SIAM J. APPL. MATH.
Vol. 82, No. 4, pp. 1558–1586

EFFICIENT AND RELIABLE OVERLAY NETWORKS FOR DECENTRALIZED FEDERATED LEARNING*

YIFAN HUA†, KEVIN MILLER‡, ANDREA L. BERTOZZI§, CHEN QIAN¶, AND BAO WANG‖

Abstract. We propose near-optimal overlay networks based on d-regular expander graphs to accelerate decentralized federated learning (DFL) and improve its generalization. In DFL a massive number of clients are connected by an overlay network, and they solve machine learning problems collaboratively without sharing raw data. Our overlay network design integrates spectral graph theory and the theoretical convergence and generalization bounds for DFL. As such, our proposed overlay networks accelerate convergence, improve generalization, and enhance robustness to client failures in DFL with theoretical guarantees. Also, we present an efficient algorithm to convert a given graph to a practical overlay network and maintain the network topology after potential client failures. We numerically verify the advantages of DFL with our proposed networks on various benchmark tasks, ranging from image classification to language modeling using hundreds of clients.

Key words. decentralized federated learning, overlay networks, random graphs

MSC codes. 65B99, 68T01, 68T09, 68W15

DOI. 10.1137/21M1465081

1. Introduction.

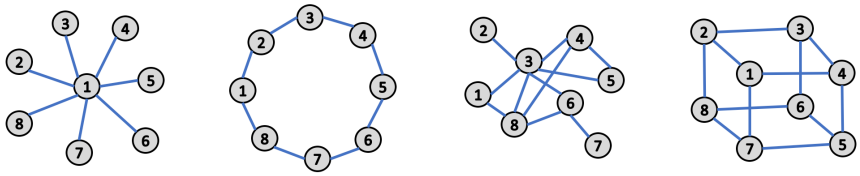
Federated learning (FL) is a machine learning (ML) setting where a massive number of entities (clients) solve an ML problem collaboratively without transferring raw data, under the coordination of a central server [38, 23]. FL trains ML models by exchanging the model parameters between clients and the central server; in each communication round, the central server distributes parameters to clients and aggregates the updated parameters from clients. FL decouples the model training from the need for collecting or direct access to the private training data; therefore, FL significantly reduces privacy and security risks. Many algorithms have been developed for FL, such as FedAvg [38], SCAFFOLD [24], FedProx [27], FedPD [72], FedSplit [46], and FedOpt [51]. Compared to many distributed optimization settings [42, 37, 6, 71, 48, 16, 53], FL has tremendous advantages in communication efficiency. We can mathematically formulate FL as solving the following optimization problem:

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w}) := \frac{1}{N}\sum_{i=1}^{N} f_i(\boldsymbol{w}), \tag{1.1}$$

*Received by the editors December 13, 2021; accepted for publication (in revised form) May 27, 2022; published electronically August 18, 2022.
https://doi.org/10.1137/21M1465081
Funding: This material is based on research sponsored by the NSF grants DMS-1924935, DMS-1952339, DMS-2110145, DMS-2152717, DMS-2152762, DMS-2208361, CNS-1750704, CNS-1932447, and CNS-2114113 and the DOE grant DE-SC0021142. The second author is supported by the DOD National Defense Science and Engineering Graduate (NDSEG) Research Fellowship.

†Co–first author. Department of Computer Science and Engineering, University of California at Santa Cruz, Santa Cruz, CA 95064 USA (yhua294@ucsc.edu).
‡Co–first author. Department of Mathematics, University of California at Los Angeles, Los Angeles, CA 90095 USA (millerk22@math.ucla.edu).
§Department of Mathematics, University of California at Los Angeles, Los Angeles, CA 90095 USA (bertozzi@math.ucla.edu).
¶Department of Computer Science and Engineering, University of California at Santa Cruz, Santa Cruz, CA 95064 USA (cqian12@ucsc.edu).
‖Department of Mathematics, Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT 84112 USA (wangbaonj@gmail.com).

where $f_i(\boldsymbol{w}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_i}\mathcal{L}(g(\boldsymbol{x},\boldsymbol{w}),y)$ with $(\boldsymbol{x},y)$ being a data-label pair sampled from the data distribution $\mathcal{D}_i$ on the ith client, and $g(\cdot,\boldsymbol{w})$ is the ML model. As shown in Figure 1(a), in the ith communication round, FedAvg [38], one of the most popular FL algorithms, iterates as follows: the server (node 1) sends the current parameters $\boldsymbol{w}_i$ to a small fraction of selected clients $\{k_j|k_j \in \{1,2,\ldots,N\}$ for $j = 1,2,\ldots,m\}$. Each selected client then updates $\boldsymbol{w}_i$ for T iterations by using its local data and stochastic gradient-based algorithms. The server then aggregates these locally updated parameters to get the updated model after the current communication round. The existence of the central server raises several concerns about FL: (1) the communication cost between the server and clients can be excessive since a large number of clients are involved in a practical FL system, (2) the failure of the server would disrupt the training process of all clients, and (3) the privacy of the whole FL system can be fragile since the central server is exposed to adversaries.



(a) FL (b) Ring (c) Erdös-Rényi (d) Expander

FIG. 1. *Illustration of the network topology for federated learning and decentralized federated learning with Ring, Erdös–Rényi, and expander graphs.*

Decentralized federated learning (DFL) replaces the server-clients communication with client-client (peer-to-peer) communication, which significantly reduces the communication burden and privacy risks [7, 18, 45, 61, 60, 67, 33, 39, 29, 9, 31, 62, 66, 30, 1, 59]. In DFL, all clients are connected by an overlay network, e.g., Figure 1(b) Ring, (c) Erdös–Rényi, and (d) d-regular expander graphs. The clients update in the same way as that in FL, and each client only sends its locally updated model to its topological neighbors and aggregates the updated models from its neighbors. Network topology has a profound impact on the convergence, generalization, and robustness of DFL. In this paper, *we focus on designing efficient network topologies that guarantee fast and accurate DFL and are resilient to client failures.*

1.1. Our contribution. Based on the theoretical convergence rate [68, 58, 59] and our first established generalization bound of DFL, where each client trains ML models using stochastic gradient descent with momentum, we design near-optimal network topology to connect clients to train ML models collectively. In particular, leveraging random graph theory, we propose d-regular expander graphs for the network topologies, which is provably near-optimal. The major advantages of leveraging d-regular expander graphs for the overlay networks design are threefold:

- DFL with d-regular expander graphs converges remarkably faster and generalizes better than DFL using other sparse graphs, including Ring and Erdö–Rényi graphs.
- Expander graphs connect each node with d neighbors, resulting in low communication cost in DFL.
- DFL with d-regular expander graphs enables robust DFL with respect to potential client (node) failures.

1.2. Additional related works.

Network design. Chow et al. [10] have designed expander graphs for decentralized optimization using deterministic local optimization algorithms. In [36], the authors

use theory of the max-plus linear systems and design-efficient topology for cross-silo FL, in which close-by data silos can exchange information faster with the central server. We focus on designing efficient networks for cross-device DFL that are scalable to a massive number of devices.

Analysis of DFL/FL algorithms. The convergence properties of FedAvg or local stochastic gradient descent (SGD) have been studied extensively [63, 57, 22], mainly focusing on the independent and identically distributed (IID) case. Non-IID convergence for FL has been shown in [22, 72, 68, 28]. Convergence analysis of DFL has been shown in [64, 59]. While convergence analysis for the myriad of problem setups has been provided, generalization guarantees have been more elusive.

Convergence analysis of DFL hinges on connectedness properties of the underlying graph topology, captured in the spectral properties of the associated mixing matrix (see section 2). The authors of [63] discuss how different versions of local SGD correspond to different graph topologies, and [64] provides an efficient decomposition of graph topology for improved communication costs.

Practical network construction. Building overlay networks has been studied extensively in previous works. However, in the past, overlay networks are mainly used for peer-to-peer file sharing [34], online social networks [20], and routing infrastructures [25, 49]. For peer-to-peer file-sharing networks, existing studies have proposed utilizing random walks to achieve distributed d-regular expander graphs by assuming each node could choose d neighbors at random [15, 26]. However, such an assumption does not hold in DFL because no node can uniformly choose d neighbors among existing nodes at random since there is no central coordinator. However, it is possible to build an expander graph with tight connectivity if the global information is given, such as maintaining distributed Delaunay triangulation graphs for wireless sensor networks [25], metro Ethernet [49], random regular graphs for data center networks [69], and memory interconnection networks [44].

1.3. Notation. We denote scalars by lowercase or uppercase letters and vectors and matrices by lowercase and uppercase boldface letters, respectively. For a vector $\boldsymbol{x} = (x_1, \ldots, x_d)^\top \in \mathbb{R}^d$, we use $\|\boldsymbol{x}\| := (\sum_{i=1}^d |x_i|^2)^{1/2}$ and $\|\boldsymbol{x}\|_\infty := \max_{i=1}^d |x_i|$ to denote its ℓ_2- and ℓ_∞-norm, respectively. We denote the vector whose entries are all 0's as $\mathbf{0}$. For a matrix $\boldsymbol{A}$, we use $\boldsymbol{A}^\top$, $\boldsymbol{A}^{-1}$, and $\|\boldsymbol{A}\|$ to denote its transpose, inverse, and spectral norm, respectively. We denote the identity matrix as $\boldsymbol{I}$. For a function $f(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$, we denote $\nabla f(\boldsymbol{x})$ as its gradient. Given two sequences $\{a_n\}$ and $\{b_n\}$, we write $a_n = \mathcal{O}(b_n)$ and $a_n = \Omega(b_n)$ if there exist positive constants C such that $a_n \le Cb_n$ and $a_n \ge Cb_n$ for $n \ge n_0$, respectively.

1.4. Organization. We organize this paper as follows: In section 2, we present the theoretical results for DFL on the convergence rate and generalization bound. According to these theoretical results we present our network topology design and its practical implementation in sections 3 and 4, respectively. We verify the efficiency and robustness to the potential node failures of DFL with the designed network topology on various benchmarks in section 5. Technical proofs are provided in the appendix.

2. Theory of DFedAvg. An important notion in DFL is the *mixing matrix*, which is associated with an undirected connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with vertex set $\mathcal{V} = \{1, 2, \ldots, N\} := [N]$ and edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, and the edge $(i, j) \in \mathcal{E}$ represents a communication channel between clients i and j.

DEFINITION 2.1 (mixing matrix). *A matrix $\boldsymbol{M} = [m_{i,j}] \in \mathbb{R}^{N \times N}$ is a mixing matrix if it satisfies* 1. *(Graph) If $i \ne j$ and $(i, j) \notin \mathcal{E}$, then $m_{i,j} = 0$; otherwise,*

$m_{i,j} > 0$; 2. (Symmetry) $\boldsymbol{M} = \boldsymbol{M}^\top$; 3. (Null space property) null$\{\boldsymbol{I} - \boldsymbol{M}\} =$ span$\{\mathbf{1}\}$, where $\boldsymbol{I} \in \mathbb{R}^{N\times N}$ and $\mathbf{1} \in \mathbb{R}^N$ are the identity matrix and the vector whose entries are all 1's; 4. (Spectral property) $\boldsymbol{I} \succeq \boldsymbol{M} \succ -\boldsymbol{I}$, where $\boldsymbol{I} \succeq \boldsymbol{M}$ means $\boldsymbol{I} - \boldsymbol{M}$ is positive semidefinite and $\boldsymbol{M} \succ -\boldsymbol{I}$ means $\boldsymbol{M} + \boldsymbol{I}$ is positive definite.

Given the adjacency matrix of a network, its maximum-degree matrix and Metropolis–Hastings matrix are both mixing matrices [8]. The symmetric property of $\boldsymbol{M}$ indicates that its eigenvalues are real and can be sorted in nonincreasing order. Let $\lambda_i(\boldsymbol{M})$ denote the ith largest eigenvalue of $\boldsymbol{M}$; then we have $\lambda_1(\boldsymbol{M}) = 1 > \lambda_2(\boldsymbol{M}) \geq \cdots \geq \lambda_N(\boldsymbol{M}) > -1$ based on the spectral property of the mixing matrix. The mixing matrix also serves as a probability transition matrix of a Markov chain. An important constant is $\lambda = \lambda(\boldsymbol{M}) := \max\{|\lambda_2(\boldsymbol{M})|, |\lambda_N(\boldsymbol{M})|\}$, which describes the speed of the Markov chain, induced by the mixing matrix $\boldsymbol{M}$, as it converges to its stable state.

We consider DFL using the following update on client i:

$$\boldsymbol{w}_i^{t,k+1} = \boldsymbol{w}_i^{t,k} - \eta_t \nabla f_i(\boldsymbol{w}_i^{t,k}; \xi_i^{t,k}) + \beta(\boldsymbol{w}_i^{t,k} - \boldsymbol{w}_i^{t,k-1}), \tag{2.1}$$

where t is the communication round, k is the local iteration, and $\xi_i^{t,k} = (\boldsymbol{x}_i^{t,k}, y_i^{t,k}) \sim \mathcal{D}_i$. After the Kth local iteration, communication happens according to the graph topology of the mixing matrix, $\boldsymbol{M}$; that is, we have for each $i \in [N]$

$$\boldsymbol{w}_i^{t+1,0} = \sum_{\ell=1}^{N} m_{i,\ell} \boldsymbol{w}_\ell^{t,K}.$$

To ensure well-defined iterations, we set $\boldsymbol{w}_i^{t,-1} = \boldsymbol{w}_i^{t,0}$ for each i. These iterations are referred to as DFedAvgM (decentralized federated averaging with momentum) [59] because we apply heavy-ball momentum [47] to the local SGD updates. Including the momentum term with parameter $\beta \in (0, 1)$ can accelerate ML in practice with provable acceleration when the objective function is quadratic or under other specific circumstances [47, 65, 32].

To guarantee convergence of generalization of DFedAvgM, we collect below the necessary assumptions on the local functions f_i and global function f.

Assumption 1 (L-smooth). *$f_1, \ldots, f_N$ are all L-smooth, i.e., $f_i(\boldsymbol{w}) \leq f_i(\boldsymbol{v}) + \langle \nabla f_i(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v}\rangle + \frac{L}{2}\|\boldsymbol{w} - \boldsymbol{v}\|_2^2$ for all $\boldsymbol{w}, \boldsymbol{v}$.*

Assumption 2 (bounded local gradient variance (BLGV)). *Let $\xi_i^t := (\boldsymbol{x}_i^{t,k}, y_i^{t,k})$ be sampled from the ith device's local data $\mathcal{D}_i$ uniformly at random. Then for all $i \in [N]$, $\mathbb{E}\|\nabla f_i(\boldsymbol{w}_i^{t,k}; \xi_i^{t,k}) - \nabla f_i(\boldsymbol{w}_i^{t,k})\|_2^2 \leq \sigma^2$, i.e., the stochastic gradients have bounded variance.*

Assumption 3 (bounded global gradient variance (BGGV)). *The global variance is bounded, i.e., $\|\nabla f_i(\boldsymbol{w}) - \nabla f(\boldsymbol{w})\|^2 \leq \zeta^2$.*

Assumption 4 (bounded local gradient norm (BLGN)). *At each node $i \in [N]$, the norm of the gradients is uniformly bounded, i.e., $\max_{\boldsymbol{w}} \|\nabla f_i(\boldsymbol{w})\| \leq B$.*

While convergence guarantees for FL and DFL have been studied extensively [63, 57, 22, 59], we provide stability analysis for DFedAvgM to give generalization guarantees under Assumptions 1–4. Along with related convergence guarantees, our work here elucidates the importance of beneficial graph topology design.

