# 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 clients failures in DFL with theoretical guarantees. Also, we present an efficient algorithm to convert a given graph to a practical overlay network and maintaining 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.

**AMS subject classifications.** 65B99, 68T01, 68T09, 68W15.

**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 [28, 15]. 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 [28], SCAFFOLD [16], FedProx [19], FedPD [52], FedSplit [33], and FedOpt [36]. Compared to many distributed optimization settings [30, 27, 2, 51, 34, 10, 37], FL gains tremendous advantages in communication efficiency. We can mathematically formulate FL as solving the following optimization problem

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

where $f_i(\boldsymbol{w}) = \mathbb{E}_{(\boldsymbol{x},y) \sim \mathcal{D}_i} \mathcal{L}(g(\boldsymbol{x}, \boldsymbol{w}), y)$ with $(\boldsymbol{x}, y)$ be a data-label pair sampled from the data distribution $\mathcal{D}_i$ on the $i^{th}$ client, and $g(\cdot, \boldsymbol{w})$ is the ML model. As shown in Fig. 1 (a), in the $i^{th}$ communication round, FedAvg [28], 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 $\{k_j | k_j \in \{1, 2, \cdots, N\}, \text{for } j = 1, 2, \cdots, m\}$. Each selected client then updates $\boldsymbol{w}_i$ for $T$ iterations by using its local data and stochastic

---

*Submitted to the editors DATE.

†Department of Computer Science and Engineering, University of California, Santa Cruz (yhua294@ucsc.edu) (Co-first).

‡Department of Mathematics, University of California, Los Angeles (millerk22@math.ucla.edu) (Co-first).

§Department of Mathematics, University of California, Los Angeles (bertozzi@math.ucla.edu) (Co-last).

¶Department of Computer Science and Engineering, University of California, Santa Cruz (cqian12@ucsc.edu) (Co-last).

‖Department of Mathematics, Scientific Computing and Imaging Institute, University of Utah (wangbaonj@gmail.com) (Co-last).

gradient-based algorithms. The server then aggregates these locally updated parameters to get the updated model after the current communication round. The existence of 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.
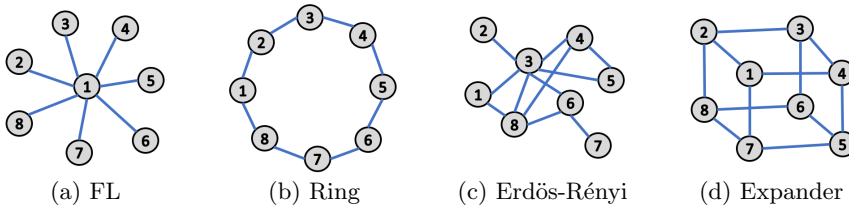


(a) FL      (b) Ring      (c) Erdös-Rényi      (d) Expander

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 [3, 11, 32, 43, 42, 48, 24, 29, 21, 5, 23, 44, 47, 22, 1, 41]. In DFL, all clients are connected by an overlay network, e.g. Fig. 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 resilient to client failures.*

**1.1    Our contribution**   Based on the theoretical convergence rate [49, 40, 41] 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 the random graph theory, we propose $d$-regular expander graphs for the network topology, which is provably to be near-optimal. The major advantages of leveraging $d$-regular expander graphs for the overlay networks design are threefolds:

- 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 graph connects each node with $d$ neighbors, resulting in low communication cost in the decentralized federated learning.
- DFL with $d$-regular expander graphs enables robust decentralized federated learning with respect to potential node failures.

**1.2    Additional related works**

*Network Design.* Chow et al. [6] have designed expander graphs for decentralized optimization using deterministic local optimization algorithms. In [26], 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 amount of devices.

*Analysis of DFL/FL algorithms.* The convergence properties of FedAvg or local SGD have been studied extensively [45, 39, 14], mainly focusing on the IID case. Non-IID convergence for FL has been shown in [14, 52, 49, 20]. Convergence analysis of DFL has been shown in [46, 41]. 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 [45] discuss how different versions of local SGD correspond to different graph topologies and [46] provides an efficient decomposition of graph topology for improved communication costs.

*Practical network construction.* Building overlay networks have been studied extensively in previous works. However, in the past, overlay networks are mainly used for peer-to-peer file sharing [25], online social networks [13], and routing infrastructures [17, 35]. For peer-to-peer file-sharing networks, existing studies have proposed to utilize random walks to achieve distributed $d$-regular expander graphs with assuming each node could choose $d$ neighbor at random [9, 18]. However, such 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 are given such as maintaining distributed Delaunay triangulation graphs for wireless sensor networks [17], metro Ethernet [35], random regular graphs for data center networks [50], and memory interconnection networks [31].

