2} \sigma^{2}.\end{aligned}$$

Proof. Clearly,

$$\begin{aligned}&\mathbb{E}\left[\left\|\nabla \boldsymbol{F}\left(\boldsymbol{X}^{(t+1)}\right)-\nabla \boldsymbol{F}\left(\boldsymbol{X}^{(t)}\right)\right\|_{\mathrm{F}}^{2}\right] \leq 3 \mathbb{E}\left[\left\|\nabla \boldsymbol{F}\left(\bar{\boldsymbol{X}}^{(t+1)}\right)-\nabla \boldsymbol{F}\left(\bar{\boldsymbol{X}}^{(t)}\right)\right\|_{\mathrm{F}}^{2}\right]\\ &+3 \mathbb{E}\left[\left\|\nabla \boldsymbol{F}\left(\boldsymbol{X}^{(t+1)}\right)-\nabla \boldsymbol{F}\left(\bar{\boldsymbol{X}}^{(t+1)}\right)\right\|_{\mathrm{F}}^{2}\right]+3 \mathbb{E}\left[\left\|\nabla \boldsymbol{F}\left(\boldsymbol{X}^{(t)}\right)-\nabla \boldsymbol{F}\left(\bar{\boldsymbol{X}}^{(t)}\right)\right\|_{\mathrm{F}}^{2}\right].\end{aligned}$$

It follows from Assumption A.1 that

$$\begin{aligned}&\mathbb{E}\left[\left\|\nabla \boldsymbol{F}\left(\boldsymbol{X}^{(t+1)}\right)-\nabla \boldsymbol{F}\left(\boldsymbol{X}^{(t)}\right)\right\|_{\mathrm{F}}^{2}\right]\\ \leq & 3 L^{2}\left(\mathbb{E}\left[\left\|\bar{\boldsymbol{X}}^{(t+1)}-\bar{\boldsymbol{X}}^{(t)}\right\|_{\mathrm{F}}^{2}\right]+\mathbb{E}\left[\left\|\boldsymbol{\Pi} \boldsymbol{X}^{(t+1)}\right\|_{\mathrm{F}}^{2}\right]+\mathbb{E}\left[\left\|\boldsymbol{\Pi} \boldsymbol{X}^{(t)}\right\|_{\mathrm{F}}^{2}\right]\right).\end{aligned} \tag{17}$$

Notice that by induction, $\sum_{i=1}^{n} \boldsymbol{y}^{(t)}=\sum_{i=1}^{n} \boldsymbol{g}^{(t)}$. Recalling (9) gives

$$\begin{aligned}&\bar{\boldsymbol{x}}^{(t+1)}-\bar{\boldsymbol{x}}^{(t)}=\frac{\gamma}{n} \sum_{i=1}^{n} \boldsymbol{y}^{(t)}=\frac{\gamma}{n} \sum_{i=1}^{n} \boldsymbol{g}^{(t)}\\ =&\frac{\gamma}{n} \sum_{i=1}^{n}\left[\nabla f_{i}\left(\bar{\boldsymbol{x}}^{(t)}\right)+\left(\nabla f_{i}\left(\boldsymbol{x}_{i}^{(t)}\right)-\nabla f_{i}\left(\bar{\boldsymbol{x}}^{(t)}\right)\right)+\left(\boldsymbol{g}_{i}^{(t)}-\nabla f_{i}\left(\boldsymbol{x}_{i}^{(t)}\right)\right)\right]\\ =&\gamma \nabla f\left(\bar{\boldsymbol{x}}^{(t)}\right)+\frac{\gamma}{n} \sum_{i=1}^{n}\left[\left(\nabla f_{i}\left(\boldsymbol{x}_{i}^{(t)}\right)-\nabla f_{i}\left(\bar{\boldsymbol{x}}^{(t)}\right)\right)+\left(\boldsymbol{g}_{i}^{(t)}-\nabla f_{i}\left(\boldsymbol{x}_{i}^{(t)}\right)\right)\right].\end{aligned} \tag{18}$$

By Assumptions A.1 and A.2, we derive

$$\begin{aligned}&\mathbb{E}\left[\left\|\bar{\boldsymbol{x}}^{(t+1)}-\bar{\boldsymbol{x}}^{(t)}\right\|^{2}\right]\\ =&\gamma^{2} \mathbb{E}\left[\left\|\nabla f\left(\bar{\boldsymbol{x}}^{(t)}\right)+\frac{1}{n} \sum_{i=1}^{n}\left(\nabla f_{i}\left(\boldsymbol{x}_{i}^{(t)}\right)-\nabla f_{i}\left(\bar{\boldsymbol{x}}^{(t)}\right)\right)\right\|^{2}\right]+\frac{\gamma^{2} \sigma^{2}}{n}\\ \leq & 2 \gamma^{2} \mathbb{E}\left[\left\|\nabla f\left(\bar{\boldsymbol{x}}^{(t)}\right)\right\|^{2}\right]+\frac{2 \gamma^{2} L^{2}}{n} \mathbb{E}\left[\left\|\boldsymbol{\Pi} \boldsymbol{X}^{(t)}\right\|_{\mathrm{F}}^{2}\right]+\frac{\gamma^{2} \sigma^{2}}{n}.\end{aligned} \tag{19}$$