2.1. Convergence of DFedAvgM. We state a convergence result for DFedAvgM to highlight the effect of graph topology on convergence rates in DFL. This result analyzes the convergence of the sequence $\{\bar{\boldsymbol{w}}^t\}_{t=1}^T$ over the T communication rounds, where $\bar{\boldsymbol{w}}^t := \frac{1}{N}\sum_{i=1}^N \boldsymbol{w}_i^t$ is the averaged weight vector over all the nodes. The following result comes from [59] and highlights how the spectral properties of the overlay network affect the balance between local updates at each client and the communication effects of the graph topology for the convergence of the global weight vector $\bar{\boldsymbol{w}}^t$.

THEOREM 2.2 (general nonconvexity [59]). *Let the sequence $\{\bar{\boldsymbol{w}}_i^t\}_{t\geq 0}$ be generated by the DFedAvgM for each $i = 1, 2, \ldots, N$, and suppose Assumptions 1–4 hold. Moreover, assume the constant stepsize η satisfies $0 < \eta \leq 1/8LK$ and $64L^2K^2\eta^2 + 64LK\eta < 1$, where L is the Lipschitz constant from Assumption 1 and K is the number of local updates before communication. Then,*

$$\min_{1\leq t\leq T} \mathbb{E}\|\nabla f(\bar{\boldsymbol{w}}^t)\|^2 \leq \frac{2(f(\bar{\boldsymbol{w}}^1) - \min f)}{\gamma(K,\eta)T} + \alpha(K,\eta) + \frac{\Xi(K,\eta)}{(1-\lambda)^2}, \tag{2.2}$$

where T is the total number of communication rounds and $\gamma(K,\eta)$, $\alpha(K,\eta)$, and $\Xi(K,\eta)$ are constants; the detailed forms are given in the appendix.[1]

In this result from [59], we can clearly see that the convergence of the auxiliary sequence $\bar{\boldsymbol{w}}^t$ depends on the third term, which is determined by the value of the mixing parameter $\lambda \in (0,1)$; namely, the closer that λ is to 1, the worse the convergence bound of the final term of (2.2) becomes. In [68], a similar dependence on this graph-dependent value λ appears in their convergence result for a slightly different version of DFL with momentum. All this motivates selecting a graph topology that will minimize λ.

The three terms involved in the bound of (2.2) collectively capture the balancing between local updates at each client and the mixing of information between clients through the graph topology. Namely, the final term that is dependent on λ represents how the graph topology helps to propagate the information learned by each client.

While the first term explicitly depends on the total number of communication rounds T, the other two terms do not explicitly have dependence on the number of communication rounds. One can choose stepsize η that is dependent on T so that each term can be seen to diminish as we choose greater T. As shown in [59], the choice $\eta = 1/LK\sqrt{T}$ represents such a choice of stepsize, and we can then write

$$\begin{aligned}\min_{1\leq t\leq T} \mathbb{E}\|\nabla f(\bar{\boldsymbol{w}}^t)\|^2 \leq \mathcal{O}\Bigg(&\frac{(1-\beta)(f(\bar{\boldsymbol{w}}^1) - \min f)}{\sqrt{T}} \\ &+ \frac{(1-\beta)(\sigma^2 + K\zeta^2) + \frac{\beta^2}{(1-\beta)}K(\sigma^2 + B^2)}{K\sqrt{T}} \\ &+ \frac{(1-\beta)(\sigma^2 + K\zeta^2 + KB^2) + \frac{\beta^2}{(1-\beta)}K(\sigma^2 + B^2)}{(1-\lambda)^2 KT^{3/2}}\Bigg).\end{aligned}$$

With this special case, we can see how each term will diminish as the number of global communication rounds (T) and the number of local updates (K) increase. There still

[1] As of the writing of this paper, the current draft of [59] states a scaling of $1/(1-\lambda)$ on the last term of (2.2). This is a typo and will be corrected soon; we state the correct scaling here.

TABLE 1
Notation used in results of Theorems 2.2 *and* 2.5.

Symbol	Definition
N	total no. of nodes in graph
n	no. of datapoints available at each node
T	no. of *global* communication rounds
K	no. of *local* updates
η	stepsize of local updates
L	L-smoothness constant for f (Assumption 1)
σ	BLGV constant (Assumption 2)
ζ	BGGV constant (Assumption 3)
B	BLGN constant (Assumption 4)
λ	Markov chain mixing constant, $\max\{\lvert\lambda_2(\boldsymbol{M})\rvert, \lvert\lambda_N(\boldsymbol{M})\rvert\}$
ϵ	uniform stability bound in Lemma 2.4

is a strong dependence of the third term on the mixing parameter λ, which further motivates our choice of graph topology in section 3.

2.2. Generalization of DFedAvgM. In this section, we will establish a generalization bound of DFedAvgM. Given an algorithm $\mathcal{A}$ that acts on data $\mathcal{D}$ with output $\mathcal{A}(\mathcal{D})$, the generalization error is given by $\epsilon_{gen} := \mathbb{E}_{\mathcal{D},\mathcal{A}}[F(\mathcal{A}(\mathcal{D})) - F_{\mathcal{D}}(\mathcal{A}(\mathcal{D}))]$, where $F(\boldsymbol{x}) = \mathbb{E}_{\xi\sim\mathcal{D}} f(\boldsymbol{x};\xi)$ is the *true risk* and $F_{\mathcal{D}}(\boldsymbol{x}) = \sum_{i=1}^{n} f(\boldsymbol{x};\xi)/n$ is the *empirical risk* of the machine learning model with model parameter $\boldsymbol{x}$ and loss function f. Uniform stability is a useful property used to bound the generalization error ϵ_{gen}; see, e.g., [19, 13].

DEFINITION 2.3. *A randomized algorithm $\mathcal{A}$ is ϵ-uniformly stable if for any two data sets $\mathcal{D}, \mathcal{D}'$ with N samples, each differing in one example, we have*

$$\sup_{\xi} \mathbb{E}_{\mathcal{A}}[f(\mathcal{A}(\mathcal{D});\xi) - f(\mathcal{A}(\mathcal{D}');\xi)] \le \epsilon.$$

With this definition in hand, it has been proven that uniform stability implies bounded generalization error.

LEMMA 2.4 (see [52, 19]). *Let $\mathcal{A}$ be ϵ-uniformly stable; then it follows that*

$$\lvert\mathbb{E}_{\mathcal{D},\mathcal{A}}[F(\mathcal{A}(\mathcal{D})) - F_{\mathcal{D}}(\mathcal{A}(\mathcal{D}))]\rvert \le \epsilon.$$

Therefore, to ensure the generalization bound of a given random algorithm $\mathcal{A}$, we simply compute the uniform stability bound ϵ. To establish this result, we additionally require Assumption 4, i.e., boundedness of the local gradients.

The following theorem summarizes our result of uniform stability for DFedAvgM given the assumptions stated previously; the proof can be found in the appendix. To aid in parsing the result of Theorem 2.5, we provide a table of notation in Table 1.

THEOREM 2.5 (uniform stability). *Under Assumptions* 1–4*, we have that for any T if the stepsize $\eta_t \le \frac{c}{t}$ and c is small enough, then DFedAvgM satisfies uniform stability with*

$$\epsilon \le T^{\frac{cLK}{1+cLK}} \left(\frac{(\sup f) K (cLK)^{\frac{1}{1+cLK}}}{n} + \frac{2\sigma B}{NL(cLK)^{\frac{cLK}{1+cLK}}} \right) + \frac{B(\sigma+B)(cK+2C_\lambda)}{cLK}, \tag{2.3}$$

where $\sup f < \infty$ is the uniform bound on the size of the nonnegative global loss function f, n is the local data set size, and

$$C_\lambda := 2\lambda^2 + 4\lambda^2 \ln\frac{1}{\lambda} + 2\lambda + \frac{2}{\ln\frac{1}{\lambda}}$$

is a constant depending on the graph topology.

Per Lemma 2.4, we have that the generalization error for DFedAvgM is bounded by the same constant that bounds the uniform stability, ϵ. Again, we note here the *explicit dependence* of the generalization error on the corresponding value of λ for the mixing matrix $\boldsymbol{M}$ of the graph topology. C_λ is an increasing function of $\lambda \in (0, 1)$, which implies that the bound in (2.3) improves with smaller λ.

The result of Theorem 2.5 is an adaptation of Lemma 3.12 in [19] to the DFL setting utilizing DFedAvgM updates. The most striking difference between these results is the final, graph-dependent term on the right-hand side of the inequality (2.3) that contains the constant C_λ. Theorem 2.5 implies that good generalization guarantees—as captured by uniform stability—in DFL not only require advantageous properties for the function f (as does [19]), but also are *significantly* dependent on the graph topology. That is, with all other assumptions about f held constant, a graph topology with a large value of C_λ can have significantly poor uniform stability guarantees. This motivates careful choice of graph topology for DFL applications.

3. Network topology design. The results of section 2 show that the network topology has a profound impact on both optimization and generalization of DFedAvgM. According to Theorems 2.2 and 2.5, the closer λ is to 1, the more slowly DFedAvgM converges (Theorem 2.2) and the worse it generalizes (Theorem 2.5). To improve DFedAvgM, we propose a theoretically efficient and practical *sparse network topology* whose λ is far away from 1.

For the sake of notation, we recall graph definitions and properties to introduce network construction. Given an undirected, connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we define the *graph Laplacian* $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{A} = [a_{i,j}]$ (with $a_{i,j} = 1$ if $(i, j) \in \mathcal{E}$) is the adjacency matrix and $\boldsymbol{D}_{i,i} = \sum_{j=1}^{N} a_{i,j}$ is the diagonal degree matrix of $\mathcal{G}$. Since $\mathcal{G}$ is undirected, we have that both $\boldsymbol{A}$ and $\boldsymbol{L}$ are symmetric. Note that $\boldsymbol{L}$ is positive semidefinite, with a trivial eigenvalue of 0 occurring with multiplicity equal to the number of connected components in $\mathcal{G}$. As we assume that $\mathcal{G}$ is connected, this means that only the first eigenvalue $\lambda_1(\boldsymbol{L}) = 0$, and we can order the rest of the eigenvalues as $0 = \lambda_1(\boldsymbol{L}) < \lambda_2(\boldsymbol{L}) \leq \lambda_3(\boldsymbol{L}) \leq \cdots \leq \lambda_N(\boldsymbol{L})$. Define the *reduced condition number* of $\boldsymbol{L}$ as

$$\kappa(\boldsymbol{L}) := \frac{\lambda_N(\boldsymbol{L})}{\lambda_2(\boldsymbol{L})}, \tag{3.1}$$

which is a measure of graph connectivity because a *smaller* $\kappa(\boldsymbol{L})$ corresponds to a graph with *higher* connectivity. We note that graph connectivity is usually characterized by the size of $\lambda_2(\boldsymbol{L})$ alone [12, 17], but this reduced condition number also provides a relative view on the connectivity properties as it is inversely proportional to $\lambda_2(\boldsymbol{L})$. Our focus on $\kappa(\boldsymbol{L})$ then is mainly motivated by our choice of mixing matrix that we define here shortly, wherein this reduced condition number explicitly appears.

To illustrate the relationship between graph connectivity and $\kappa(\boldsymbol{L})$, we briefly mention the values of $\kappa(\boldsymbol{L})$ for two extremes of connectivity—complete graphs and Ring graphs. Consider the value of $\kappa(\boldsymbol{L}_{com})$ for a *complete* graph with corresponding graph Laplacian matrix $\boldsymbol{L}_{com}$ wherein each node is connected to every other node in the graph. One can straightforwardly show that $\lambda_2(\boldsymbol{L}_{com}) = \lambda_N(\boldsymbol{L}_{com}) = N$ so that $\kappa(\boldsymbol{L}_{com}) = 1$, the smallest possible value for the reduced condition number. A complete graph represents one extreme of connectivity, being as connected as possible for a simple graph and giving the most optimal mixing properties for an associated Markov chain. At the other extreme—namely very low connectivity—the Ring graph

on N nodes with associated graph Laplacian matrix $\boldsymbol{L}_{ring}$ yields a reduced condition number of $\kappa(\boldsymbol{L}_{ring}) \geq N^2/\pi^2 \gg 1$ (see section 3.1). This is an important constant that allows us to quantify how useful a given graph topology is for the purposes of improving convergence and generalization of DFedAvgM.

We restrict our attention to graphs that are undirected (i.e., with symmetric adjacency matrix $\boldsymbol{A}$) to ensure the desired symmetric property of the associated mixing matrix $\boldsymbol{M}$. An interesting direction for further study would be to consider directed graphs, where the adjacency matrix is no longer symmetric as edges are not necessarily reciprocated in the graph. As a result, the eigenvalues of the associated graph Laplacian matrix $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ are no longer guaranteed to be real-valued. In this case, we lose much of the spectral graph theoretic guarantees that form the basis of our analysis. For example, expander graphs by definition are undirected [35, 21]. We do note that one could consider a symmetrization of the graph Laplacian matrix for a directed graph [11], but we leave that as a direction for future work.