**1.3 Notations** We denote scalars by lower or upper case letters; vectors and matrices by lower and upper case boldface letters, respectively. For a vector $\boldsymbol{x} = (x_1, \cdots, x_d)^\top \in \mathbb{R}^d$, 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 $\ell_2$- and $\ell_\infty$-norm, respectively. We denote the vector whose entries are all 0s as $\boldsymbol{0}$. 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 $a_n = \mathcal{O}(b_n)$ if there exists a positive constant $C$ such that $a_n \leq Cb_n$.

**1.4 Organization** We organize this paper as follows: In Section 2, we present the theoretical results for DFL on convergence rate and generalization bound. Based on 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 $\mathcal{V} = \{1, 2, \cdots, N\} := [N]$ 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 $i \neq j$ 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) $null\{\boldsymbol{I} - \boldsymbol{M}\} = span\{\boldsymbol{1}\}$ where $\boldsymbol{I} \in \mathbb{R}^{N \times N}$ and $\boldsymbol{1} \in \mathbb{R}^N$ are the identity matrix and the vector whose entries are all 1s; 4. (Spectral property) $\boldsymbol{I} \succeq \boldsymbol{M} \succ -\boldsymbol{I}$, where $\boldsymbol{I} \succeq \boldsymbol{M}$ means $\boldsymbol{I} - \boldsymbol{M}$ is positive semi-definite and $\boldsymbol{M} \succ -\boldsymbol{I}$ stands for $\boldsymbol{M} + \boldsymbol{I}$ is positive definite.*

Given the adjacency matrix of a network, its maximum-degree matrix and metropolis-hastings matrix are both mixing matrices [4]. The symmetric property of $\boldsymbol{M}$ indicates that its eigenvalues are real and can be sorted in the non-increasing order. Let $\lambda_i(\boldsymbol{M})$

denote the $i^{th}$ 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 $\boldsymbol{M}$, converges to its stable state.

We consider DFL using the following update on client $i$

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

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 $K^{th}$ 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) [41].

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). $f_1, \ldots, f_m$ 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 $i^{th}$ 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 $i \in \{1, \ldots, m\}$, 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 [45, 39, 14, 41], 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 convergence results for DFedAvgM and 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 [41].

THEOREM 2.2 (General nonconvexity [41]). 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 1-3 hold. Moreover, assume the constant stepsize $\eta$ 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,

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

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

In this result we clearly see the convergence of the auxiliary sequence depends on the value of $\lambda \in (0, 1)$; namely, the closer that $\lambda$ is to 1, the worse the convergence bound of the final term of (2.2). In [49], a similar dependence on this graph-dependent value $\lambda$ appears in their convergence result for a slightly different version of DFL with momentum. All this motivates selecting a graph topology that will minimize the value of $\lambda$.

**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 for input $\boldsymbol{x}$ with loss function $f$. Uniform stability is a useful property used to bound the generalization error $\epsilon_{gen}$, see e.g. [12, 7].

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

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

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

LEMMA 2.4 ([12]). *Let $\mathcal{A}$ be $\epsilon$-uniformly stable, then it follows that*

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

Therefore, to ensure the generalization bound of a given random algorithm $\mathcal{A}$, we simply compute the uniform stability bound $\epsilon$. 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.

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

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

*where $\sup f < \infty$ is the uniform bound on the size of the non-negative 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, $\epsilon$. Again, we note here the *explicit dependence* of the generalization error on the corresponding value of $\lambda$ for the mixing matrix $\boldsymbol{M}$ of the graph topology. $C_\lambda$ is an increasing function of $\lambda \in (0, 1)$ which implies that the bound in (2.3) improves with smaller $\lambda$.

**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 $\lambda$ is to 1 the slower 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 $\lambda$ is far away from 1.