Due to $\gamma \leq \frac{1}{L}$, we have

$$\mathbb{E}\left[\left\|\bar{\boldsymbol{x}}^{(t+1)}-\bar{\boldsymbol{x}}^{(t)}\right\|^{2}\right] \leq 2 \gamma^{2} \mathbb{E}\left[\left\|\nabla f\left(\bar{\boldsymbol{x}}^{(t)}\right)\right\|^{2}\right]+\frac{2}{n} \mathbb{E}\left[\left\|\boldsymbol{\Pi} \boldsymbol{X}^{(t)}\right\|_{\mathrm{F}}^{2}\right]+\frac{\gamma^{2} \sigma^{2}}{n}. \tag{20}$$

Then the conclusion follows (17), (20) and $\left\|\bar{\boldsymbol{X}}^{(t+1)}-\bar{\boldsymbol{X}}^{(t)}\right\|_{\mathrm{F}}^{2}=n\left\|\bar{\boldsymbol{x}}^{(t+1)}-\bar{\boldsymbol{x}}^{(t)}\right\|^{2}$. □

Lemma 12 *Consider the DSGT* (9)*. Let Assumptions A.1 and A.2 hold. Suppose that $\left\{\boldsymbol{W}^{(t)}\right\}_{t \geq 0}$ have consensus rate β and $\boldsymbol{y} \in \mathbb{R}^{n}$. If $\frac{48 c_{2} \gamma^{2} L^{2}}{(1-\beta)^{4}} \leq \frac{1}{2}$, then*

$$\begin{aligned}\sum_{t=0}^{T} \mathbb{E}\left[\left\|\boldsymbol{\Pi} \boldsymbol{X}^{(t)}\right\|_{\mathrm{F}}^{2}\right] \leq & 2 \Phi^{(0)}+\frac{48 c_{2} n \gamma^{4} L^{2}}{(1-\beta)^{4}} \sum_{t=1}^{T} \mathbb{E}\left[\left\|\nabla f\left(\bar{\boldsymbol{x}}^{(t-1)}\right)\right\|^{2}\right]+\frac{8 c_{2} \gamma^{2}}{(1-\beta)^{4}}\left\|\nabla \boldsymbol{F}\left(\boldsymbol{X}^{(0)}\right)\right\|_{\mathrm{F}}^{2}\\ &+\frac{24 c_{2} \gamma^{4} L^{2}}{(1-\beta)^{4}}(T+1) \sigma^{2}+\frac{16 c_{1} n \gamma^{2}}{(1-\beta)^{3}}(T+1) \sigma^{2}.\end{aligned}$$

Proof. By the definition of $\Phi^{(t)}$ in (15), we have that for $t \geq 0$,

$$\Phi^{(t+1)} \leq \left(\frac{1+\beta}{2}\right)\Phi^{(t)} + \frac{4c_2\gamma^2}{(1-\beta)^4}\mathbb{E}\left[\left\|\nabla \boldsymbol{F}\big(\boldsymbol{X}^{(t)}\big) - \nabla \boldsymbol{F}\big(\boldsymbol{X}^{(t-1)}\big)\right\|_{\mathrm{F}}^2\right] + \frac{8c_1 n\gamma^2}{(1-\beta)^3}\sigma^2. \quad (21)$$

Then, for $t \geq 0$, by Lemma 10 and (21), we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\boldsymbol{\Pi}\boldsymbol{X}^{(t)}\right\|_{\mathrm{F}}^2\right] \leq \frac{1-\beta}{2}\Phi^{(t)} \\
\leq &\Phi^{(t)} - \Phi^{(t+1)} + \frac{4c_2\gamma^2}{(1-\beta)^4}\mathbb{E}\left[\left\|\nabla \boldsymbol{F}\big(\boldsymbol{X}^{(t)}\big) - \nabla \boldsymbol{F}\big(\boldsymbol{X}^{(t-1)}\big)\right\|_{\mathrm{F}}^2\right] + \frac{8c_1 n\gamma^2}{(1-\beta)^3}\sigma^2.
\end{aligned}$$