We focus on the following doubly stochastic mixing matrix that was used in [10],

$$\boldsymbol{M} = \boldsymbol{I} - \frac{2}{(1+\theta)\lambda_N(\boldsymbol{L})}\boldsymbol{L}, \quad \theta \in [0, 1).$$

This simple choice of mixing matrix allows us to straightforwardly quantify the effect of the reduced condition number $\kappa(L)$ on the mixing properties of associated Markov chain. Other choices of mixing matrices are possible [8, 70], but we restrict our analysis to this choice. The eigenvalues of this mixing matrix $\boldsymbol{M}$ have a straightforward relationship with eigenvalues of $\boldsymbol{L}$:

$$\lambda_i(\boldsymbol{M}) = 1 - \frac{2}{(1+\theta)\lambda_N(\boldsymbol{L})}\lambda_i(\boldsymbol{L}).$$

Then it is clear that

$$\begin{aligned}
\lambda &= \max\{|\lambda_2(\boldsymbol{M})|, |\lambda_N(\boldsymbol{M})|\} \\
&= \max\left\{\left|1 - \frac{2}{(1+\theta)\lambda_N(\boldsymbol{L})}\lambda_2(\boldsymbol{L})\right|, \left|1 - \frac{2}{(1+\theta)\lambda_N(\boldsymbol{L})}\lambda_N(\boldsymbol{L})\right|\right\} \\
&= \max\left\{\frac{\left|1+\theta-\frac{2}{\kappa(\boldsymbol{L})}\right|}{1+\theta}, \frac{1-\theta}{1+\theta}\right\}.
\end{aligned}$$

For a fixed $\kappa(\boldsymbol{L})$, we can view the Markov chain mixing constant λ as a function of θ that can be optimized to lead to the lowest value of λ. In Figure 2, we have plotted the function $|\lambda_N(\boldsymbol{M})| = (1-\theta)/(1+\theta)$ along with $|\lambda_2(\boldsymbol{M})| = |1+\theta-2/\kappa(\boldsymbol{L})|/(1+\theta)$ for two values of $\kappa(\boldsymbol{L})$. For each fixed $\kappa(\boldsymbol{L})$, the corresponding lowest value of $\lambda(\theta)$ occurs when $|\lambda_2(\boldsymbol{M})| = |\lambda_N(\boldsymbol{M})|$, shown in Figure 2 as stars, that is, when $\theta = \theta^*(\kappa(\boldsymbol{L}))$. It is clear that

$$\begin{aligned}
&|\lambda_2(\boldsymbol{M})| = |\lambda_N(\boldsymbol{M})| \\
\iff &\frac{1+\theta^*(\kappa(\boldsymbol{L})) - \frac{2}{\kappa(\boldsymbol{L})}}{1+\theta^*(\kappa(\boldsymbol{L}))} = \frac{1-\theta^*(\kappa(\boldsymbol{L}))}{1+\theta^*(\kappa(\boldsymbol{L}))} \\
\iff &\theta^*(\kappa(\boldsymbol{L})) - \frac{2}{\kappa(\boldsymbol{L})} = -\theta^*(\kappa(\boldsymbol{L})) \\
\iff &\theta^*(\kappa(\boldsymbol{L})) = \frac{1}{\kappa(\boldsymbol{L})},
\end{aligned}$$

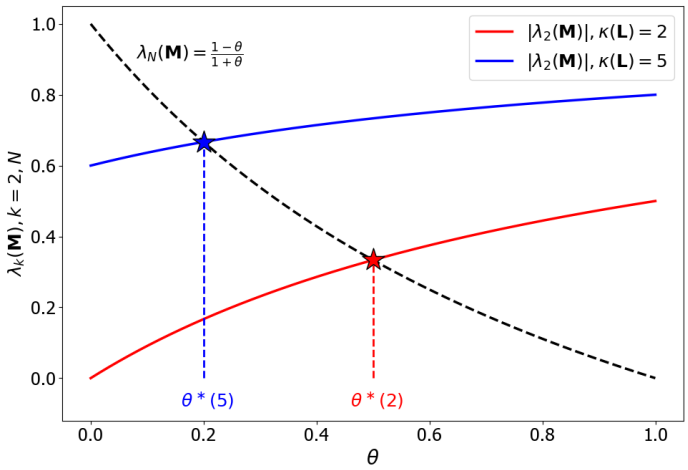


FIG. 2. *Plot of λ for the expander graph as a function of θ. The stars indicate the optimal choice of $\theta = \theta^*(\kappa)$, given the value of $\kappa(\boldsymbol{L})$. Lower κ leads to a higher value of $\theta^*(\kappa)$, which in turn leads to a lower value of λ, which is desired.*

which indeed gives that $\theta^*(\kappa(\boldsymbol{L})) \in (0, 1)$ since one can straightforwardly show that $\kappa(\boldsymbol{L}) > 1$ (unless $\mathcal{G}$ is the complete graph) due to standard bounds on $\lambda_2(\boldsymbol{L})$ and $\lambda_N(\boldsymbol{L})$ by Fiedler [17].

It is straightforward then that choosing a graph structure with a smaller value of $\kappa(\boldsymbol{L})$ gives $\mathcal{G}$ better connectivity properties. Somewhat in competition with this connectivity is the communication cost of a given graph topology; that is, better connectivity of a graph structure generally corresponds to more edges in the graph, which increases the communication cost. Each node sends its updated model to each of its neighbors, and so an increased number of edges results in more communication that must happen between nodes.

We propose using d-regular expander graphs to balance this connectivity communication tradeoff. Expander graphs are an important topic of study in the intersection of theoretical computer science and spectral graph theory with various applications such as the design of communication networks and error correcting codes [21, 55, 56]. A d-regular graph has a fixed number degree d for each node; i.e., $d(i) = d$ for all i. Expander graphs are sparse graphs that have strong connectivity properties, of which d-regular expander graphs (and the special case of *Ramanujan graphs*) are in a sense "optimal" graph connectivity structures (possessing a small constant $\kappa(\boldsymbol{L})$) with fixed communication cost. While Ramanujan graphs are not known for every value of total nodes N and degree d, with high probability most d-regular graphs are approximately Ramanujan for large enough N [10]. We briefly note here that much has been done to study the spectral properties of and the explicit construction of d-regular expander graphs [2, 3, 4, 5]. We refer the reader to [35, 21] for more information about the rich history of the analysis and application of expander graphs.

For d-regular graphs, there exists a convenient upper bound for $\kappa(\boldsymbol{L})$ per the results of [12, 43]:

$$\kappa(\boldsymbol{L}) \leq \frac{d + \lambda_2(\boldsymbol{A})}{d - \lambda_2(\boldsymbol{A})},$$

where $\lambda_2(\boldsymbol{A})$ is the first nontrivial eigenvalue of the corresponding adjacency matrix $\boldsymbol{A}$. Note that the eigenvalue $\lambda_1(\boldsymbol{A}) = d$ is trivial and corresponds to the eigenvalue

$\lambda_1(\boldsymbol{L}) = 0$ of $\boldsymbol{L}$. If the d-regular graph in question is Ramanujan [40, 41], then according to well-known results [41, 12, 43] we can bound the reduced condition number of the corresponding graph Laplacian $\boldsymbol{L}_R$ as follows:

$$\kappa(\boldsymbol{L}_R) \leq \frac{d + 2\sqrt{d-1}}{d - 2\sqrt{d-1}}. \tag{3.2}$$

As the right-hand side of (3.2) is a *decreasing* function of d, this would suggest choosing larger d in order to minimize $\kappa(\boldsymbol{L}_R)$. However, increasing d will incur greater communication costs. One can in practice choose the value of d according to a prescribed bound on the total communication cost.

3.1. Comparison to Ring and Erdös–Rényi graphs. We show that other graph topologies are in a sense suboptimal for the purposes of DFedAvgM, highlighting two common examples: Ring and Erdös–Rényi graphs. We emphasize that using *d-regular Ramanujan graphs is in a sense "optimal"* because they possess *strong connectivity* properties in the graph topology while requiring *low communication cost* for local node neighborhood communication (i.e., sparsity).

Ring graphs—poor connectivity. A Ring graph is an extremely sparse, but still connected, 2-regular graph, where the graph structure constitutes a ring (see Figure 1(b)). While a very simple and sparse topology to impose on the nodes of the graph, Ring graphs possess poor connectivity properties that we can directly compare with d-regular Ramanujan graphs.

It is well known that the eigenvalues of the graph Laplacian $\boldsymbol{L}_{ring}$ of the Ring graph on N nodes are given by

$$\{\lambda_k(\boldsymbol{L}_{ring})\}_{k=1}^N = \left\{2 - 2\cos\left(\frac{2\pi\lfloor k/2\rfloor}{N}\right)\right\}_{k=1}^N,$$

where we note that each eigenvalue has geometric multiplicity 2, except for the first eigenvalue $\lambda_1(\boldsymbol{L}_{ring}) = 0$ and possibly the last eigenvalue $\lambda_N(\boldsymbol{L}_{ring})$, which have geometric multiplicity 1. The last eigenvalue has geometric multiplicity 1 in the case that N is even. Therefore, we can straightforwardly see that the reduced condition number for a Ring graph on N nodes is

$$\begin{aligned}\kappa(\boldsymbol{L}_{ring}) &= \frac{\lambda_N(\boldsymbol{L}_{ring})}{\lambda_2(\boldsymbol{L}_{ring})} = \frac{2 - 2\cos\left(\frac{2\pi N}{2N}\right)}{2 - 2\cos\left(\frac{2\pi}{N}\right)} \\ &= \frac{4}{2 - 2\cos\left(\frac{2\pi}{N}\right)} \geq \frac{4}{2 - 2\left(1 - \frac{1}{2}\left(\frac{2\pi}{N}\right)^2\right)} = \frac{4N^2}{4\pi^2} = \frac{N^2}{\pi^2}.\end{aligned}$$

Therefore, we see that with this *lower bound*, the Ring graph has a $\kappa(\boldsymbol{L}_{ring})$ that grows *quadratically* with the size N of the graph! The corresponding value for λ approaches 1 for increasing values of N, which implies slower convergence rates (Theorem 2.2) and worse generalization bounds (Theorem 2.5). It is clear then that

$$\kappa(\boldsymbol{L}_R) \leq \frac{d + 2\sqrt{d-1}}{d - 2\sqrt{d-1}} \ll \frac{N^2}{\pi^2} \leq \kappa(\boldsymbol{L}_{ring}),$$

which shows superior convergence properties of Ramanujan expander graphs compared to the sparse Ring graph structure.

Erdös–Rényi graph—high communication cost. Another type of graph topology one could impose for DFedAvgM is an Erdös–Rényi (E-R) random graph structure, wherein each edge $(i, j) \in \mathcal{V} \times \mathcal{V}$ is sampled IID with probability $p \in (0, 1)$. It is well known that as long as $p = \Omega(\ln N/N)$, the resulting graph $\mathcal{G}$ is connected with high probability [14].

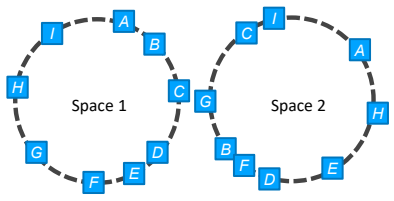
While the connectivity properties of E-R graphs are nearly guaranteed to be better than d-regular Ramanujan graphs (with d relatively small), *the communication cost of E-R graphs is prohibitively large* for large network size N. To see this, the expected degree d_i of a node $i \in \mathcal{V}$ in an E-R graph with large enough edge probability p is simply $\bar{d} = Np = \Omega(\ln N)$, which grows with the size of the graph N. This incurs a much larger communication cost than the constant cost of d-regular expander graphs as it is assumed that $d \ll N$, with $d < \ln N$ as N is large.

In section 5, we empirically verify the superior *connectivity-communication cost* balance exemplified by the d-regular expander graph structure compared to Ring and E-R graphs for DFedAvgM. These d-regular expander graphs have better connectivity properties than Ring graphs while at the same time being sparser (i.e., lower communication costs) than E-R graphs.

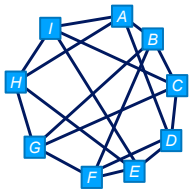
4. Practical network design. In this section, we discuss how to convert a given graph to a practical overlay network topology for DFL. We illustrate our proposed d-regular network topology in Figure 3. For a d-regular graph, suppose d is even, and let $L = d/2$. We assign to each node a set of *virtual coordinates* represented by an L-dimensional vector $\langle x_1, x_2, \ldots, x_L \rangle$, where each element x_i is a randomly generated real number $0 \leq x_i < 1$, as shown in Figure 3(a). There are L virtual ring spaces such as the two shown in Figure 3(b). In the ith space, a node is *virtually* placed on a ring based on the value of its ith coordinate x_i. Coordinates in each space are circular, and 0 and 1 are superposed. In the ith space, a node connects to the two adjacent nodes that have closest values according to the coordinate x_i; for example, B connects to A and C in space 1 and G and F in space 2 in Figure 3. Hence each node has at most $d = 2L$ neighbors. Neighboring nodes may happen to be adjacent in multiple spaces, such as A and D. In such a case, A can connect to another node in the same situation, such as E. In the end, the equivalent network topology is shown in Figure 3(c).

Node ID	Coor. 1	Coor. 2
A	0.05	0.17
B	0.13	0.62
C	0.23	0.91
D	0.36	0.53
E	0.42	0.42
F	0.51	0.58
G	0.63	0.73
H	0.78	0.26
I	0.91	0.97

(a) Coordinates



(b) Virtual spaces



(c) Actual topology

FIG. 3. *DFL network topology in working systems. Each node generates a set of coordinates, and the network is generated in a distributed manner by allowing each node to execute the proposed protocols locally.*

The proposed network is a close proximal construction for a random d-regular network [69]. Note that in practice there does not exist a perfect construction of a random d-regular graph [54], and there is no way for a network node to verify whether the entire network is Ramanujan only based on its local information.

The construction of a correct topology can be achieved by allowing each node to maintain the two closest nodes on each virtual ring. When a new node joins the network, it can always succeed to find the two closest nodes on each virtual ring by recursive queries [25].

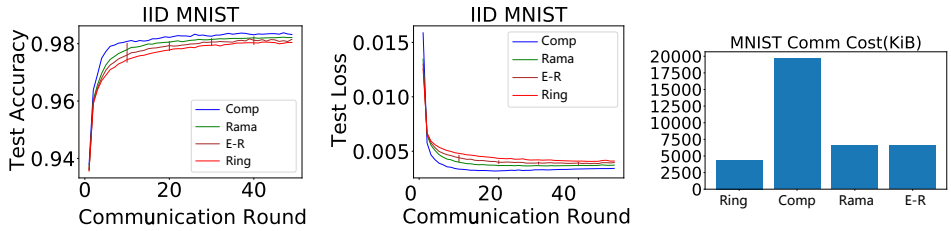
4.1. Network recovery from node failures. To maintain a correct DFL topology for a dynamic set of nodes, protocols should be designed to recover errors from node failures and leaves. Here an error is defined as a node that has a wrong neighbor set compared to a correct DFL network topology. If a node x fails from the network, in each virtual space i, its adjacent nodes y_i and z_i should remove x from their neighbors and add each other as a new neighbor. To recover from such single-node failure, the proposed recovery protocol allows each node to store the IP addresses of the two-hop neighbors. Hence if a node is detected to fail, its two adjacent nodes can directly connect as new neighbors.

5. Experimental results.

5.1. Convergence and generalization. We evaluate the training loss, test loss, test accuracy, and the communication cost versus the communication round for Ring, Erdös–Rényi (E-R), fully connected (complete), and the proposed expander graphs. Namely, we use $d = 3$ regular expander graphs (called Ramanujan). The communication cost is estimated by the size of the parameters of the models. In all experimental settings, the topology is generated by a central server before the training starts and is stored in each client, but the central host is not involved in the actual training process. The expander graph is generated by adding extra edges on top of the Ring graph in virtual spaces according to 4. The Erdös–Rényi graph is generated by selecting random edges from all possible edges with the probability $p = \frac{\ln N}{N}$, where N is the total number of clients. The expander graphs result in faster convergence and better generalization for DFedAvgM in training different models on different datasets. To conduct more comprehensive experiments and testing, both the real network settings and the simulation are used in our evaluation. To exclude other factors, no tuned optimization or data compression algorithms are used in the experiments. In the evaluation, fully connected graphs are shown as a baseline but are hardly practical in real-world applications considering the communication cost and availability of such a topology. The Ring topology is easy to implement and widely used in previous works, so it is also shown as a baseline. Due to the randomness of Erdös–Rényi graph construction, the experimental results are inconsistent when there are relatively few nodes in the graph. Consequently, we do not include the Erdös–Rényi graph in every MNIST experiment below.