For the sake of notation, we recall graph definitions and properties to introduce network construction. Given an undirect, 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 reflecting 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 $0 = \lambda_1(\boldsymbol{L}) < \lambda_2(\boldsymbol{L}) \leq \lambda_3(\boldsymbol{L}) \leq \ldots \leq \lambda_N(\boldsymbol{L})$. Define the *reduced condition number* of $\boldsymbol{L}$ as

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

which is a measure of graph connectivity because a *smaller* $\kappa(\boldsymbol{L})$ corresponds to a graph with *higher* connectivity. 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 apply the mixing matrix used in [6]

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

which allows us to quantify the associated value of $\lambda$. The eigenvalues of this mixing matrix $\boldsymbol{M}$ have a straightforward relationship with eigenvalues of $\boldsymbol{L}$:

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

Then it is clear that

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

For a fixed $\kappa(\boldsymbol{L})$, we can view $\lambda$ as a function of $\theta$ that can be optimized to lead to the lowest value of $\lambda$. In Fig. 2, we have plotted the function $|\lambda_m(\boldsymbol{M})| = (1-\theta)/(1+\theta)$ 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 $|\lambda_2(\boldsymbol{M})| = |\lambda_m(\boldsymbol{M})|$, shown
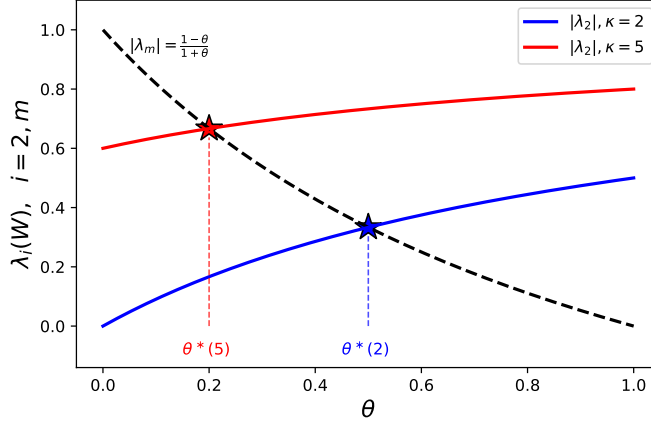
Fig. 2: Plot of $\lambda$ for the expander graph as a function of $\theta$. The stars indicate the optimal choice of $\theta = \theta^*(\kappa)$, given the value of $\kappa(\boldsymbol{L})$. *Lower* $\kappa$ leads to a *higher* value of $\theta^*(\kappa)$, which in turn leads to a *lower* value of $\lambda$, which is desired.

in Fig. 2 as stars; that is, when $\theta = \theta^*(\kappa(\boldsymbol{L}))$. It is clear that

$$|\lambda_2(\boldsymbol{M})| = |\lambda_m(\boldsymbol{M})|$$

$$\iff \frac{1 + \theta^*(\kappa(\boldsymbol{L})) - \frac{2}{\kappa(\boldsymbol{L})}}{1 + \theta^*(\kappa(\boldsymbol{L}))} = \frac{1 - \theta^*(\kappa(\boldsymbol{L}))}{1 + \theta^*(\kappa(\boldsymbol{L}))}$$

$$\iff \theta^*(\kappa(\boldsymbol{L})) - \frac{2}{\kappa(\boldsymbol{L})} = -\theta^*(\kappa(\boldsymbol{L}))$$

$$\iff \theta^*(\kappa(\boldsymbol{L})) = \frac{1}{\kappa(\boldsymbol{L})},$$

as long as $2/\kappa(\boldsymbol{L}) - \theta^*(\kappa(\boldsymbol{L})) \leq 1$, which is reasonable since most $\kappa(\boldsymbol{L}) \geq 2$.

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.* 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 (captured in the constant $\kappa(L)$) 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 $N$ [6].

For $d$-regular graphs, there exists a convenient upper bound for $\kappa(\boldsymbol{L})$. This bound

involves the first non-trivial eigenvalue, $\lambda_1(\boldsymbol{A})$, of the corresponding adjacency matrix

$$\kappa(\boldsymbol{L}) \leq \frac{d + \lambda_1(\boldsymbol{A})}{d - \lambda_1(\boldsymbol{A})}.$$

If the $d$-regular graph in question is Ramanujan, then we can bound $\kappa(\boldsymbol{L})$ as

$$(3.2) \qquad\qquad \kappa^R(\boldsymbol{L}) \leq \frac{d + 2\sqrt{d-1}}{d - 2\sqrt{d-1}}.$$

As the right-hand side of (3.2) is a *decreasing* function of $d$, this would suggest to choose larger $d$ in order to minimize $\kappa^R(\boldsymbol{L})$. 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 are in a sense "optimal"* by possessing *strong connectivity* properties in the graph topology while requiring *low communication cost* for local node neighborhood communication (i.e. sparsity).