For $t \geq 1$, by Lemma 11, we derive

$$\begin{aligned}
&\mathbb{E}\left[\left\|\boldsymbol{\Pi}\boldsymbol{X}^{(t)}\right\|_{\mathrm{F}}^2\right] \leq \Phi^{(t)} - \Phi^{(t+1)} + \frac{24c_2 n\gamma^4 L^2}{(1-\beta)^4}\mathbb{E}\left[\left\|\nabla f\big(\bar{\boldsymbol{x}}^{(t-1)}\big)\right\|^2\right] \\
&+ \frac{36c_2\gamma^2 L^2}{(1-\beta)^4}\mathbb{E}\left[\left\|\boldsymbol{\Pi}\boldsymbol{X}^{(t-1)}\right\|_{\mathrm{F}}^2\right] + \frac{12c_2\gamma^2 L^2}{(1-\beta)^4}\mathbb{E}\left[\left\|\boldsymbol{\Pi}\boldsymbol{X}^{(t)}\right\|_{\mathrm{F}}^2\right] \\
&+ \frac{12c_2\gamma^4 L^2}{(1-\beta)^4}\sigma^2 + \frac{8c_1 n\gamma^2}{(1-\beta)^3}\sigma^2.
\end{aligned} \quad (22)$$

It follows by the definition of $\Phi^{(t)}$ that

$$\Phi^{(1)} \leq \left(\frac{1+\beta}{2}\right)\Phi^{(0)} + \frac{4c_2\gamma^2}{(1-\beta)^4}\left\|\nabla \boldsymbol{F}\big(\boldsymbol{X}^{(0)}\big)\right\|_{\mathrm{F}}^2 + \frac{8c_1 n\gamma^2}{(1-\beta)^3}\sigma^2.$$

By Lemma 10, we obtain

$$\left\|\boldsymbol{\Pi}\boldsymbol{X}^{(0)}\right\|_{\mathrm{F}}^2 \leq \frac{1-\beta}{2}\Phi^{(0)} \leq \Phi^{(0)} - \Phi^{(1)} + \frac{4c_2\gamma^2}{(1-\beta)^4}\left\|\nabla \boldsymbol{F}\big(\boldsymbol{X}^{(0)}\big)\right\|_{\mathrm{F}}^2 + \frac{8c_1 n\gamma^2}{(1-\beta)^3}\sigma^2. \quad (23)$$

Taking sum on both sides of (22) and noting that $\frac{48c_2\gamma^2 L^2}{(1-\beta)^4} \leq \frac{1}{2}$, $\Phi^{(T+1)} \geq 0$, the lemma is proved. □

Lemma 13 is standard in the analysis of gradient tracking methods. We attach its proof for completeness.

Lemma 13 *Consider the DSGT* (9). *Let Assumptions A.1 and A.2 hold. If* $\gamma \leq \frac{1}{4L}$, *then*

$$\mathbb{E}\left[f\big(\bar{\boldsymbol{x}}^{(t+1)}\big)\right] \leq \mathbb{E}\left[f\big(\bar{\boldsymbol{x}}^{(t)}\big)\right] - \frac{\gamma}{4}\mathbb{E}\left[\left\|\nabla f\big(\bar{\boldsymbol{x}}^{(t)}\big)\right\|^2\right] + \frac{\gamma L^2}{n}\mathbb{E}\left[\left\|\boldsymbol{\Pi}\boldsymbol{X}^{(t)}\right\|_{\mathrm{F}}^2\right] + \frac{\gamma^2 L}{2n}\sigma^2.$$

Proof. By Assumption A.1, we have

$$\mathbb{E}\left[f\big(\bar{\boldsymbol{x}}^{(t+1)}\big)\right] \leq \mathbb{E}\left[f\big(\bar{\boldsymbol{x}}^{(t)}\big)\right] - \mathbb{E}\left[\left\langle \nabla f\big(\bar{\boldsymbol{x}}^{(t)}\big), \bar{\boldsymbol{x}}^{(t+1)} - \bar{\boldsymbol{x}}^{(t)}\right\rangle\right] + \frac{L}{2}\mathbb{E}\left[\left\|\bar{\boldsymbol{x}}^{(t+1)} - \bar{\boldsymbol{x}}^{(t)}\right\|^2\right].$$