MNIST IID. We randomly split the MNIST dataset without any biases into 10 different subsets (i.e., the number of clients is $N = 10$). Each client owns a local multilayer perceptron (MLP) model with one hidden layer of size 200. Each client has access to only one local subset as its training set. We train the local model with a batch size of 20 and use the cross-entropy loss function. We use SGD with a learning rate of $\eta = 0.01$ and momentum $\beta = 0.9$. After $K = 3$ epochs of local training, each node communicates with its topological neighbors and averages all the parameters of the MLP model. After each communication round, the test accuracy and test loss of each client are recorded and averaged in Figure 4. Based on our experiments in this IID setting, the fully connected and expander graphs converge at round 16, which is a significant improvement over the 26 rounds that the Ring graph requires to converge. According to the test accuracy shown in Figure 4, the fully connected graph has the

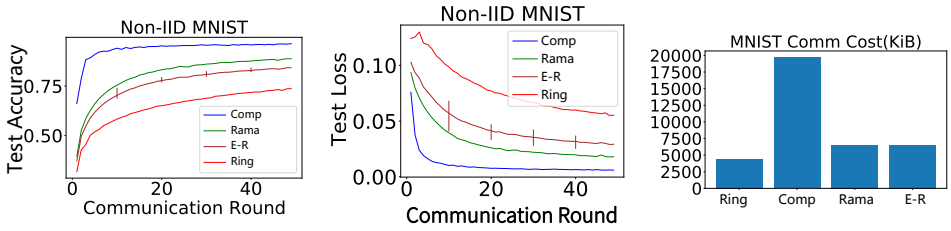
best test accuracy of 98.2%, while the expander graph reaches a similar 98.0% with only one third of the communication cost. The Ring graph reaches 97.7% accuracy due to the ideal distribution of the data.



Test Accuracy Test Loss Comm. Cost

FIG. 4. *The test accuracy, test loss, and communication cost of the Ring/E-R/3-regular expander (Ramanujan)/fully connected graphs on IID MNIST. All of the graphs enable DFL to reach over* 92% *accuracy, but DFL using the expander graph converges faster than DFL using the E-R or Ring graphs.*

MNIST non-IID. In this experiment, all settings are similar to the IID experiment except each node owns a local dataset consisting of *only one label* (one digit in MNIST). We note that this distribution is extremely unfavorable for generalization. The test dataset is sampled in a balanced form from the original dataset as the IID settings. As shown in Figure 5, the expander graph reaches 88.8% accuracy, which is much higher than the Ring graph (73.68%). As is expected, the fully connected graph reaches the best accuracy of 94%. Again, while the expander graph's final accuracy is lower than the fully connected graph's, it only incurs 33% of the communication cost. After each communication round, the testing accuracy and loss of each client are recorded and averaged and shown in Figure 5. In summary, the expander graph achieves both faster convergence as well as better generalization than the Ring graph, and the expander graph's performance is comparable to that of the fully connected graph, but with a more manageable communication cost. In addition, while the E-R graph can occasionally achieve accuracy comparable to that attained by the expander graph, it has an unstable performance in practical cases due to the random nature of its construction.

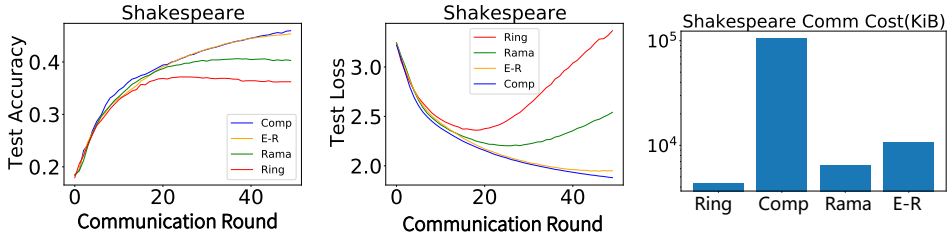


Test Accuracy Test Loss Comm. Cost

FIG. 5. *The test accuracy, test loss, and communication cost of the Ring/E-R/3-regular Ramanujan/fully connected graphs on non-IID MNIST. The expander graph reaches* 88.79% *accuracy, which is higher than the Ring graph's* 73.68%. *The E-R graph achieves a maximum accuracy of* 86.33% *(which is comparable to the expander graph's accuracy), but the minimum accuracy for the E-R graph is* 79.42%. *The communication cost of the expander graph is only one third of the fully connected graph's.*

Language modeling. We further conduct an experiment to evaluate the effect of different graph topologies for DFL applied to language modeling on the Shakespeare dataset [38]. First, we split the original dataset into 100 subsets. Each subset contains

only a small fraction of the roles' dialogue. Because of the unique characteristics and language of each role, the dataset can be considered as non-IID. We then create 100 long short-term memory (LSTM) models that each have 256 hidden units and 2 layers. We use the cross-entropy loss function and train each LSTM with the corresponding local non-IID dataset with the learning rate $\eta = 0.5$ and momentum $\beta = 0.9$. Similar to the MNIST experiments, after 3 epochs of local training, each node communicates with its neighbors, through which is averaged all the parameters of the LSTM model. After each communication round, the test loss and accuracy of each client are recorded and averaged in Figure 6. The Erdös–Rényi graph achieves an accuracy of 45.3%, which is close to the fully connected graph's 45.8%. The expander graph reaches an accuracy of 40.4%, and the Ring graph only reaches 36.2%. In this unfavorable data distribution, the Ring graph generalizes significantly worse than the expander graph. We note that the Erdös–Rényi graph has better test accuracy and loss than the expander graph because it results in significantly higher degrees to ensure connectivity of the graph. Thus, the Erdös–Rényi graph has a higher communication cost (Figure 6). Also, similar to previous experiments, DFedAvgM with an expander graph topology converges faster than the Ring graph. In this case, we can see that the communication cost for the complete graph is roughly 16 times higher than that of the expander graph. With some moderate communication costs, the expander graph seems to generalize better than the Ring graph topology while also having similar convergence to the fully connected graph.



Test Accuracy Test Loss Comm. Cost

FIG. 6. *The test accuracy, test loss, and communication cost of the Ring/3-regular Ramanujan/Erdös–Rényi/complete graph on non-IID Shakespeare dataset. The Erdös–Rényi graph achieves an accuracy rate of* 45.3%. *The expander graph reaches an accuracy rate of* 40.4%, *while the Ring graph only reaches* 36.2%.

Contrasting computational time of DFL with different graphs. We further validate the advantage of Ramanujan graphs for DFL in terms of compute time efficiency. In particular, we consider training the aforementioned DFL models using different graphs for IID and non-IID MNIST classification. Figure 7 shows the computational time—including initialization of the DFL system, model aggregation, and model training—of DFL using different graphs for communication to reach a given test accuracy. We see that DFL using 3-regular Ramanujan graphs remarkably improves computational time over the other graph topologies (complete, Erdös–Rényi, and Ring graphs) to reach 75% and 98% testing accuracies for non-IID and IID MNIST classification, respectively (Table 2).

5.2. Robustness to client failures. To test the robustness to client failures of DFedAvgM with different network topologies, we drop 10% and 20% of clients during the communication round and compare the performance of Ring, expander, Erdös–Rényi, and fully connected graphs. In the language modeling task with the Shakespeare dataset, we mask the input of the dropped nodes to simulate the com-

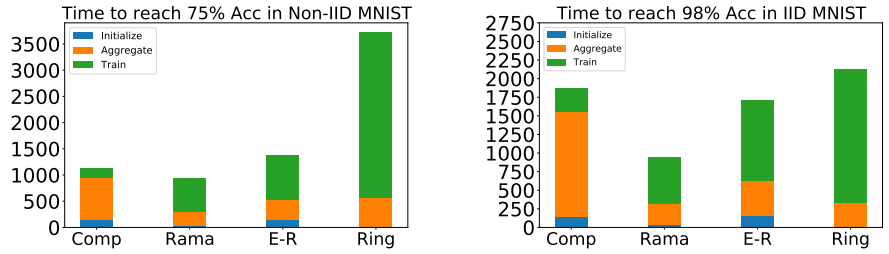
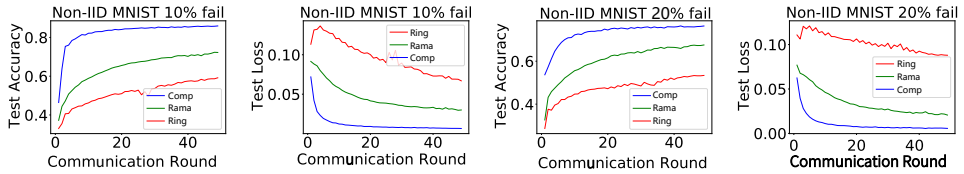


FIG. 7. *The computational time—including initialization of the DFL system, model aggregation, and model training—of DFL uses different graphs for communication to reach a given test accuracy. Left: Training DFL for non-IID MNIST classification with 75% test accuracy. Right: Training DFL for IID MNIST classification with 98% test accuracy. Here we assume the computational resource is close to the end device and has a limited 2MB/s bandwidth. (Unit: seconds)*

TABLE 2
The runtime of DFL using different graphs to reach a target test accuracy for both IID and non-IID MNIST classification. (Unit: seconds.)

Dataset	Non-IID MNIST			IID MNIST		
Accuracy	75%	70%	65%	98%	97.5%	97%
Complete	1130	882	882	1865	1130	876
Ramanujan	939	817	681	943	553	420
Erdös–Rényi	1452	1125	998	1716	932	735
Ring	3717	2763	2021	2127	802	594



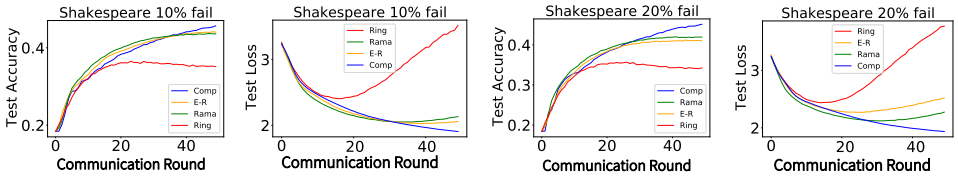
Acc 10% failure Loss 10% failure Acc 20% failure Loss 20% failure

FIG. 8. *The test accuracy and test loss of the Ring/3-regular Ramanujan/complete graphs on non-IID MNIST with client failures.*

munication failure. All the dropped nodes are randomly selected and excluded from the final results.

MNIST non-IID. As shown in Figure 8, communication failure not only causes the loss of corresponding training samples globally but also breaks the connection of the topology. With the weakest connectivity, the Ring graph degrades to 51.3% accuracy when 20% of the nodes are dropped. The clients are partitioned when multiple nodes fail in a Ring graph. On the other hand, the expander graph reaches an accuracy of 65.3% due to its high connectivity and lack of a partition.

Language modeling. In Figure 9, we have a similar situation as Figure 8. With the weakest connectivity, the Ring graph degrades to 33.7% accuracy when 20% of the nodes are dropped. Similar to the MNIST non-IID communication failure experiment, the clients are partitioned when multiple nodes fail in a Ring graph. The expander graph, however, reaches an accuracy of 41.5%. Additionally, although the Erdös–Rényi graph performs slightly better than the expander graph with 10% client failures, it performs worse than the expander graph with a 20% client failure because of its weaker connectivity property.



Acc 10% failure Loss 10% failure Acc 20% failure Loss 20% failure

FIG. 9. *The test accuracy and test loss of the Ring/3-regular Ramanujan/complete/Erdös–Rényi graphs on non-IID MNIST with client failures.*

6. Concluding remarks. In this paper, we presented the theoretical advantages of expander graph-based overlay networks and their practical construction. We numerically verified the efficacy in accelerating training, improving generalization, and enhancing robustness to client failures of decentralized federated learning by using expander graph-based overlay networks on various benchmarks. How to establish the theoretical robustness guarantees of the expander graph-based overlay networks to the node failure is an interesting future direction. Other interesting future directions include integrating the decentralized federated learning framework with decentralized stochastic algorithms other than decentralized stochastic gradient descent, e.g., Push-SAGA [50], and designing the corresponding efficient and reliable overlay networks.

Appendix A. Technical proofs. We repeat Table 1 here for the reader's convenience.

TABLE 3
Notation used in proof of Theorem A.1.

Symbol	Definition
N	total no. of nodes in graph
n	no. of datapoints available at each node
T	no. of *global* communication rounds
K	no. of *local* updates
η	stepsize of local updates
L	L-smoothness constant for f (Assumption 1)
σ	BLGV constant (Assumption 2)
ζ	BGGV constant (Assumption 3)
B	BLGN constant (Assumption 4)
λ	Markov chain mixing constant, $\max\{\lvert\lambda_2(\boldsymbol{M})\rvert, \lvert\lambda_N(\boldsymbol{M})\rvert\}$
ϵ	uniform stability bound in Lemma 2.4

THEOREM A.1 (uniform stability (Theorem 2.5 restated)). *Under Assumptions* 1–4, *we have that for any* T *if the stepsize* $\eta_t \leq \frac{c}{t}$ *and* c *is small enough, then DFedAvg satisfies uniform stability with*

$$\epsilon_{stab} \leq T^{\frac{cLK}{1+cLK}} \left(\frac{(\sup f) K (cLK)^{\frac{1}{1+cLK}}}{n} + \frac{\frac{2\sigma B}{NL}}{(cLK)^{\frac{cLK}{1+cLK}}} \right) + \frac{B(\sigma+B)(cK+2C_\lambda)}{cLK},$$

where $\sup f < \infty$ *is the uniform bound on the size of the nonnegative loss function.*

Proof. Assume that each node i has access to local datasets $\mathcal{D}_i = \{(\boldsymbol{x}_i^\ell, y_i^\ell)\}_{\ell=1}^{n_i}$ of size $n_i = n$, and let $\mathcal{D} = \cup_{i=1}^N \mathcal{D}_i$ be the set of Nn datapoints over the whole graph. Assume then that the datasets $\mathcal{D}, \tilde{\mathcal{D}}$ differ by only one point; that is, there exists exactly one $i^* \in \{1, \ldots, N\}$ such that $\mathcal{D}_i$ and $\tilde{\mathcal{D}}_i$ differ in exactly one point. Define the random variables

$$\xi_i^{t,k} \sim Unif(\mathcal{D}_i),$$

where $\{\xi_i^{t,k}\}_{k=1}^K$ are sampled IID (with replacement). We denote the collection of random variables sampled from $\mathcal{D}$ at all N nodes in the graph as $\Xi^{(t,k)} := \{\xi_i^{t,k}\}_{i=1}^N$. Likewise, define $\tilde{\Xi}^{(t,k)} = \{\xi_i^{t,k}\}_{i=1}^N$ to be the collection of samples from $\tilde{\mathcal{D}}$ at all N nodes in the graph.