*Ring graphs – poor connectivity.* Ring graph is an extremely sparse, but still connected, 2-regular graph structure where the graph structure constitutes a ring (see Fig. 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

$$\{\lambda_k(\boldsymbol{L}_{ring})\}_{k=0}^{\frac{N}{2}} = \left\{ 2 - 2\cos\left(\frac{2\pi k}{N}\right) \right\}_{k=0}^{\frac{N}{2}},$$

each with geometric multiplicity 2, except for the first eigenvalue $\mu_0(\boldsymbol{L}_{ring}) = 0$ which has geometric multiplicity 1; if $N$ is even, the last eigenvalue $\lambda_{N/2}(\boldsymbol{L}_{ring})$ has multiplicity 1 as well. Therefore, we can straightforwardly see that the reduced condition number for a Ring graph on $N$ nodes is

$$\begin{aligned}
\kappa^{ring}(\boldsymbol{L}_{ring}) &= \frac{\lambda_{N/2}(\boldsymbol{L}_{ring})}{\lambda_1(\boldsymbol{L}_{ring})} = \frac{2 - 2\cos\left(\frac{2\pi N}{2N}\right)}{2 - 2\cos\left(\frac{2\pi}{N}\right)} \\
&= \frac{4}{2 - 2\cos\left(\frac{2\pi}{N}\right)} \geq \frac{4}{2 - 2\left(1 - \frac{1}{2}\left(\frac{2\pi}{N}\right)^2\right)} = \frac{4N^2}{4\pi^2} = \frac{N^2}{\pi^2}.
\end{aligned}$$

Therefore, we see that with this *lower bound* for the Ring graph has a $\kappa(\boldsymbol{L}_{ring})$ grows *quadratically* with the size $N$ of the graph! The corresponding value for $\lambda$ 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

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

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 (ER) random graph structure, wherein each edge $(i, j) \in \mathcal{V} \times \mathcal{V}$ is sampled independently and indentically distributed with probability $p \in (0, 1)$. It is well-known that as long as $p = \mathcal{O}(\ln N / N)$, then the resulting graph $\mathcal{G}$ is connected with high probability [8].

While the connectivity properties of ER graphs are nearly guaranteed to be better than $d$-regular Ramanujan graphs (with $d$ is relatively small), *the communication cost of ER 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 ER graph with large enough edge probability $p$ is simply $\bar{d} = Np = \mathcal{O}(\ln N)$, 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 ER 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 ER 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 Fig. 3: for $d$-regular graph suppose $d$ is even and let $L = d/2$, we assign for each node a set of *virtual coordinates* represented by a $L$-dimensional vector $\langle x_1, x_2, ..., x_L \rangle$, where each element $x_i$ is a randomly generated real number $0 \le x_i < 1$, as shown in Fig. 3 (a). There are $L$ virtual ring spaces such as the two shown in Fig. 3 (b). In the $i^{th}$ space, a node is *virtually* placed on a ring based on the value of its $i^{th}$ coordinate $x_i$. Coordinates in each space are circular, and 0 and 1 are superposed. For each space, a node will connect to the two adjacent nodes, for example, $B$ connects to $A$ and $C$ in Space 1 and $G$ and $F$ in Space 2. Hence each node has at most $d = 2L$ neighbors. A neighbor of a node may happen to be adjacent to it in multiple spaces, such as $A$ and $D$. In such case, $A$ can connect to another node in the same situation, such as $E$. In the end, the equivalent network topology is shown in Fig. 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
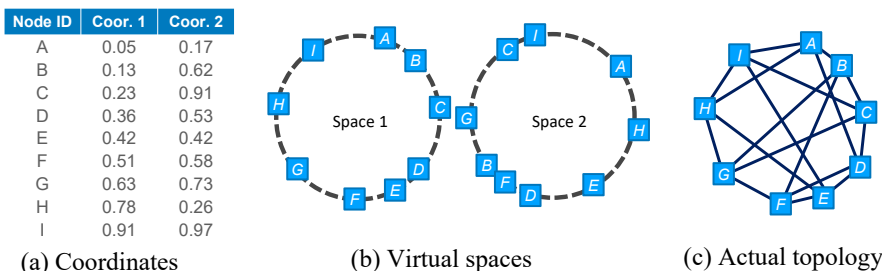
Fig. 3: DFL network topology in working systems. Each node generates a set of coordinates and the network is generated in a distributed manner by allowing each node to execute the proposed protocols locally.

The proposed network is a close proximal construction for a random $d$-regular network [50]. Note that in practice there does not exist a perfect construction of a random $d$-regular graph [38], 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 [17].