It follows from (9) that

$$\begin{aligned}
&\mathbb{E}\left[\left\langle \nabla f\left(\bar{\boldsymbol{x}}^{(t)}\right), \bar{\boldsymbol{x}}^{(t+1)} - \bar{\boldsymbol{x}}^{(t)}\right\rangle\right] \\
=&\gamma\mathbb{E}\left[\left\langle \nabla f\left(\bar{\boldsymbol{x}}^{(t)}\right), \nabla f\big(\bar{\boldsymbol{x}}^{(t)}\big) + \frac{1}{n}\sum_{i=1}^{n}\left(\nabla f_i\big(\boldsymbol{x}_i^{(t)}\big) - \nabla f_i\big(\bar{\boldsymbol{x}}^{(t)}\big)\right)\right\rangle\right] \\
\geq&\gamma\mathbb{E}\left[\left\|\nabla f\big(\bar{\boldsymbol{x}}^{(t)}\big)\right\|^2\right] - \frac{\gamma}{2}\mathbb{E}\left[\left\|\nabla f\big(\bar{\boldsymbol{x}}^{(t)}\big)\right\|^2\right] - \frac{\gamma}{2n}\sum_{i=1}^{n}\mathbb{E}\left[\left\|\nabla f_i\big(\boldsymbol{x}_i^{(t)}\big) - \nabla f_i\big(\bar{\boldsymbol{x}}^{(t)}\big)\right\|\right]^2 \\
\geq&\frac{\gamma}{2}\mathbb{E}\left[\left\|\nabla f\big(\bar{\boldsymbol{x}}^{(t)}\big)\right\|^2\right] - \frac{\gamma L^2}{2n}\mathbb{E}\left[\left\|\boldsymbol{\Pi}\boldsymbol{X}^{(t)}\right\|_{\mathrm{F}}\right]^2,
\end{aligned}$$

where the first equality is by Assumption A.2 and (18); the second inequality is by Assumption A.1.

Recalling (19) yields

$$
\begin{aligned}
&\mathbb{E}\left[f\left(\bar{\boldsymbol{x}}^{(t+1)}\right)\right] \\
\leq &\mathbb{E}\left[f\left(\bar{\boldsymbol{x}}^{(t)}\right)\right] - \frac{\gamma}{2}\left(1-2\gamma L\right)\mathbb{E}\left[\left\|\nabla f\left(\bar{\boldsymbol{x}}^{(t)}\right)\right\|^2\right] + \frac{\gamma L^2}{2n}\left(1+2\gamma L\right)\mathbb{E}\left[\left\|\boldsymbol{\Pi}\boldsymbol{X}^{(t)}\right\|_{\mathrm{F}}^2\right] + \frac{\gamma^2\sigma^2 L}{2n} \\
\leq &\mathbb{E}\left[f\left(\bar{\boldsymbol{x}}^{(t)}\right)\right] - \frac{\gamma}{4}\mathbb{E}\left[\left\|\nabla f\left(\bar{\boldsymbol{x}}^{(t)}\right)\right\|^2\right] + \frac{\gamma L^2}{n}\mathbb{E}\left[\left\|\boldsymbol{\Pi}\boldsymbol{X}^{(t)}\right\|_{\mathrm{F}}^2\right] + \frac{\gamma^2\sigma^2 L}{2n}.
\end{aligned}
$$

The lemma is proved. □

Referring to Lemma 26 of [11], we have the following result.

Lemma 14 *Let A, B, C, T and α be positive constants. Define*

$$
g(\gamma) = \frac{A}{\gamma T} + B\gamma + C\gamma^2.
$$

Then

$$
\inf_{\gamma\in(0,\alpha]} g\left(\gamma\right) \leq 2\left(\frac{AB}{T}\right)^{\frac{1}{2}} + 2C^{\frac{1}{3}}\left(\frac{A}{T}\right)^{\frac{2}{3}} + \frac{A}{\alpha T}.
$$

Proof of Theorem 6 Define $f^* = \inf_x f(x)$, $F_0 = f\left(\bar{\boldsymbol{x}}^{(0)}\right) - f^*$, $C_0 = \left\|\boldsymbol{\Pi}\boldsymbol{X}^{(0)}\right\|_{\mathrm{F}}^2$ and $D_0 = \sum_{i=1}^n \left\|\nabla f_i\left(\boldsymbol{x}_i^{(0)}\right)\right\|^2$. We show that

$$
\begin{aligned}
&\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}\left[\left\|\nabla f\left(\bar{\boldsymbol{x}}^{(t)}\right)\right\|_{\mathrm{F}}^2\right] \\
\leq &\mathcal{O}\left(\sqrt{\frac{F_0 L\sigma^2}{nT}} + \frac{1}{1-\beta}\left(\frac{F_0 L\sigma}{T}\right)^{\frac{2}{3}} + \frac{F_0}{\left(1-\beta\right)^2 T} + \frac{L^2 C_0}{\left(1-\beta\right)^2 nT} + \frac{D_0}{nT}\right),
\end{aligned}
$$