Now define $\bar{\boldsymbol{w}}^t, \bar{\boldsymbol{v}}^t$ to be the averages generated by DFedAvgM with training data $\mathcal{D}, \tilde{\mathcal{D}}$, respectively; that is,

$$\bar{\boldsymbol{w}}^t = \frac{1}{N}\sum_{i=1}^N \boldsymbol{w}_i^{t,0}, \quad \bar{\boldsymbol{v}}^t = \frac{1}{N}\sum_{i=1}^N \boldsymbol{v}_i^{t,0}.$$

Further, define the matrices

$$\boldsymbol{X}^{(t,k)} := [\boldsymbol{w}_1^{t,k}\ \boldsymbol{w}_2^{t,k}\ \ldots\ \boldsymbol{w}_N^{t,k}], \quad \boldsymbol{Y}^{(t,k)} := [\boldsymbol{v}_1^{t,k}\ \boldsymbol{v}_2^{t,k}\ \ldots\ \boldsymbol{v}_N^{t,k}]$$

and the gradient matrices

$$\boldsymbol{G}^{(t,k)}\left(\boldsymbol{X}^{(t,k)}; \Xi^{(t,k)}\right) := [\nabla f_1(\boldsymbol{w}_1^{t,k}; \xi_1^{t,k})\ \nabla f_2(\boldsymbol{w}_2^{t,k}; \xi_2^{t,k})\ \ldots\ \nabla f_N(\boldsymbol{w}_N^{t,k}; \xi_N^{t,k})],$$
$$\boldsymbol{G}^{(t,k)}\left(\boldsymbol{Y}^{(t,k)}; \tilde{\Xi}^{(t,k)}\right) := [\nabla f_1(\boldsymbol{v}_1^{t,k}; \tilde{\xi}_1^{t,k})\ \nabla f_2(\boldsymbol{v}_2^{t,k}; \tilde{\xi}_2^{t,k})\ \ldots\ \nabla f_N(\boldsymbol{v}_N^{t,k}; \tilde{\xi}_N^{t,k})].$$

We have that by definition of the DFedAvgM iterations

$$\begin{aligned}
&\boldsymbol{w}_i^{t,k+1} - \boldsymbol{w}_i^{t,k} \\
&= -\eta_t \nabla f_i(\boldsymbol{w}_i^{t,k}; \xi_i^{t,k}) + \theta(\boldsymbol{w}_i^{t,k} - \boldsymbol{w}_i^{t,k-1}) \\
&= -\eta_t \nabla f_i(\boldsymbol{w}_i^{t,k}; \xi_i^{t,k}) + \theta\left(-\eta_t \nabla f_i(\boldsymbol{w}_i^{t,k-1}; \xi_i^{t,k-1}) + \theta(\boldsymbol{w}_i^{t,k-1} - \boldsymbol{w}_i^{t,k-2})\right) \\
&= \ldots \\
&= -\eta_t \left(\sum_{s=0}^k \theta^{k-s} \nabla f_i(\boldsymbol{w}_i^{t,s}; \xi_i^{t,s})\right)
\end{aligned}$$

and that

$$\begin{aligned}
\boldsymbol{w}_i^{t,k+1} - \boldsymbol{w}_i^{t,k} &= -\eta_t \nabla f_i(\boldsymbol{w}_i^{t,k}; \xi_i^{t,k}) + \theta(\boldsymbol{w}_i^{t,k} - \boldsymbol{w}_i^{t,k-1}) \\
\implies \boldsymbol{w}_i^{t,K} - \boldsymbol{w}_i^{t,0} &= \sum_{k=0}^{K-1} -\eta_t \nabla f_i(\boldsymbol{w}_i^{t,k}; \xi_i^{t,k}) + \theta(\boldsymbol{w}_i^{t,k} - \boldsymbol{w}_i^{t,k-1}) \\
&= \sum_{k=0}^{K-1} -\eta_t \nabla f_i(\boldsymbol{w}_i^{t,k}; \xi_i^{t,k}) + \theta(\boldsymbol{w}_i^{t,K-1} - \boldsymbol{w}_i^{t,0}) \\
\implies \boldsymbol{w}_i^{t,K} - \boldsymbol{w}_i^{t,0} &= \frac{-\eta_t}{1-\theta}\sum_{k=0}^{K-1} \nabla f_i(\boldsymbol{w}_i^{t,k}; \xi_i^{t,k}) - \frac{\theta}{1-\theta}(\boldsymbol{w}_i^{t,K} - \boldsymbol{w}_i^{t,K-1}) \\
&= \frac{-\eta_t}{1-\theta}\sum_{k=0}^{K-1} \nabla f_i(\boldsymbol{w}_i^{t,k}; \xi_i^{t,k}) - \frac{-\eta_t\theta}{1-\theta}\sum_{k=0}^{K-1}\sum_{s=0}^{k} \theta^{k-s} \nabla f_i(\boldsymbol{w}_i^{t,s}; \xi_i^{t,s}) \\
&= \frac{-\eta_t}{1-\theta}\left(\sum_{k=0}^{K-1} \nabla f_i(\boldsymbol{w}_i^{t,k}; \xi_i^{t,k}) - \theta\sum_{k=0}^{K-1} \nabla f_i(\boldsymbol{w}_i^{t,k}; \xi_i^{t,k}) \sum_{s=0}^{K-k-1} \theta^s\right) \\
&= \frac{-\eta_t}{1-\theta}\sum_{k=0}^{K-1}\left(1 - \frac{\theta - \theta^{K-k+1}}{1-\theta}\right) \nabla f_i(\boldsymbol{w}_i^{t,k}; \xi_i^{t,k})
\end{aligned}$$

$$= \frac{\eta_t}{(1-\theta)^2} \sum_{k=0}^{K-1} (1 - 2\theta + \theta^{K-k+1}) \nabla f_i(\boldsymbol{w}_i^{t,k}; \xi_i^{t,k}). \tag{A.1}$$

Then by A.1 we can write

$$\boldsymbol{X}^{(t,K)} - \boldsymbol{X}^{(t,0)} = \frac{\eta_t}{(1-\theta)^2} \sum_{k=0}^{K-1} p_k(\theta) \boldsymbol{G}^{(t,k)}(\boldsymbol{X}^{(t,k)}; \Xi^{(t,k)}),$$

$$\boldsymbol{Y}^{(t,K)} - \boldsymbol{Y}^{(t,0)} = \frac{\eta_t}{(1-\theta)^2} \sum_{k=0}^{K-1} p_k(\theta) \boldsymbol{G}^{(t,k)}(\boldsymbol{Y}^{(t,k)}; \tilde{\Xi}^{(t,k)}),$$

where we have defined $p_k(\theta) = 1 - 2\theta + \theta^{K-k+1}$.

Letting $\mathbb{1} \in \mathbb{R}^N$ denote the vector of all ones, then the mixing matrix $\boldsymbol{M}$ satisfies $\boldsymbol{M}\frac{\mathbb{1}}{N} = \frac{\mathbb{1}}{N}$. Then we have with probability $\left(\frac{n-1}{n}\right)^K$ that the random variables $\{\Xi^{(t,k)}\}_{k=1}^K = \{\tilde{\Xi}^{(t,k)}\}_{k=1}^K$ are exactly the same:

$$\begin{aligned}
&\bar{\boldsymbol{w}}^{t+1} - \bar{\boldsymbol{v}}^{t+1} \\
&= \boldsymbol{X}^{(t,K)} \boldsymbol{M} \frac{\mathbb{1}}{N} - \boldsymbol{Y}^{(t,K)} \boldsymbol{M} \frac{\mathbb{1}}{N} \\
&= \left(\boldsymbol{X}^{(t,0)} + \left(\boldsymbol{X}^{(t,K)} - \boldsymbol{X}^{(t,0)}\right)\right) \frac{\mathbb{1}}{N} - \left(\boldsymbol{Y}^{(t,0)} + \left(\boldsymbol{Y}^{(t,K)} - \boldsymbol{Y}^{(t,0)}\right)\right) \frac{\mathbb{1}}{N} \\
&= \left(\boldsymbol{X}^{(t,0)} - \boldsymbol{Y}^{(t,0)} + \frac{\eta_t}{(1-\theta)^2} \left(\sum_{k=0}^{K-1} p_k(\theta) \boldsymbol{G}^{(t,k)}(\boldsymbol{X}^{(t,k)}; \Xi^{(t,k)}) - \sum_{k=0}^{K-1} p_k(\theta) \boldsymbol{G}^{(t,k)}(\boldsymbol{Y}^{(t,k)}; \Xi^{(t,k)})\right)\right) \frac{\mathbb{1}}{N} \\
&= \left(\left(\boldsymbol{X}^{(t,0)} - \boldsymbol{Y}^{(t,0)}\right)(\boldsymbol{I} - \boldsymbol{P}) + \left(\boldsymbol{X}^{(t,0)} - \boldsymbol{Y}^{(t,0)}\right) P\right) \frac{\mathbb{1}}{N} \\
&\quad + \frac{\eta_t}{(1-\theta)^2} \left(\sum_{k=0}^{K-1} p_k(\theta) \left[\boldsymbol{G}^{(t,k)}(\boldsymbol{X}^{(t,k)}; \Xi^{(t,k)}) - \boldsymbol{G}^{(t,k)}(\bar{\boldsymbol{w}}^t \mathbb{1}^T; \Xi^{(t,k)}) + \boldsymbol{G}^{(t,k)}(\bar{\boldsymbol{w}}^t \mathbb{1}^T; \Xi^{(t,k)})\right]\right) \frac{\mathbb{1}}{N} \\
&\quad - \frac{\eta_t}{(1-\theta)^2} \left(\sum_{k=0}^{K-1} p_k(\theta) \left[\boldsymbol{G}^{(t,k)}(\boldsymbol{Y}^{(t,k)}; \Xi^{(t,k)}) - \boldsymbol{G}^{(t,k)}(\bar{\boldsymbol{v}}^t \mathbb{1}^T; \Xi^{(t,k)}) + \boldsymbol{G}^{(t,k)}(\bar{\boldsymbol{v}}^t \mathbb{1}^T; \Xi^{(t,k)})\right]\right) \frac{\mathbb{1}}{N} \\
&= \underbrace{\frac{\eta_t}{N(1-\theta)^2} \sum_{i=1}^N \sum_{k=0}^{K-1} p_k(\theta) \left[\left(\nabla f_i(\bar{\boldsymbol{w}}^t; \xi_i^{t,k}) - \nabla f_i(\boldsymbol{w}_i^{t,k}; \xi_i^{t,k})\right) - \left(\nabla f_i(\bar{\boldsymbol{v}}^t; \xi_i^{t,k}) - \nabla f_i(\boldsymbol{v}_i^{t,k}; \xi_i^{t,k})\right)\right]}_{=:A_1}
\end{aligned} \tag{A.2}$$

$$+ \frac{1}{N} \sum_{i=1}^N \left(\bar{\boldsymbol{w}}^t - \frac{\eta_t}{(1-\theta)^2} \sum_{k=0}^{K-1} p_k(\theta) \nabla f_i(\bar{\boldsymbol{w}}^t; \xi_i^{t,k})\right) - \left(\bar{\boldsymbol{v}}^t - \frac{\eta_t}{(1-\theta)^2} \sum_{k=0}^{K-1} p_k(\theta) \nabla f_i(\bar{\boldsymbol{v}}^t; \xi_i^{t,k})\right), \tag{A.3}$$

where we note that $(\boldsymbol{I} - \boldsymbol{P})\frac{\mathbb{1}}{N} = 0$. Now, we have that since $\theta \in [0, 1)$, then $|p_k(\theta)| = p_k(\theta) \leq p_{K-1}(\theta) = (1-\theta)^2$ for each $k = 0, 1, \ldots, K-1$. This means we can calculate

$$\begin{aligned}
&\|A_1\| \\
&\leq \frac{\eta_t}{N} \sum_{i=1}^N \sum_{k=0}^{K-1} \|\nabla f_i(\bar{\boldsymbol{w}}^t; \xi_i^{t,k}) - \nabla f_i(\boldsymbol{w}_i^{t,k}; \xi_i^{t,k})\| + \|\nabla f_i(\bar{\boldsymbol{v}}^t; \xi_i^{t,k}) - \nabla f_i(\boldsymbol{v}_i^{t,0}; \xi_i^{t,k})\| \\
&\leq \frac{\eta_t L}{N} \sum_{i=1}^N \sum_{k=0}^{K-1} \|\bar{\boldsymbol{w}}^t - \boldsymbol{w}_i^{t,k}\| + \|\bar{\boldsymbol{v}}^t - \boldsymbol{v}_i^{t,k}\| \\
&\leq \frac{\eta_t L}{N} \sum_{k=0}^{K-1} \sum_{i=1}^N \left(\|\bar{\boldsymbol{w}}^t - \boldsymbol{w}_i^{t,0}\| + \|\bar{\boldsymbol{v}}^t - \boldsymbol{v}_i^{t,0}\|\right)
\end{aligned}$$

$$
\begin{aligned}
&+\frac{\eta_t L}{N}\sum_{k=1}^{K-1}\sum_{i=1}^{N}\left(\|\boldsymbol{w}_i^{t,0}-\boldsymbol{w}_i^{t,k}\|+\|\boldsymbol{v}_i^{t,0}-\boldsymbol{v}_i^{t,k}\|\right)\\
&\le \frac{\eta_t L}{\sqrt{N}}\sum_{k=0}^{K-1}\left(\|\boldsymbol{X}^{(t,0)}(\boldsymbol{I}-\boldsymbol{P})\|_F+\|\boldsymbol{Y}^{(t,0)}(\boldsymbol{I}-\boldsymbol{P})\|_F\right)\\
&\qquad+\frac{\eta_t^2 L}{N(1-\theta)^2}\sum_{i=1}^{N}\sum_{k=1}^{K-1}\sum_{s=0}^{k-1}p_s(\theta)\left(\|\nabla f_i(\boldsymbol{w}_i^{t,s};\xi_i^{t,s})\|+\|\nabla f_i(\boldsymbol{v}_i^{t,s};\xi_i^{t,s})\|\right)\\
&\le \frac{\eta_t L}{\sqrt{N}}\sum_{k=0}^{K-1}\left(\|\boldsymbol{X}^{(t,0)}(\boldsymbol{I}-\boldsymbol{P})\|_F+\|\boldsymbol{Y}^{(t,0)}(\boldsymbol{I}-\boldsymbol{P})\|_F\right)\\
&\qquad+\frac{\eta_t^2 L}{N}\sum_{i=1}^{N}\sum_{k=1}^{K-1}\sum_{s=0}^{k-1}\left(\|\nabla f_i(\boldsymbol{w}_i^{t,s};\xi_i^{t,s})\|+\|\nabla f_i(\boldsymbol{v}_i^{t,s};\xi_i^{t,s})\|\right).
\end{aligned}
$$