**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 communication round versus training loss, test loss, test accuracy, and the communication cost for Ring, Erdös-Rényi, fully-connected, and the proposed expander graphs. We pick $d = 3$ regular expander graphs (called Ramanujan). The communication cost could be estimated by the model size. In all experimental settings, the topology is generated by a central server before the training starts and stored in each user, but the central host are not involved in the actual training process. The expander graph is generated by adding an extra edge on top of the Ring graph. 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 expander graphs result in faster convergence and better generalization of DFedAvgM in training different models on different datasets. To conduct more comprehensive and solid experiments and testing, both the real network settings and the simulation are used in our evaluation. To exclude other factors no tuned optimization and data compression algorithms are used in the experiments. In the evaluation, fully-connected graphs are shown as a baseline but it is hardly practical in real world applications considering the communication cost and availability. Ring topology is easy to implement and widely used in previous works, so it is also shown as a baseline. Because of the randomness of the Erdös-Rényi graph, the experimental results are inconsistent when there are relatively few nodes; and so we do not include the Erdös-Rényi graph in all MNIST experiments below.

*MNIST IID.* We randomly split the MNIST dataset without any biases into 10 different subsets. Each user owns a local multilayer perceptron (MLP) model with one hidden layer of size 200. Each user only has access to only one local subset as its training set. We train the local model with the batch size of 20 and use the cross entropy as the loss function. We use SGD with the learning rate 0.01 and the momentum 0.9. After 3 epochs of local training, all the local nodes communicate with the topological neighbors and average all the parameters of the MLP model. After each communication round, the test accuracy, test loss, and training loss of each user are recorded and averaged in Fig. 4. Based on our experiments, in this IID settings the fully-connected and expander graph converge at round 16 which have advantage over 26 rounds of the Ring graph. According to the test accuracy shown in Fig. 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 its communication cost. The Ring graph reaches 97.7% accuracy due to the ideal distribution of the data.

*MNIST Non-IID.* All the settings are similar to the IID settings except each node owns a local dataset consisting of only one label (one digit in MNIST). The distribution is extremely unfavorable to the generalization. The test dataset is balanced sampled
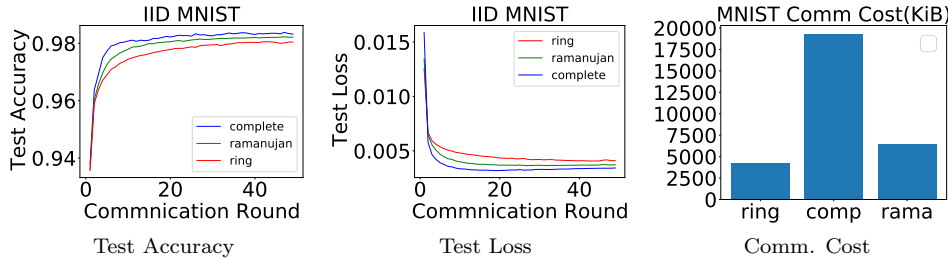
Fig. 4: The test accuracy, test loss, and communication cost of the Ring/3-regular expander (Ramanujan)/Fully connected graphs on IID MNIST. All of the graphs reach over 92% accuracy but the expander graph starts to converge at round 12 while the Ring graph starts to converge at round 20.

from the original dataset as the IID settings. As shown in Fig. 5 the expander graph reaches 88.8% accuracy and much higher than the Ring graph (73.68%). The fully connected graph reaches the best accuracy of 94%. Although the expander graph's accuracy is lower than the fully connected graphs' but with 33% of its communication cost. After each communication round, the training and test loss, and test accuracy of each user are recorded and averaged (Fig. 5). The expander graph could achieve a faster convergence and better generalization than the Ring graph and the performance is close to the fully connected graph but with a more manageable communication cost.
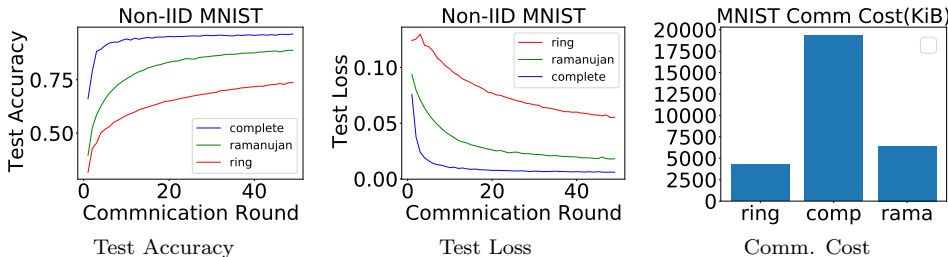


Fig. 5: The test accuracy, test loss, and communication cost of the Ring/3-regular Ramanujan/Fully connected Graph on non-IID MNIST. The expander graph reaches 88.79% accuracy, which is higher than the Ring graph's 73.68%. The communication cost of the expander graph is only one third of the fully connected graphs'.