Proof. Let $\gamma \leq \frac{(1-\beta)^2}{50L}$ to satisfy the conditions in Lemmas 12 and 13. Then, we have

$$
\begin{aligned}
&\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}\left[\left\|\nabla f\left(\bar{\boldsymbol{x}}^{(t)}\right)\right\|_{\mathrm{F}}^2\right] \\
\leq &\frac{4}{\gamma\left(T+1\right)}\left(f\left(\bar{\boldsymbol{x}}^{(0)}\right) - f^*\right) + \frac{4L^2}{n\left(T+1\right)}\sum_{t=0}^{T}\mathbb{E}\left[\left\|\boldsymbol{\Pi}\boldsymbol{X}^{(0)}\right\|_{\mathrm{F}}^2\right] + \frac{2\gamma L}{n}\sigma^2 \\
\leq &\frac{4}{\gamma T}\left(f\left(\bar{\boldsymbol{x}}^{(0)}\right) - f^*\right) + \frac{2\gamma L}{n}\sigma^2 + \frac{8L^2}{nT}\Phi^{(0)} + \frac{32c_2\gamma^2 L^2}{\left(1-\beta\right)^4 nT}\left\|\nabla\boldsymbol{F}\left(\boldsymbol{X}^{(0)}\right)\right\|_{\mathrm{F}}^2 \\
&+ \frac{192c_2\gamma^4 L^4}{\left(1-\beta\right)^4\left(T+1\right)}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla f\left(\bar{\boldsymbol{x}}^{(t-1)}\right)\right\|^2\right] + \frac{96c_2\gamma^4 L^4}{\left(1-\beta\right)^4 n}\sigma^2 + \frac{64c_1\gamma^2 L^2}{\left(1-\beta\right)^3}\sigma^2.
\end{aligned}
$$

If $\gamma \leq \frac{1-\beta}{10L}$, then $\frac{192c_2\gamma^4 L^4}{(1-\beta)^4} \leq \frac{1}{2}$. Then, we have

$$
\begin{aligned}
\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}\left[\left\|\nabla f\left(\bar{\boldsymbol{x}}^{(t)}\right)\right\|_{\mathrm{F}}^2\right] \leq &\frac{8}{\gamma T}\left(f\left(\bar{\boldsymbol{x}}^{(0)}\right) - f^*\right) + \frac{4\gamma L}{n}\sigma^2 + \frac{192c_2\gamma^4 L^4}{\left(1-\beta\right)^4 n}\sigma^2 \\
&+ \frac{16L^2}{nT}\Phi^{(0)} + \frac{64c_2\gamma^2 L^2}{\left(1-\beta\right)^4 nT}\left\|\nabla\boldsymbol{F}\left(\boldsymbol{X}^{(0)}\right)\right\|_{\mathrm{F}}^2 + \frac{128c_1\gamma^2 L^2}{\left(1-\beta\right)^3}\sigma^2.
\end{aligned} \tag{24}
$$

By (16) and $c_1 = 4$, $c_2 = 16$ defined in Lemma 8, if $\gamma \leq \frac{(1-\beta)^2}{50L}$ and $T \geq \frac{1}{1-\beta}$, we have

$$
\begin{aligned}
&\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}\left[\left\|\nabla f\left(\bar{\boldsymbol{x}}^{(t)}\right)\right\|_{\mathrm{F}}^2\right] \\
\leq &\frac{8}{\gamma T}\left(f\left(\bar{\boldsymbol{x}}^{(0)}\right)-f^*\right)+\frac{4\gamma L}{n}\sigma^2 \\
&+\frac{1}{T}\left(\frac{64L^2}{(1-\beta)^2 n}\left\|\boldsymbol{\Pi}\boldsymbol{X}^{(0)}\right\|_{\mathrm{F}}^2+\frac{1024\gamma^2L^2}{(1-\beta)^4 n}\left\|\nabla\boldsymbol{F}\left(\boldsymbol{X}^{(0)}\right)\right\|_{\mathrm{F}}^2\right) \\
&+\frac{2048\gamma^2L^2}{(1-\beta)^4 T}\sigma^2+\frac{3072\gamma^4L^4}{(1-\beta)^4 n}\sigma^2+\frac{512\gamma^2L^2}{(1-\beta)^3}\sigma^2 \\
\leq &\frac{8}{\gamma T}\left(f\left(\bar{\boldsymbol{x}}^{(0)}\right)-f^*\right)+\frac{5\gamma L}{n}\sigma^2+\frac{2560\gamma^2L^2}{(1-\beta)^3}\sigma^2 \\
&+\frac{1}{nT}\left(\frac{64L^2}{(1-\beta)^2}\left\|\boldsymbol{\Pi}\boldsymbol{X}^{(0)}\right\|_{\mathrm{F}}^2+\left\|\nabla\boldsymbol{F}\left(\boldsymbol{X}^{(0)}\right)\right\|_{\mathrm{F}}^2\right).
\end{aligned}
\tag{25}
$$

To meet the conditions of Lemma 12, Lemma 13, (24), (25), it suffices to let $\gamma \leq \frac{(1-\beta)^2}{50L}$. Then, the result follows by setting $g(\gamma)$ to be the RHS of (25), $A = 8\left(f\left(\bar{\boldsymbol{x}}^{(0)}\right)-f^*\right)$, $B = \frac{5L}{n}\sigma^2$, $C = \frac{2560L^2}{(1-\beta)^3}\sigma^2$ and $\alpha = \frac{(1-\beta)^2}{50L}$ in Lemma 14. □

D Numerical Experiments

D.1 Network-size independent consensus rate

In this experiment, we set $M = 5\ln(n)$ for D-EquiStatic and $M = 2\ln(n)$ for U-EquiStatic, which is consistent with Theorems 1 and 3. For OD-EquiDyn and OU-EquiDyn, we set $M = 5\ln(n)$ and $\eta = 0.5$. Fig. 9 shows that the consensus rate is independent of the network size for all EquiTopo graphs. The results are obtained by averaging over 3 independent random experiments.