Then, we have

$$
\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N}\|\nabla f_i(\boldsymbol{w}_i^{t,s};\xi_i^{t,s})\| &\le \frac{1}{N}\sum_{i=1}^{N}\|\nabla f_i(\boldsymbol{w}_i^{t,s};\xi_i^{t,s})-\nabla f_i(\boldsymbol{w}_i^{t,s})\|+\|\nabla f_i(\boldsymbol{w}_i^{t,s})\|\\
&\le \frac{1}{\sqrt{N}}\left(\sum_{i=1}^{N}\|\nabla f_i(\boldsymbol{w}_i^{t,s};\xi_i^{t,s})-\nabla f_i(\boldsymbol{w}_i^{t,s})\|^2\right)^{\frac{1}{2}}+B\\
\implies \mathbb{E}\frac{1}{N}\sum_{i=1}^{N}\|\nabla f_i(\boldsymbol{w}_i^{t,s};\xi_i^{t,s})\| &\le \frac{1}{\sqrt{N}}\left(\sum_{i=1}^{N}\mathbb{E}\|\nabla f_i(\boldsymbol{w}_i^{t,s};\xi_i^{t,s})-\nabla f_i(\boldsymbol{w}_i^{t,s})\|^2\right)^{\frac{1}{2}}+B\\
&\le \frac{1}{\sqrt{N}}\left(N\sigma^2\right)^{\frac{1}{2}}+B=\sigma+B,
\end{aligned}
$$

so that by applying Lemma A.2 we obtain

$$
\begin{aligned}
\|A_1\| &\le 2\eta_t LK(\sigma+B)\left(\sum_{j=1}^{t}\eta_{t-j}\lambda^j\right)+2\eta_t^2 L(\sigma+B)\sum_{k=1}^{K-1}k\\
&= 2\eta_t LK(\sigma+B)\left(\sum_{j=1}^{t}\eta_{t-j}\lambda^j\right)+\eta_t^2 L(\sigma+B)K(K-1)\\
&\le \eta_t LK(\sigma+B)\left(2\sum_{j=1}^{t-1}\eta_{t-j}\lambda^j+\eta_t K\right).
\end{aligned}
$$

Now, noticing that with each f_i being L-smooth, we can calculate

$$
\begin{aligned}
&\left\|\bar{\boldsymbol{w}}^t-\frac{\eta_t}{(1-\theta)^2}\sum_{k=0}^{K-1}p_k(\theta)\nabla f_i(\bar{\boldsymbol{w}}^t;\xi_i^{t,k})-\left(\bar{\boldsymbol{v}}^t-\frac{\eta_t}{(1-\theta)^2}\sum_{k=0}^{K-1}p_k(\theta)\nabla f_i(\bar{\boldsymbol{v}}^t;\xi_i^{t,k})\right)\right\|\\
&\qquad\le \|\bar{\boldsymbol{w}}^t-\bar{\boldsymbol{v}}^t\|+\eta_t\sum_{k=0}^{K-1}\left\|\nabla f_i(\bar{\boldsymbol{w}}^t;\xi_i^{t,k})-\nabla f_i(\bar{\boldsymbol{v}}^t;\xi_i^{t,k})\right\|\\
&\qquad\le (1+\eta_t LK)\|\bar{\boldsymbol{w}}^t-\bar{\boldsymbol{v}}^t\|.
\end{aligned}
$$

Plugging everything into (A.2), we obtain

$$\mathbb{E}\|\bar{\boldsymbol{w}}^{t+1}-\bar{\boldsymbol{v}}^{t+1}\| \leq (1+\eta_t LK)\|\bar{\boldsymbol{w}}^t-\bar{\boldsymbol{v}}^t\| + \eta_t LK(\sigma+B)\left(2\sum_{j=1}^{t-1}\eta_{t-j}\lambda^j+\eta_t K\right).$$

Now, with probability $1-\left(\frac{n-1}{n}\right)^K$, we have that the random variables $\{\tilde{\Xi}^{(t,k)}\}_{k=1}^K$ might be different from $\{\Xi^{(t,k)}\}_{k=1}^K$ *in the draws from node* i^*. We calculate, similarly to the previous case,

$$\begin{aligned}
&\bar{\boldsymbol{w}}^{t+1}-\bar{\boldsymbol{v}}^{t+1}\\
&=\frac{\eta_t}{N(1-\theta)^2}\sum_{i=1}^{N}\sum_{k=0}^{K-1}p_k(\theta)\left[\left(\nabla f_i(\bar{\boldsymbol{w}}^t;\xi_i^{t,k})-\nabla f_i(\boldsymbol{w}_i^{t,k};\xi_i^{t,k})\right)-\left(\nabla f_i(\bar{\boldsymbol{v}}^t;\tilde{\xi}_i^{t,k})-\nabla f_i(\boldsymbol{v}_i^{t,k};\tilde{\xi}_i^{t,k})\right)\right]\\
&\quad+\frac{1}{N}\sum_{i=1}^{N}\left(\bar{\boldsymbol{w}}^t-\frac{\eta_t}{(1-\theta)^2}\sum_{k=0}^{K-1}p_k(\theta)\nabla f_i(\bar{\boldsymbol{w}}^t;\xi_i^{t,k})\right)-\left(\bar{\boldsymbol{v}}^t-\frac{\eta_t}{(1-\theta)^2}\sum_{k=0}^{K-1}p_k(\theta)\nabla f_i(\bar{\boldsymbol{v}}^t;\tilde{\xi}_i^{t,k})\right),
\end{aligned}$$

which allows us to conclude by performing the same calculations we did on A_1:

(A.4)
$$\implies \mathbb{E}\|\bar{\boldsymbol{w}}^{t+1}-\bar{\boldsymbol{v}}^{t+1}\| \leq \eta_t LK(\sigma+B)\left(2\sum_{j=1}^{t-1}\eta_{t-j}\lambda^j+\eta_t K\right)$$

(A.5)
$$+\mathbb{E}\Bigg\|\underbrace{\frac{1}{N}\sum_{i=1}^{N}\left(\bar{\boldsymbol{w}}^t-\frac{\eta_t}{(1-\theta)^2}\sum_{k=0}^{K-1}p_k(\theta)\nabla f_i(\bar{\boldsymbol{w}}^t;\xi_i^{t,k})\right)-\left(\bar{\boldsymbol{v}}^t-\frac{\eta_t}{(1-\theta)^2}\sum_{k=0}^{K-1}p_k(\theta)\nabla f_i(\bar{\boldsymbol{v}}^t;\tilde{\xi}_i^{t,k})\right)}_{=:A_2}\Bigg\|.$$

Now, turning our attention to the term A_2, we can calculate

$$\begin{aligned}
A_2&=\frac{1}{N}\sum_{i=1}^{N}\left[\bar{\boldsymbol{w}}^t-\bar{\boldsymbol{v}}^t+\frac{\eta_t}{(1-\theta)^2}\sum_{k=0}^{K-1}p_k(\theta)\left(\nabla f_i(\bar{\boldsymbol{w}}^t;\xi_i^{t,k})-\nabla f_i(\bar{\boldsymbol{v}}^t;\tilde{\xi}_i^{t,k})\right)\right]\\
&=\bar{\boldsymbol{w}}^t-\bar{\boldsymbol{v}}^t+\frac{\eta_t}{N(1-\theta)^2}\sum_{i\neq i^*}\sum_{k=0}^{K-1}p_k(\theta)\left(\nabla f_i(\bar{\boldsymbol{w}}^t;\xi_i^{t,k})-\nabla f_i(\bar{\boldsymbol{v}}^t;\xi_i^{t,k})\right)\\
&\qquad+\frac{\eta_t}{N(1-\theta)^2}\sum_{k=0}^{K-1}p_k(\theta)\left(\nabla f_{i^*}(\bar{\boldsymbol{w}}^t;\xi_{i^*}^{t,k})-\nabla f_{i^*}(\bar{\boldsymbol{v}}^t;\tilde{\xi}_{i^*}^{t,k})\right)\\
\implies \|A_2\| &\leq \|\bar{\boldsymbol{w}}^t-\bar{\boldsymbol{v}}^t\|+\frac{\eta_t LK(m-1)}{N}\|\bar{\boldsymbol{w}}^t-\bar{\boldsymbol{v}}^t\|+\frac{\eta_t}{N}\sum_{k=0}^{K-1}\|\nabla f_{i^*}(\bar{\boldsymbol{w}}^t;\xi_{i^*}^{t,k})-\nabla f_{i^*}(\bar{\boldsymbol{w}}^t)\|\\
&\qquad+\frac{\eta_t}{N}\sum_{k=0}^{K-1}\|\nabla f_{i^*}(\bar{\boldsymbol{w}}^t)-\nabla f_{i^*}(\bar{\boldsymbol{v}}^t)\|+\|\nabla f_{i^*}(\bar{\boldsymbol{v}}^t)-\nabla f_{i^*}(\bar{\boldsymbol{v}}^t;\tilde{\xi}_{i^*}^{t,k})\|\\
\implies \mathbb{E}\|A_2\| &\leq \left(1+\frac{\eta_t LK(m-1)}{N}\right)\mathbb{E}\|\bar{\boldsymbol{w}}^t-\bar{\boldsymbol{v}}^t\|+\frac{\eta_t K}{N}\left(2\sigma+L\mathbb{E}\|\bar{\boldsymbol{w}}^t-\bar{\boldsymbol{v}}^t\|\right)\\
&=(1+\eta_t LK)\,\mathbb{E}\|\bar{\boldsymbol{w}}^t-\bar{\boldsymbol{v}}^t\|+\frac{2\eta_t\sigma K}{N},
\end{aligned}$$

where in the second to last line we have used the fact that by Jensen's inequality

$$\mathbb{E}\left(\|\mathbf{z}\|^2\right)^{\frac{1}{2}} \leq \left(\mathbb{E}\|\mathbf{z}\|^2\right)^{\frac{1}{2}}$$

combined with the assumption of bounded variance of stochastic gradients.

By defining $\delta_t := \|\bar{\boldsymbol{w}}^t - \bar{\boldsymbol{v}}^t\|$ and $t_0 \in \{1, 2, \ldots, n\}$ as in Lemma A.3, we can combine both cases to obtain

$$\begin{aligned}
&\mathbb{E}(\delta_{t+1}|\delta_{t_0}=0)\\
&\le \left(\frac{n-1}{n}\right)^K \left((1+\eta_t LK)\mathbb{E}(\delta_t|\delta_{t_0}=0) + \eta_t LK(\sigma+B)\left(2\sum_{j=1}^{t-1}\eta_{t-j}\lambda^j + \eta_t K\right)\right)\\
&+ \left(1-\left(\frac{n-1}{n}\right)^K\right)\left(\eta_t LK(\sigma+B)\left(2\sum_{j=1}^{t-1}\eta_{t-j}\lambda^j + \eta_t(K+1)\right)\right)\\
&\quad + \left(1-\left(\frac{n-1}{n}\right)^K\right)\left((1+\eta_t LK)\,\mathbb{E}(\delta_t|\delta_{t_0}=0) + \frac{2\eta_t\sigma K}{N}\right)\\
&= (1+\eta_t LK)\,\mathbb{E}(\delta_t|\delta_{t_0}=0) + \eta_t LK(\sigma+B)\left(2\sum_{j=1}^{t-1}\eta_{t-j}\lambda^j + \eta_t K\right)\\
&\quad + \left(1-\left(\frac{n-1}{n}\right)^K\right)\frac{2\eta_t\sigma K}{N}.
\end{aligned}$$

With a similar result to bound the sum $\sum_{j=1}^{t-1}\eta_{t-j}\lambda^j$ to that of [59], if we set

$$\eta_t \le \frac{c}{t},$$

then we should be able to calculate

$$\begin{aligned}
\mathbb{E}(\delta_{t+1}|\delta_{t_0}=0) &\le \left(1+\frac{cLK}{t}\right)\mathbb{E}(\delta_t|\delta_{t_0}=0) + \frac{cLK}{t}(\sigma+B)\left(2\frac{C_\lambda}{t}+\frac{cK}{t}\right)\\
&\quad + \left(1-\left(\frac{n-1}{n}\right)^K\right)\frac{2c\sigma K}{Nt}\\
&\le \left(1+\frac{cLK}{t}\right)\mathbb{E}(\delta_t|\delta_{t_0}=0) + \underbrace{\frac{2cK\sigma}{N}}_{=:C_1}\frac{1}{t} + \underbrace{cLK(\sigma+B)\,(cK+2C_\lambda)}_{=:C_2}\frac{1}{t^2}\\
&\le \exp\left(\frac{cLK}{t}\right)\mathbb{E}(\delta_t|\delta_{t_0}=0) + \frac{C_1}{t} + \frac{C_2}{t^2}.
\end{aligned}$$

Unraveling this recursion, we obtain

$$\begin{aligned}
\mathbb{E}(\delta_T|\delta_{t_0}=0) &\le \sum_{t=t_0+1}^{T}\exp\left(cLK\sum_{k=t+1}^{T}\frac{1}{k}\right)\left(\frac{C_1}{t}+\frac{C_2}{t^2}\right)\\
&\le \sum_{t=t_0+1}^{T}\exp\left(cLK\ln\frac{T}{t}\right)\left(\frac{C_1}{t}+\frac{C_2}{t^2}\right)\\
&= T^{cLK}\left(\sum_{t=t_0+1}^{T}\frac{1}{t^{cLK+1}}\left(C_1+\frac{C_2}{t}\right)\right)\\
&\le T^{cLK}\left(\frac{C_1}{cLKt_0^{cLK}} + \frac{C_2}{(cLK+1)t_0^{cLK+1}}\right)
\end{aligned}$$

$$\leq \left(\frac{T}{t_0}\right)^{cLK} \frac{1}{cLK}\left(C_1 + \frac{C_2}{t_0}\right).$$

Plugging in the definitions of C_1, C_2, we get

$$\begin{aligned}\mathbb{E}(\delta_T|\delta_{t_0}=0) &\leq \left(\frac{T}{t_0}\right)^{cLK} \frac{1}{cLK}\left[\frac{2cK\sigma}{N} + \frac{cLK(\sigma+B)\,(cK+2C_\lambda)}{t_0}\right] \\ &= \left(\frac{T}{t_0}\right)^{cLK}\left[\frac{2\sigma}{NL} + \frac{(\sigma+B)\,(cK+2C_\lambda)}{t_0}\right],\end{aligned}$$

which gives by Lemma A.3

$$\begin{aligned}\mathbb{E}|f(\bar{\boldsymbol{w}}^T;\Xi) - f(\bar{\boldsymbol{v}}^T;\Xi)| &\leq t_0(\sup f)\left(1-\left(\frac{n-1}{n}\right)^K\right) \\ &\quad + B\left(\frac{T}{t_0}\right)^{cLK}\left[\frac{2\sigma}{NL} + \frac{(\sigma+B)\,(cK+2C_\lambda)}{t_0}\right] \\ &\leq \frac{t_0 K}{n}(\sup f) + \left(\frac{T}{t_0}\right)^{cLK}\left[\frac{2\sigma B}{NL} + \frac{B(\sigma+B)\,(cK+2C_\lambda)}{t_0}\right].\end{aligned}$$