*Language modeling.* We further conduct the simulation to evaluate the effect of different topology to the language models. First, we split the Shakespeare dataset [28] into 100 subsets (with some overlaps). Then we create 100 LSTM models (each one with 256 hidden units and 2 layers). In this Non-IID sampling scenario, the underlying distribution of data for each node is consistent with the raw data. Since we assume that data distributions vary between users in the raw data, we take this setting as Non-IID. We use cross entropy as the loss function. Then we train each LSTM with the corresponding local Non-IID dataset with the learning rate 0.5 and momentum 0.9. After 3 epochs of local training, all local nodes communicate with the neighbors through which is similar to the previous method employed by the MNIST experiment and average all the parameters of the LSTM model. After each communication round, the training loss, test loss, and test accuracy of each user are recorded and averaged in Fig. 6. The Erdös-Rényi graph have an accuracy of 45.3% which is close to the fully

connected graph's 45.8%. The expander graph reach an accuracy of 40.4% and the Ring graph only reaches 36.2%. In this unfavorable data distribution, the Ring graph generalize worse than the expander graph. The Erdös-Rényi graph has better test accuracy and test loss than the expander graph because it needs significantly more degrees to ensure the connectivity of the graph. Thus it has a higher communication cost. Also, DFedAvgM with the expander graph converges faster than the Ring graph. In this case, we could see that the communication cost for the complete graph is 16 times higher than the expander graph. With some moderate communication cost, expander graph could generalize better and has the similar convergence to the fully connected graph.
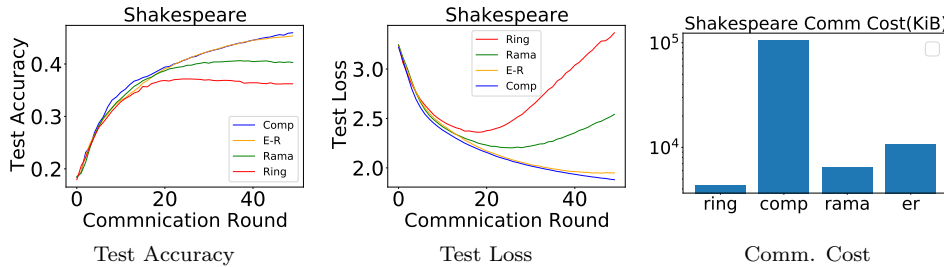


Fig. 6: The test accuracy, test loss, and communication cost of the Ring/3-regular Ramanujan/Erdös-Rényi/Complete graph on non-IID Shakespeare dataset. Erdös-Rényi graph have an accuracy rate of 45.3%. The expander graph reach a accuracy rate of 40.4% and the Ring graph only reaches 36.2%.

**5.2   Robustness to client failures**   To test the robustness of DFedAvgM with different network topology to client failures, we drop 10% and 20% of clients during the communication and compare the performance of Ring, expander, Erdös-Rényi, and fully-connected graphs. In the language modeling, we mask the input of the dropped nodes to simulate the communication failure. All the dropped nodes are randomly selected and excluded from the final results.
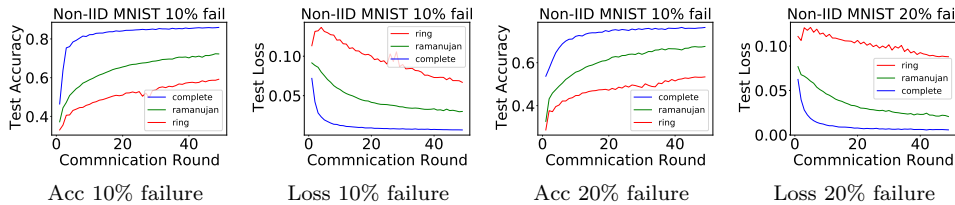


Fig. 7: The test accuracy and test loss of the Ring/3-regular Ramanujan/Complete graphs on non-IID MNIST with client failures.