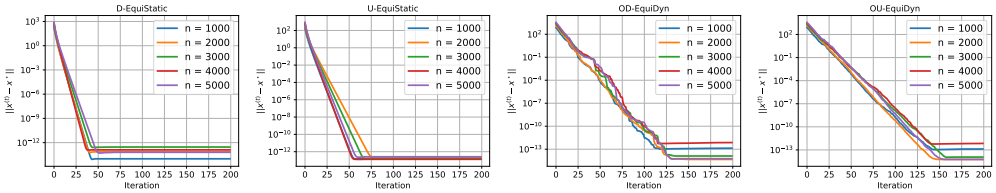

Figure 9: The EquiTopo graphs can achieve network-size independent consensus rates.

D.2 Comparison with other topologies

We compare the consensus rate between topologies with one-peer or $\Theta(\ln(n))$ neighbors on network-size $n = 300$ and $n = 4900$. In the one-peer case, each topology has exactly one neighbor. For OD-EquiDyn and OU-EquiDyn, we set $M = n-1$ and $\eta = 0.5$. In the $\Theta(\ln(n))$ neighbors case, we set $M = 9$ and $M = 13$ for $n = 300$ and $n = 4900$, respectively, so that the number of neighbors is identical to the static exponential graph for a fair comparison. The results are obtained by averaging over 10 and 3 independent random experiments for $n = 300$ and $n = 4900$, respectively.

D.3 DSGD with EquiTopo

Least-square The distributed least square problems are defined with $f_i(\boldsymbol{x}) = \|\boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{b}_i\|^2$, in which $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{A}_i \in \mathbb{R}^{K\times d}$. In the simulation, we let $d = 10$ and $K = 50$. At node i, we generate each element in $\boldsymbol{A}_i$ following standard normal distribution. Measurement $\boldsymbol{b}_i$ is generated by $\boldsymbol{b}_i = \boldsymbol{A}_i\boldsymbol{x}^\star + \boldsymbol{s}_i$ with a given arbitrary $\boldsymbol{x}^\star \in \mathbb{R}^d$ where $\boldsymbol{s}_i \sim \mathcal{N}(0, \sigma_s^2\boldsymbol{I})$ is some white noise. At

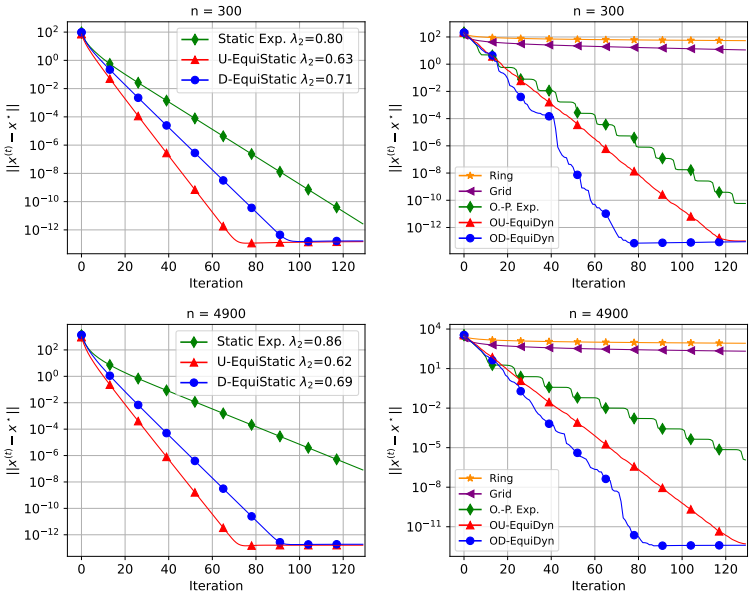

Figure 10: Consensus rate comparison among different network topologies in average consensus problem. Left: all graphs are with $\Theta(\ln(n))$ degree. Right: all graphs are with $\Theta(1)$ degree.

each iteration t, each node will generate a stochastic gradient via $\widehat{\nabla f}_i(\boldsymbol{x}) = \nabla f_i(\boldsymbol{x}) + \boldsymbol{n}_i$ where $\boldsymbol{n}_i \sim \mathcal{N}(0, \sigma_n^2 \boldsymbol{I})$ is a white gradient noise. By adjusting constant σ_n, we can control the noise variance. In this experiment, we set $\sigma_s = 0.1$ and $\sigma_n = 1$. The network size n is 300, and we set $M = 9$ so that D/U-EquiStatic has the same degree as the static exponential graph. After fixing M, we sample the basis until the second-largest eigenvalue of the gossip matrix is small enough. The initial learning rate is 0.037 and decays by 1.4 every 40 iterations. The results are obtained by averaging over 10 independent random experiments.