The right-hand side is approximately minimized if we choose

$$t_0 = T^{\frac{cLK}{1+cLK}}(cLK)^{\frac{1}{1+cLK}},$$

which we can ensure is less than n for c sufficiently small. We then can calculate

$$\begin{aligned}\mathbb{E}|f(\bar{\boldsymbol{w}}^T;\Xi) - f(\bar{\boldsymbol{v}}^T;\Xi)| &\leq \frac{(\sup f)K(cLK)^{\frac{1}{1+cLK}}}{n}T^{\frac{cLK}{1+cLK}} + \frac{\frac{2\sigma B}{NL}}{(cLK)^{\frac{cLK}{1+cLK}}}T^{\frac{cLK}{1+cLK}} \\ &\quad + \frac{B(\sigma+B)\,(cK+2C_\lambda)\,T^{cLK}}{\left((cLK)^{\frac{1}{1+cLK}}T^{\frac{cLK}{1+cLK}}\right)^{cLK+1}} \\ &= T^{\frac{cLK}{1+cLK}}\left(\frac{(\sup f)K(cLK)^{\frac{1}{1+cLK}}}{n} + \frac{\frac{2\sigma B}{NL}}{(cLK)^{\frac{cLK}{1+cLK}}}\right) \\ &\quad + \frac{B(\sigma+B)\,(cK+2C_\lambda)}{cLK},\end{aligned}$$

as desired. ∎

LEMMA A.2. *Under Assumptions* 1–4 *and on the mixing matrix* $\boldsymbol{M}$*, we have that*

$$\|\boldsymbol{X}^{(t,0)}(\boldsymbol{I}-\boldsymbol{P})\|_F \leq K\sqrt{N}(\sigma+B)\sum_{j=1}^{t-1}\eta_{t-j}\lambda^j,$$

where σ*,* B*, and* λ *are constants from our assumptions and* K *is the number of local updates performed before aggregation via the graph topology.*

Proof. Let the vector $\bar{\boldsymbol{w}}^{t,k} = \sum_{i=1}^N \boldsymbol{w}_i^{t,k}/N$ be the average parameter vector during intermediate, local updates. Then, let the matrix of "true gradients" of the global objective function f be

$$\nabla \boldsymbol{F}(\boldsymbol{X}^{(t,k)}) := [\nabla f(\bar{\boldsymbol{w}}^{t,k})\ \nabla f(\bar{\boldsymbol{w}}^{t,k})\ \ldots \nabla f(\bar{\boldsymbol{w}}^{t,k})],$$

which is obtained by horizontally concatenating the true gradient vector $\nabla f(\bar{\boldsymbol{w}}^{t,k})$. Recalling that $p_k(\theta) \leq (1-\theta)^2$ for $k = 0, \ldots, K-1$, we have

$$
\begin{aligned}
&\|\boldsymbol{X}^{(t,0)}(\boldsymbol{I}-\boldsymbol{P})\|_F \\
&= \left\| \sum_{j=1}^{t} \frac{\eta_{t-j}}{(1-\theta)^2} \sum_{k=0}^{K-1} p_k(\theta) \boldsymbol{G}^{(t,k)}(\boldsymbol{X}^{(t,k)}; \Xi^{(t,k)}) \left(\boldsymbol{M}^j - P \right) \right\|_F \\
&\leq \sum_{j=1}^{t} \eta_{t-j} \left\| \boldsymbol{M}^j - P \right\|_F \sum_{k=0}^{K-1} \left\| \boldsymbol{G}^{(t-j,k)}(\boldsymbol{X}^{(t-j,k)}; \Xi^{(t,k)}) - \nabla \boldsymbol{F}(\boldsymbol{X}^{(t-j,k)}) + \nabla \boldsymbol{F}(\boldsymbol{X}^{(t-j,k)}) \right\|_F \\
&\leq \sum_{j=1}^{t} \eta_{t-j} \lambda^j \sum_{k=0}^{K-1} \left(\sum_{i=1}^{N} \|\nabla f_i(\boldsymbol{w}_i^{t-j,k}; \xi_i^{t-j,k}) - \nabla f_i(\boldsymbol{w}_i^{t-j,k})\|^2 \right)^{\frac{1}{2}} + \left(\sum_{i=1}^{N} \|\nabla f_i(\boldsymbol{w}_i^{t-j,k})\|^2 \right)^{\frac{1}{2}} \\
&\leq \sum_{j=1}^{t} \eta_{t-j} \lambda^j \sum_{k=0}^{K-1} \left[\left(\sum_{i=1}^{N} \|\nabla f_i(\boldsymbol{w}_i^{t-j,k}; \xi_i^{t-j,k}) - \nabla f_i(\boldsymbol{w}_i^{t-j,k})\|^2 \right)^{\frac{1}{2}} + B\sqrt{N} \right],
\end{aligned}
$$

$$
\begin{aligned}
\implies \quad & \mathbb{E}\|\boldsymbol{X}^{(t,0)}(\boldsymbol{I}-\boldsymbol{P})\|_F \\
&\leq \sum_{j=1}^{t} \eta_{t-j} \lambda^j \sum_{k=0}^{K-1} \left[\left(\sum_{i=1}^{N} \mathbb{E}\|\nabla f_i(\boldsymbol{w}_i^{t-j,k}; \xi_i^{t-j,k}) - \nabla f_i(\boldsymbol{w}_i^{t-j,k})\|^2 \right)^{\frac{1}{2}} + B\sqrt{N} \right] \\
&\leq \sum_{j=1}^{t} \eta_{t-j} \lambda^j \sum_{k=0}^{K-1} \left[\sigma\sqrt{N} + B\sqrt{N} \right] \\
&= K\sqrt{N}(\sigma + B) \sum_{j=1}^{t} \eta_{t-j} \lambda^j,
\end{aligned}
$$

where in the third line from the bottom we have used Jensen's inequality, since the square root function is concave. ∎

LEMMA A.3. *Assume that the loss function $f(\cdot; \Xi)$ is nonnegative and B-Lipschitz for all Ξ. Let $\mathcal{D}$, $\tilde{\mathcal{D}}$ be two samples of size Nn differing in only a single example. Let $\bar{\boldsymbol{w}}^T, \bar{\boldsymbol{v}}^T$ denote the output of DFedAvgM after T steps with the dataset samples $\mathcal{D}$ and $\tilde{\mathcal{D}}$, respectively. Then, for every Ξ and every $t_0 \in \{0, 1, \ldots, n\}$, under the random selection rule, we have*

$$
\mathbb{E}|f(\bar{\boldsymbol{w}}^T; \Xi) - f(\bar{\boldsymbol{v}}^T; \Xi)| \leq t_0(\sup f) \left(1 - \left(\frac{n-1}{n} \right)^K \right) + B\mathbb{E}(\delta_T | \delta_{t_0} = 0).
$$

Proof. Our proof closely follows that of Lemma 3.11 of [19], with only a small distinction. By virtue of the nonnegativity of f and the Lipschitz continuity assumption on f, we obtain the inequality

$$
\begin{aligned}
\mathbb{E}|f(\bar{\boldsymbol{w}}^T; \Xi) - f(\bar{\boldsymbol{v}}^T; \Xi)| &= \mathbb{P}\{\mathcal{E}\}\mathbb{E}(\delta_T | \mathcal{E}) + \mathbb{P}\{\mathcal{E}^c\}\mathbb{E}(\delta_T | \mathcal{E}^c) \\
&\leq B\mathbb{E}(\delta_T | \delta_{t_0} = 0) + \mathbb{P}\{\mathcal{E}^c\}(\sup f),
\end{aligned}
$$

where the event $\mathcal{E}$ denotes the event that $\delta_{t_0} := \|\bar{\boldsymbol{w}}^{t_0} - \bar{\boldsymbol{v}}^{t_0}\| = 0$. Now, we must bound $P\{\mathcal{E}^c\}$. Defining the random variable I to assume the index of the first time step in which DFedAvg uses the example ξ^*, which occurs at node $i^* \in [N]$ and is located in the $j^* \in [n]$ entry of $\tilde{\mathcal{D}}_{i^*}$, we have

$$
P\{\mathcal{E}^c\} = \mathbb{P}\{\delta_{t_0} \neq 0\} \leq \mathbb{P}\{I \leq t_0\} \leq \sum_{t=1}^{t_0} \mathbb{P}\{I = t\}.
$$

Now since the draws at each round t of DFedAvgM are sampled uniformly at random across both the nodes and the local datasets (that is, $\xi_i^{t,k} \sim Unif(D_i)$ with replacement across iterations k), we have that

$$\mathbb{P}\{I = t\} = 1 - \mathbb{P}\{I \neq t\} = 1 - \left(\frac{n-1}{n}\right)^K,$$

from which we conclude the proof. ∎

LEMMA A.4. *If $\eta_t \leq \frac{c}{t}$ for $t = 1, 2, \ldots,$ then*

$$\sum_{j=1}^{t} \eta_{t-j}\lambda^j \leq \frac{C_\lambda}{t},$$

where

$$C_\lambda := \min\left\{2\lambda, \frac{1}{\ln\frac{1}{\lambda}}\lambda^{\frac{1}{\ln\frac{1}{\lambda}}}\right\} + \min\left\{4\lambda\ln\frac{1}{\lambda}, \frac{4}{\ln\frac{1}{\lambda}}\lambda^{\frac{2}{\ln\frac{1}{\lambda}}}\right\} + \min\left\{2\lambda, \frac{1}{\ln\frac{1}{\lambda}}\lambda^{\frac{1}{\ln\frac{1}{\lambda}}}\right\} + \frac{2}{\ln\frac{1}{\lambda}}.$$

Proof. First, we can compute

$$\begin{aligned}
\sum_{j=1}^{t} \eta_{t-j}\lambda^j &= \sum_{j=1}^{t} \eta_j \lambda^{t-j} \leq \lambda^t \sum_{j=1}^{t} \frac{\lambda^{-j}}{j} = \lambda^t + \lambda^t \sum_{j=2}^{t} \frac{\lambda^{-j}}{j} \\
&\leq \lambda^t + \lambda^t \sum_{j=2}^{t} \int_{j-1}^{j} \frac{\lambda^{-x}}{x} dx \\
&= \lambda^t + \lambda^t \int_1^t \frac{\lambda^{-x}}{x} dx = \lambda^{t-1} + \lambda^t \left(\int_1^{t/2} \frac{\lambda^{-x}}{x} dx + \int_{t/2}^{t} \frac{\exp(x\ln(\frac{1}{\lambda}))}{x} dx\right), \qquad \text{(A.6)}
\end{aligned}$$

from which with $\lambda < 1$ we have that $\ln(1/\lambda) > 0$, and so we can simplify the integrals as

$$\int_1^{t/2} \frac{\lambda^{-x}}{x} dx \leq \frac{2}{t} \int_1^{t/2} \lambda^{-x} dx = \frac{2}{t\ln\frac{1}{\lambda}} \left(\lambda^{-t/2} - \lambda^{-t}\right) \leq \frac{2\lambda^{-t/2}}{t\ln\frac{1}{\lambda}}$$

and

$$\begin{aligned}
\int_{t/2}^{t} \frac{\exp(x\ln(\frac{1}{\lambda}))}{x}dx &= \int_1^{t/2} \frac{1}{x}\sum_{k=0}^{\infty}\frac{(\ln\frac{1}{\lambda})^k x^k}{k!}dx \\
&= \int_1^{t/2}\frac{1}{x}dx + \left(\ln\frac{1}{\lambda}\right)\int_1^{t/2} 1dx + \sum_{k=2}^{\infty}\frac{(\ln\frac{1}{\lambda})^k}{k!}\int_1^{t/2} x^{k-1}dx \\
&= 1 - \frac{4}{t^2} + \frac{1}{2}(t-2)\ln\frac{1}{\lambda} + \sum_{k=2}^{\infty}\frac{(\ln\frac{1}{\lambda})^k}{(k)k!}\left(\left(\frac{t}{2}\right)^k - 1\right) \\
&\le 1 - \frac{4}{t^2} + \frac{1}{2}(t-2)\ln\frac{1}{\lambda} + \frac{1}{2}\sum_{k=2}^{\infty}\frac{(\ln\frac{1}{\lambda})^k}{k!}\left(\left(\frac{t}{2}\right)^k - 1\right) \\
&= 1 - \frac{4}{t^2} + \frac{1}{2}(t-2)\ln\frac{1}{\lambda} \\
&\qquad + \frac{1}{2}\left(\frac{1}{\lambda^{t/2}} - 1 + \frac{t}{2}\ln\frac{1}{\lambda} - \left[\frac{1}{\lambda} - 1 + \ln\frac{1}{\lambda}\right]\right) \\
&= 1 - \frac{4}{t^2} + \frac{1}{2}(t-2)\ln\frac{1}{\lambda} + \frac{1}{2}\left(\frac{1}{\lambda^{t/2}} - \frac{1}{\lambda} + \frac{1}{2}(t-2)\ln\frac{1}{\lambda}\right) \\
&= 1 - \frac{4}{t^2} + \frac{3}{4}(t-2)\ln\frac{1}{\lambda} + \frac{1}{2\lambda^{t/2}} - \frac{1}{2\lambda}.
\end{aligned}$$

Plugging this result into (A.6), we obtain

$$\begin{aligned}
\sum_{j=1}^{t}\eta_{t-j}\lambda^j &\le \lambda^{t-1} + \lambda^t\left(1 - \frac{4}{t^2} + \frac{3}{4}(t-2)\ln\frac{1}{\lambda} + \frac{1}{2\lambda^{t/2}} - \frac{1}{2\lambda} + \frac{2\lambda^{-t/2}}{t\ln\frac{1}{\lambda}}\right) \\
&= \frac{\lambda^{t-1}}{2} + \lambda^t\left(1 - \frac{4}{t^2} + \frac{3}{4}(t-2)\ln\frac{1}{\lambda}\right) + \lambda^{t/2}\left(\frac{1}{2} + \frac{2}{t\ln\frac{1}{\lambda}}\right) \\
&\le \frac{1}{t}\left[\lambda^t\left(t + \ln\frac{1}{\lambda}t^2\right) + \lambda^{t/2}\left(t + \frac{2}{\ln\frac{1}{\lambda}}\right)\right], \qquad \text{(A.7)}
\end{aligned}$$

where in the last line we have used that $t - 1 \ge t/2$ for $t \ge 2$.

Seeking a uniform bound over $t = 2, 3, \ldots$, we bound each of the last two terms of the right-hand side of the above equation. It is easy to check that

$$\begin{aligned}
t\lambda^t &\le \min\left\{2\lambda^2, \frac{1}{\ln\frac{1}{\lambda}}\lambda^{\frac{1}{\ln\frac{1}{\lambda}}}\right\}, \\
t^2\lambda^t &\le \min\left\{4\lambda^2, \frac{4}{\left(\ln\frac{1}{\lambda}\right)^2}\lambda^{\frac{2}{\ln\frac{1}{\lambda}}}\right\}, \\
t\lambda^{t/2} &\le \min\left\{2\lambda, \frac{1}{\ln\frac{1}{\lambda}}\lambda^{\frac{1}{\ln\frac{1}{\lambda}}}\right\},
\end{aligned}$$

where we have noted that each of these functions is a decreasing function of t. There-

fore, our bound becomes

$$\begin{aligned}
&\sum_{j=1}^{t} \eta_{t-j}\lambda^j \\
&\le \frac{1}{t}\left[\min\left\{2\lambda^2, \frac{1}{\ln\frac{1}{\lambda}}\lambda^{\frac{1}{\ln\frac{1}{\lambda}}}\right\} + \min\left\{4\lambda^2\ln\frac{1}{\lambda}, \frac{4}{\ln\frac{1}{\lambda}}\lambda^{\frac{2}{\ln\frac{1}{\lambda}}}\right\} + \min\left\{2\lambda, \frac{1}{\ln\frac{1}{\lambda}}\lambda^{\frac{1}{\ln\frac{1}{\lambda}}}\right\}\right] \\
&\qquad + \frac{2}{t\ln\frac{1}{\lambda}} \\
&= \frac{1}{t}\left[2\lambda^2 + 4\lambda^2\ln\frac{1}{\lambda} + \frac{2}{\ln\frac{1}{\lambda}} + \min\left\{2\lambda, \frac{1}{\ln\frac{1}{\lambda}}\lambda^{\frac{1}{\ln\frac{1}{\lambda}}}\right\}\right] \\
&=: \frac{C_\lambda}{t}.
\end{aligned}$$

We note that all terms of C_λ except for $2/\ln\frac{1}{\lambda}$ are *uniformly bounded* on $\lambda \in (0,1)$. It is true that $2/\ln\frac{1}{\lambda} \to \infty$ as $\lambda \to 1^-$, but for each $\lambda < 1$ this bound C_λ is valid. ∎

REFERENCES

[1] W. Abramson, A. J. Hall, P. Papadopoulos, N. Pitropakis, and W. J. Buchanan, *A distributed trust framework for privacy-preserving machine learning*, in Proceedings of the International Conference on Trust and Privacy in Digital Business, Springer, New York, 2020, pp. 205–220.

[2] N. Alon, *Eigenvalues and expanders*, Combinatorica, 6 (1986), pp. 83–96.

[3] N. Alon, *On the edge-expansion of graphs*, Combin. Probab. Comput., 11 (1993), pp. 1–10.

[4] N. Alon, *Explicit expanders of every degree and size*, Combinatorica, 41 (2021), pp. 447–463.

[5] N. Alon, O. Schwartz, and A. Shapira, *An elementary construction of constant-degree expanders*, Combin. Probab. Comput., 17 (2008), pp. 319–327.

[6] M. F. Balcan, A. Blum, S. Fine, and Y. Mansour, *Distributed learning, communication complexity and privacy*, in Proceedings of the 25th Annual Conference on Learning Theory, JMLR Workshop and Conference Proceedings 23, PMLR, 2012, 26.

[7] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, *Practical Secure Aggregation for Federated Learning on User-Held Data*, preprint, https://arxiv.org/abs/1611.04482, 2016.

[8] S. Boyd, P. Diaconis, and L. Xiao, *Fastest mixing Markov chain on a graph*, SIAM Rev., 46 (2004), pp. 667–689, https://doi.org/10.1137/S0036144503423264.

[9] O. Choudhury, A. Gkoulalas-Divanis, T. Salonidis, I. Sylla, Y. Park, G. Hsu, and A. Das, *Anonymizing Data for Privacy-Preserving Federated Learning*, preprint, https://arxiv.org/abs/2002.09096, 2020.

[10] Y.-T. Chow, W. Shi, T. Wu, and W. Yin, *Expander graph and communication-efficient decentralized optimization*, in Proceedings of the 2016 50th Asilomar Conference on Signals, Systems and Computers, IEEE, Washington, DC, 2016, pp. 1715–1720.

[11] F. Chung, *Laplacians and the Cheeger inequality for directed graphs*, Ann. Combin., 9 (2005), pp. 1–19.

[12] F. R. Chung and F. C. Graham, *Spectral Graph Theory*, CBMS Reg. Conf. Ser. Math. 92, AMS, Providence, RI, 1997.

[13] A. Elisseeff, T. Evgeniou, M. Pontil, and L. P. Kaelbing, *Stability of randomized learning algorithms.*, J. Mach. Learn. Rese., 6 (2005), pp. 55–79.

[14] P. Erdös and A. Rényi, *On the evolution of random graphs*, Magyar Tud. Akad. Mat. Kutató Int. Közl., 5 (1960), pp. 17–61.

[15] T. Feder, A. Guetz, M. Mihail, and A. Saberi, *A local switch Markov chain on given degree graphs with application in connectivity of peer-to-peer networks*, in Proceedings of FOCS 2006, IEEE, Washington, DC, 2006, pp. 69–76.

[16] O. Fercoq, Z. Qu, P. Richtárik, and M. Takáč, *Fast distributed coordinate descent for non-strongly convex losses*, in Proceedings of the 2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), IEEE, Washington, DC, 2014, pp. 1–6.

[17] M. Fiedler, *Algebraic connectivity of graphs*, Czech. Math. J., 23 (1973), pp. 298–305.

[18] R. C. Geyer, T. Klein, and M. Nabi, *Differentially Private Federated Learning: A Client Level Perspective*, preprint, https://arxiv.org/abs/1712.07557, 2017.
[19] M. Hardt, B. Recht, and Y. Singer, *Train faster, generalize better: Stability of stochastic gradient descent*, in International Conference on Machine Learning, New York, NY, 2016.
[20] J. Heidemann, M. Klier, and F. Probst, *Online social networks: A survey of a global phenomenon*, Computer Networks, 56 (2012), pp. 3866–3878.
[21] S. Hoory, N. Linial, and A. Wigderson, *Expander graphs and their applications*, Bull. Amer. Math. Soc., 43 (2006), pp. 439–561.
[22] P. Jiang and G. Agrawal, *A linear speedup analysis of distributed deep learning with sparse and quantized communication*, in Advances in Neural Information Processing Systems, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds., NIPS 31, Curran Associates, Red Hook, NY, 2018, https://proceedings.neurips.cc/paper/2018/file/17326d10d511828f6b34fa6d751739e2-Paper.pdf.
[23] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., *Advances and Open Problems in Federated Learning*, preprint, https://arxiv.org/abs/1912.04977, 2019.
[24] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, *Scaffold: Stochastic controlled averaging for federated learning*, in International Conference on Machine Learning, PMLR, 2020, pp. 5132–5143.
[25] S. S. Lam and C. Qian, *Geographic routing in d-dimensional spaces with guaranteed delivery and low stretch*, in Proceedings of ACM SIGMETRICS, ACM, New York, 2011, pp. 257–268.
[26] C. Law and K.-Y. Siu, *Distributed construction of random expander networks*, in Proceedings of IEEE INFOCOM, IEEE, Washington, DC, 2003, pp. 2133–2143.
[27] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, *Federated optimization in heterogeneous networks*, in Proceedings of Machine Learning and Systems 2020 (MLSys 2020), 2020.
[28] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, *On the Convergence of FedAvg on Non-IID Data*, ICLR, 2020, https://iclr.cc/virtual_2020/poster_HJxNAnVtDS.html (accessed 2021-04-22).
[29] Z. Liang, B. Wang, Q. Gu, S. Osher, and Y. Yao, *Exploring Private Federated Learning with Laplacian Smoothing*, preprint, https://arxiv.org/abs/2005.00218, 2020.
[30] D. Liu and O. Simeone, *Privacy for free: Wireless federated learning via uncoded transmission with adaptive power control*, IEEE J. Sel. Areas Commun., 39 (2020), pp. 170–185.
[31] R. Liu, Y. Cao, M. Yoshikawa, and H. Chen, *FedSel: Federated SGD under local differential privacy with top-k dimension selection*, in International Conference on Database Systems for Advanced Applications, Springer, New York, 2020, pp. 485–501.
[32] Y. Liu, Y. Gao, and W. Yin, *An improved analysis of stochastic gradient descent with momentum*, in Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds., NIPS 33, Curran Associates, Red Hook, NY, 2020, pp. 18261–18271, https://proceedings.neurips.cc/paper/2020/file/d3f5d4de09ea19461dab00590df91e4f-Paper.pdf.
[33] Z. Liu, T. Li, V. Smith, and V. Sekar, *Enhancing the Privacy of Federated Learning with Sketching*, preprint, https://arxiv.org/abs/1911.01812, 2019.
[34] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, *A survey and comparison of peer-to-peer overlay network schemes*, IEEE Commun. Surveys Tutorials, 7 (2005), pp. 72–93.
[35] A. Lubotzky, *Expander graphs in pure and applied mathematics*, Bull. Amer. Math. Soc., 49 (2012), pp. 113–162.
[36] O. Marfoq, C. Xu, G. Neglia, and R. Vidal, *Throughput-optimal topology design for cross-silo federated learning*, in Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada, 2020.
[37] R. McDonald, K. Hall, and G. Mann, *Distributed training strategies for the structured perceptron*, in Human Language Technologies, The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010, pp. 456–464.
[38] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, *Communication-efficient learning of deep networks from decentralized data*, in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (Fort Lauderdale, FL), A. Singh and J. Zhu, eds., Proc. Mach. Learn. Res. 54, PMLR, 2017, pp. 1273–1282, http://proceedings.mlr.press/v54/mcmahan17a.html.
[39] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, *Exploiting unintended feature leakage in collaborative learning*, in 2019 IEEE Symposium on Security and Privacy (SP),

IEEE, Washington, DC, 2019, pp. 691–706.
[40] M. R. Murty, *Ramanujan graphs*, J. Ramanujan Math. Soc., 18 (2003), pp. 33–52.
[41] M. R. Murty, *Ramanujan graphs: An introduction*, Indian J. Discrete Math., 6 (2020), pp. 91–127.
[42] A. Nedic and A. Ozdaglar, *Distributed subgradient methods for multi-agent optimization*, IEEE Trans. Automatic Control, 54 (2009), pp. 48–61.
[43] M. W. Newman, *The Laplacian Spectrum of Graphs*, Master's thesis, University of Manitoba, Winnipeg, MB, Canada, 2001.
[44] M. A. Ogleari, Y. Yu, C. Qian, E. Miller, and J. Zhao, *String figure: A scalable and elastic memory network architecture*, in Proceedings of the IEEE International Symposium on High Performance Computer Architecture (HPCA), IEEE, Washington, DC, 2019, pp. 647–660.
[45] T. Orekondy, S. J. Oh, Y. Zhang, B. Schiele, and M. Fritz, *Gradient-Leaks: Understanding and Controlling Deanonymization in Federated Learning*, preprint, https://arxiv.org/abs/1805.05838, 2018.
[46] R. Pathak and M. J. Wainwright, *Fedsplit: An Algorithmic Framework for Fast Federated Optimization*, preprint, https://arxiv.org/abs/2005.05238, 2020.
[47] B. T. Polyak, *Some methods of speeding up the convergence of iterative methods*, Ž. Vyčisl. Mat. i Mat. Fiz., 4 (1964), pp. 791–803 (in Russian).
[48] D. Povey, X. Zhang, and S. Khudanpur, *Parallel Training of DNNs with Natural Gradient and Parameter Averaging*, preprint, https://arxiv.org/abs/1410.7455, 2014.
[49] C. Qian and S. Lam, *A scalable and resilient layer-2 network with ethernet compatibility*, IEEE/ACM Trans. Networking, 24 (2016), pp. 231–244.
[50] M. I. Qureshi, R. Xin, S. Kar, and U. A. Khan, *Push-saga: A decentralized stochastic algorithm with variance reduction over directed graphs*, IEEE Control Syst. Lett., 6 (2021), pp. 1202–1207.
[51] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, *Adaptive Federated Optimization*, preprint, https://arxiv.org/abs/2003.00295, 2020.
[52] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan, *Learnability, stability and uniform convergence*, J. Mach. Learn. Res., 11 (2010), pp. 2635–2670.
[53] O. Shamir and N. Srebro, *Distributed stochastic optimization and learning*, in Proceedings of the 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton, IL), IEEE, Washington, DC, 2014, pp. 850–857.
[54] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, *Jellyfish: Networking data centers randomly*, in Proceedings of USENIX NSDI, USENIX, Berkeley, CA, 2012, pp. 225–238.
[55] M. Sipser and D. Spielman, *Expander codes*, IEEE Trans. Inform. Theory, 42 (1996), pp. 1710–1722, https://doi.org/10.1109/18.556667.
[56] D. A. Spielman, *Constructing error-correcting codes from expander graphs*, in Emerging Applications of Number Theory, Springer, New York, 1999, pp. 591–600.
[57] S. U. Stich and S. P. Karimireddy, *The error-feedback framework: SGD with delayed gradients*, J. Mach. Learn. Res., 21 (2020), pp. 1–36, http://jmlr.org/papers/v21/19-748.html.
[58] T. Sun, D. Li, and B. Wang, *Stability and generalization of the decentralized stochastic gradient descent*, in Proceedings of the 34th AAAI Conference on Artificial Intelligence, 2021.
[59] T. Sun, D. Li, and B. Wang, *Decentralized Federated Averaging*, preprint, https://arxiv.org/abs/2104.11375, 2021.
[60] A. Triastcyn and B. Faltings, *Federated learning with Bayesian differential privacy*, in 2019 IEEE International Conference on Big Data (Big Data), IEEE, Washington, DC, 2019, pp. 2587–2596.
[61] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, *A hybrid approach to privacy-preserving federated learning*, in Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, ACM, New York, 2019, pp. 1–11.
[62] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei, *LDP-Fed: Federated learning with local differential privacy*, in Proceedings of the 3rd ACM International Workshop on Edge Systems, Analytics and Networking, ACM, New York, 2020, pp. 61–66.
[63] J. Wang and G. Joshi, *Cooperative SGD: A Unified Framework for the Design and Analysis of Communication-Efficient SGD Algorithms*, preprint, https://arxiv.org/abs/1808.07576, 2018.
[64] J. Wang, A. K. Sahu, Z. Yang, G. Joshi, and S. Kar, *MATCHA: Speeding up decentralized SGD via matching decomposition sampling*, in Proceedings of the Sixth Indian Control Conference (ICC), IEEE, Washington, DC, 2019, pp. 299–300.
[65] J.-K. Wang, C.-H. Lin, and J. Abernethy, *A Modular Analysis of Provable Acceleration*

via Polyak's Momentum: Training a Wide ReLU Network and a Deep Linear Network, prepring, https://arxiv.org/abs/2010.01618, 2020.

[66] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, *Federated learning with differential privacy: Algorithms and performance analysis*, IEEE Trans. Inform. Forensics Security, 15 (2020), pp. 3454–3469.

[67] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, and H. Ludwig, *Hybridalpha: An efficient approach for privacy-preserving federated learning*, in Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, ACM, New York, 2019, pp. 13–23.

[68] H. Yu, R. Jin, and S. Yang, *On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization*, in Proceedings of the 36th International Conference on Machine Learning, K. Chaudhuri and R. Salakhutdinov, eds., Proc. Mach. Learn. Res. 97, PMLR, 2019, pp. 7184–7193.

[69] Y. Yu and C. Qian, *Space shuffle: A scalable, flexible, and high-bandwidth data center network*, in Proceedings of the 22nd International Conference on Network Protocols, ICNP 2014, IEEE, Washington, DC, 2014, pp. 13–24.

[70] K. Yuan, Q. Ling, and W. Yin, *On the Convergence of Decentralized Gradient Descent*, preprint, https://arxiv.org/abs/1310.7063, 2013.

[71] S. Zhang, A. Choromanska, and Y. LeCun, *Deep learning with elastic averaging SGD*, preprint, https://arxiv.org/abs/1412.6651, 2014.

[72] X. Zhang, M. Hong, S. Dhople, W. Yin, and Y. Liu, *FedPD: A Federated Learning Framework with Optimal Rates and Adaptivity to Non-IID Data*, preprint, https://arxiv.org/abs/2005.11418, 2020.