Deep learning

MNIST. We utilize EquiTopo graphs in DSGD to solve the image classification task with CNN over MNIST dataset [15]. Like the CIFAR-10 experiment, we utilize BlueFog [38] to support decentralized communication and topology settings in a cluster of 17 Tesla P100 GPUs. The network architecture is defined by a two-layer convolutional neural network with kernel size 5 followed by two feed-forward layers. Each convolutional layer contains a max pooling layer and a Rectified Linear Unit (ReLu). We generate D/U-EquiStaic with $M = 4$ and sample OD/OU-EquiDyn with $M = 16$ and $\eta = 0.53$. The local batch size is 64, momentum is 0.5, the learning rate is 0.01, and we train for 20 epochs. Centralized SGD and Ring are included for comparison. See Fig. 11 for the training loss and test accuracy of D/U-EquiStaic and OD/OU-EquiDyn graphs. See Table 5 for the test accuracy calculated by averaging over last 3 epochs. EquiTopo graphs achieve competitive train loss and test accuracy to centralized SGD.

CIFAR-10. We use the ResNet-20 model implemented by [10]. In this experiment, we train for 130 epochs with local batch size 8, momentum 0.9, weight decay 10^{-4}, and the initial learning rate 0.01, which is divided by 10 at 50th, 100th, and 120th epochs. We follow the data augmentation from [14], a 4×4 padding followed by a random horizontal flip and a 32×32 random crop. We generate D/U-EquiStaic with $M = 5$ and sample OD/OU-EquiDyn with $M = 16$ and $\eta = 0.53$. See Fig. 12 for the training loss and test accuracy of OD/OU-EquiDyn. See Table 5 for the test accuracy calculated by averaging over last 5 epochs.

D.4 DSGT with EquiTopo

In addition to the DSGD experiments, we apply the OD/OU-EquiDyn graphs to the DSGT algorithm when solving logistic regression with non-convex regularizations, i.e., $f_i(\boldsymbol{x}) = \frac{1}{L}\sum_{\ell=1}^{L} \ln(1 + \exp(-y_{i,\ell}\boldsymbol{h}_{i,\ell}^T \boldsymbol{x})) + R\sum_{j=1}^{d} x_{[j]}^2/(1 + x_{[j]}^2)$ where $x_{[j]}$ is the j-th element of $\boldsymbol{x}$, and $\{\boldsymbol{h}_{i,\ell}, y_{i,\ell}\}_{\ell=1}^{L}$

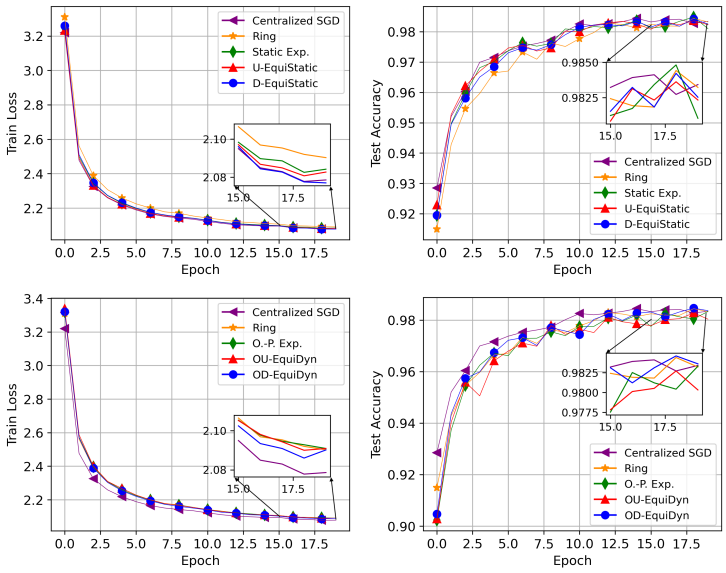

Figure 11: Train loss and test accuracy comparisons among different topologies for CNN on MNIST.

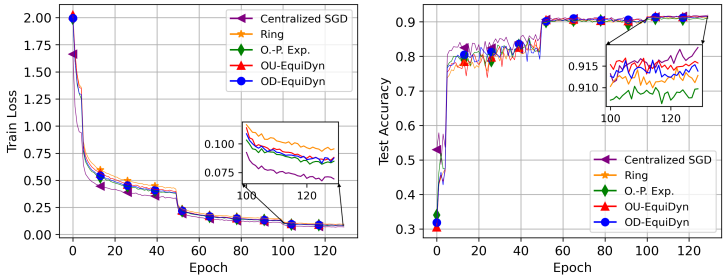

Figure 12: Train loss and test accuracy comparisons among different topologies for ResNet-20 on CIFAR-10.

Table 5: Comparison of test accuracy(%) with different topologies over MNIST and CIFAR-10 datasets.

Topology	MNIST Acc.	CIFAR-10 Acc.
Centralized SGD	98.34	91.76
Ring	98.32	91.25
Static Exp.	98.31	91.48
O.-P. Exp.	98.17	90.86
D-EquiStatic	98.29	**92.01**
U-EquiStatic	98.26	91.74
OD-EquiDyn	**98.39**	91.44
OU-EquiDyn	98.12	91.56

is the data kept by node i. Data heterogeneity exists when local data ξ_i follows different distributions $\mathcal{D}_i$ in problem (7). To control data heterogeneity across the nodes, we first let each node i be associated with a local solution $\boldsymbol{x}_i^\star$, and such $\boldsymbol{x}_i^\star$ is generated by $\boldsymbol{x}_i^\star = \boldsymbol{x}^\star + \boldsymbol{v}_i$ where $\boldsymbol{x}^\star \sim \mathcal{N}(0, \boldsymbol{I}_d)$ is a randomly generated vector while $\boldsymbol{v}_i \sim \mathcal{N}(0, \sigma_h^2 \boldsymbol{I}_d)$ controls the similarity between each local solution. Generally speaking, a large σ_h^2 results in local solutions $\{\boldsymbol{x}_i^\star\}$ that are vastly different from each other. With $\boldsymbol{x}_i^\star$ at hand, we can generate local data that follows distinct distributions. At node i, we generate each feature vector $\boldsymbol{h}_{i,\ell} \sim \mathcal{N}(0, \boldsymbol{I}_d)$. To produce the corresponding label $y_{i,\ell}$, we generate a random variable $z_{i,\ell} \sim \mathcal{U}(0,1)$. If $z_{i,\ell} \leq 1 + \exp(-y_{i,\ell}\boldsymbol{h}_{i,\ell}^T \boldsymbol{x}_i^\star)$, we set $y_{i,\ell} = 1$; otherwise $y_{i,\ell} = -1$. Clearly, solution $\boldsymbol{x}_i^\star$ controls the distribution of the labels. This way, we can easily control data heterogeneity by adjusting σ_h^2. Furthermore, to easily control the influence of

gradient noise, we will achieve the stochastic gradient by imposing a Gaussian noise to the real gradient, i.e., $\widehat{\nabla f}_i(\boldsymbol{x}) = \nabla f_i(\boldsymbol{x}) + \boldsymbol{s}_i$ in which $\boldsymbol{s}_i \sim \mathcal{N}(0, \sigma_n^2 \boldsymbol{I}_d)$. We can control the magnitude of the gradient noise by adjusting σ_n^2.

We let $d = 10$, $L = 1000$, $n = 300$, $R = 0.001$, and $\sigma_h = 0.2$ in the simulation. For OD/OU-EquiDyn, we set $M = n - 1$ and $\eta = 0.5$. The learning rate for OD/OU-EquiDyn and O.-P. Exp. is 3 and 1.5, respectively so that all of them converge to the same level of accuracy. The left plot in Fig. 13 depicts the performance of different one-peer graphs in DSGT. The right plot in Fig. 13 illustrates how O.-P. Exp. behaves if it has the same learning rate 3 as OU/OD-EquiDyn. The gradient norm is used as a metric to gauge the convergence performance. The results are calculated by averaging over 10 independent random experiments. It is observed that OD/OU-EquiDyn converges faster than a one-peer exponential graph.

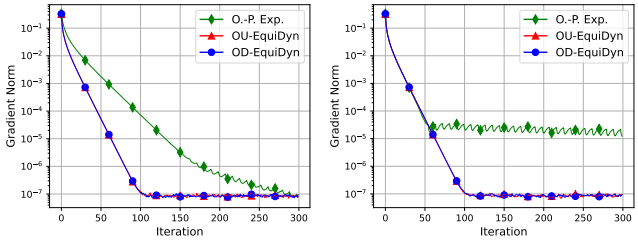

Figure 13: OD/OU-EquiDyn in DSGT. Left: The learning rates for O.-P. Exp. and OU/OD-EquiDyn are 1.6 and 3, respectively, so that all algorithms achieve the same accuracy. Right: The learning rates for all algorithms are 3 so that they share the same convergence rate in the initial stage.