*MNIST Non-IID.* As shown in Fig. 7, the communication failure not only cause 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. The expander graph reaches 65.3% of accuracy due to its high connectivity and no partition.

*Language Modeling.* In Fig. 8, we have a similar situation as Fig. 7. With the weakest connectivity, the Ring graph degrades to 33.7% accuracy when 20% of the

Acc 10% failure     Loss 10% failure     Acc 20% failure     Loss 20% failure
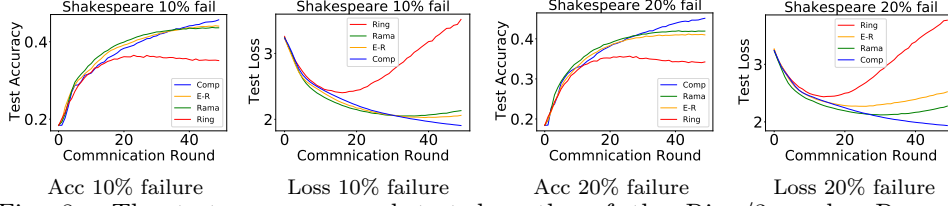
Fig. 8: The test accuracy and test loss the of the Ring/3-regular Ramanujan/Complete/Erdös-Rényi graphs on non-IID MNIST with client failures.

nodes are dropped. The clients are partitioned when multiple nodes fail in a Ring graph. The expander graph reaches 41.5% of accuracy due to its merits. Additionally, although the Erdös-Rényi graph performs slightly better than the expander graph with a 10% client failures, it become worse than the expander graph with a 20% client failure because of its weaker connectivity property.

**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.

### Appendix A. Technical Proofs.

THEOREM A.1 (General nonconvexity [41], (Theorem 2.2 restate)). *Let the sequence $\{\boldsymbol{w}_i^t\}_{t\geq 0}$ be generated by the DFedAvgM for each $i = 1, 2, \ldots, N$, and suppose Assumptions 1-3 hold. Moreover, assume the constant stepsize $\eta$ 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,*

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

*where $T$ is the total number of communication rounds and the constants are given as*

$$\gamma(K,\eta) := \frac{\eta(K-\beta)}{1-\beta} - \frac{64(1-\beta)L^2K^4\eta^3}{K-\beta} - 64LK^2\eta^2$$

$$\alpha(K,\eta) := \frac{\left(\frac{(1-\beta)L^2K^2\eta^3}{K-\beta} + L\eta^2\right)\left(8K\sigma^2 + 32K^2\zeta^2 + \frac{64K^2\beta^2(\sigma^2+B^2)}{(1-\beta)^2}\right)}{\frac{\eta(K-\beta)}{1-\beta} - \frac{64(1-\beta)L^2K^4\eta^3}{K-\beta} - 64LK^2\eta^2}$$

$$\Xi(K,\eta) := \left(\frac{64(1-\beta)L^4K^4\eta^5}{K-\beta} + 64L^3K^2\eta^4\right) \times$$

$$\left(\frac{\left(8K\sigma^2 + 32K^2\zeta^2 + 32K^2B^2 + \frac{64K^2\beta^2}{(1-\beta)^2}(\sigma^2 + B^2)\right)}{\left(\frac{\eta(K-\beta)}{1-\beta} - \frac{64(1-\beta)L^2K^4\eta^3}{K-\beta} - 64LK^2\eta^2\right)}\right)$$

THEOREM A.2 (Uniform stability (Theorem 2.5 restate)). *Under Assumptions 1-4, we have that for any $T$ is the step size $\eta_t \leq \frac{c}{t}$ 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 non-negative 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 denote be $\mathcal{D} = \cup_{i=1}^N \mathcal{D}_i$ 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 $\mathcal{D}_i$ 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

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

and that

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

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

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

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

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

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

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

Then by A.2 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 $W$ satisfies $W\frac{\mathbb{1}}{N} = \frac{\mathbb{1}}{N}$. Then we have with probability $\left(\frac{n-1}{n}\right)^K$ the random variables $\{\Xi^{(t,k)}\}_{k=1}^K = \{\tilde{\Xi}^{(t,k)}\}_{k=1}^K$ are exactly the same:

(A.3)

$$\bar{\boldsymbol{w}}^{t+1} - \bar{\boldsymbol{v}}^{t+1}$$

$$= \boldsymbol{X}^{(t,K)} W \frac{\mathbb{1}}{N} - \boldsymbol{Y}^{(t,K)} W \frac{\mathbb{1}}{N}$$

$$= \left(\boldsymbol{X}^{(t,0)} + \left(\boldsymbol{X}^{(t,K)} - \boldsymbol{X}^{(t,0)}\right)\right) \frac{\mathbb{1}}{N} - \left(\boldsymbol{Y}^{(t,0)} + \left(\boldsymbol{Y}^{(t,K)} - \boldsymbol{Y}^{(t,0)}\right)\right) \frac{\mathbb{1}}{N}$$

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

$$= \left(\left(\boldsymbol{X}^{(t,0)} - \boldsymbol{Y}^{(t,0)}\right)(\boldsymbol{I} - \boldsymbol{P}) + \left(\boldsymbol{X}^{(t,0)} - \boldsymbol{Y}^{(t,0)}\right) \boldsymbol{P}\right) \frac{\mathbb{1}}{N}$$

$$+ \frac{\eta_t}{(1-\theta)^2} \left(\sum_{k=0}^{K-1} p_k(\theta) \left[\boldsymbol{G}^{(t,k)}(\boldsymbol{X}^{(t,k)}; \Xi^{(t,k)}) - \boldsymbol{G}^{(t,k)}(\bar{\boldsymbol{w}}^t \mathbb{1}^T; \Xi^{(t,k)}) + \boldsymbol{G}^{(t,k)}(\bar{\boldsymbol{w}}^t \mathbb{1}^T; \Xi^{(t,k)})\right]\right) \frac{\mathbb{1}}{N}$$

$$- \frac{\eta_t}{(1-\theta)^2} \left(\sum_{k=0}^{K-1} p_k(\theta) \left[\boldsymbol{G}^{(t,k)}(\boldsymbol{Y}^{(t,k)}; \Xi^{(t,k)}) - \boldsymbol{G}^{(t,k)}(\bar{\boldsymbol{v}}^t \mathbb{1}^T; \Xi^{(t,k)}) + \boldsymbol{G}^{(t,k)}(\bar{\boldsymbol{v}}^t \mathbb{1}^T; \Xi^{(t,k)})\right]\right) \frac{\mathbb{1}}{N}$$

$$= \underbrace{\frac{\eta_t}{N(1-\theta)^2} \sum_{i=1}^{N} \sum_{k=0}^{K-1} p_k(\theta) \left[\left(\nabla f_i(\bar{\boldsymbol{w}}^t; \xi_i^{t,k}) - \nabla f_i(\boldsymbol{w}_i^{t,k}; \xi_i^{t,k})\right) - \left(\nabla f_i(\bar{\boldsymbol{v}}^t; \xi_i^{t,k}) - \nabla f_i(\boldsymbol{v}_i^{t,k}; \xi_i^{t,k})\right)\right]}_{=:A_1}$$

(A.4)

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

where we note that $(\boldsymbol{I} - \boldsymbol{P})\frac{1}{N} = 0$. 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

$$\|A_1\|$$

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

$$\leq \frac{\eta_t L}{N} \sum_{i=1}^{N} \sum_{k=0}^{K-1} \|\bar{\boldsymbol{w}}^t - \boldsymbol{w}_i^{t,k}\| + \|\bar{\boldsymbol{v}}^t - \boldsymbol{v}_i^{t,k}\|$$

$$\leq \frac{\eta_t L}{N} \sum_{k=0}^{K-1} \sum_{i=1}^{N} \left( \|\bar{\boldsymbol{w}}^t - \boldsymbol{w}_i^{t,0}\| + \|\bar{\boldsymbol{v}}^t - \boldsymbol{v}_i^{t,0}\| \right)$$

$$+ \frac{\eta_t L}{N} \sum_{k=1}^{K-1} \sum_{i=1}^{N} \left( \|\boldsymbol{w}_i^{t,0} - \boldsymbol{w}_i^{t,k}\| + \|\boldsymbol{v}_i^{t,0} - \boldsymbol{v}_i^{t,k}\| \right)$$

$$\leq \frac{\eta_t L}{\sqrt{N}} \sum_{k=0}^{K-1} \left( \|\boldsymbol{X}^{(t,0)}(\boldsymbol{I} - \boldsymbol{P})\|_F + \|\boldsymbol{Y}^{(t,0)}(\boldsymbol{I} - \boldsymbol{P})\|_F \right)$$

$$+ \frac{\eta_t^2 L}{N(1-\theta)^2} \sum_{i=1}^{N} \sum_{k=1}^{K-1} \sum_{s=0}^{k-1} p_s(\theta) \left( \|\nabla f_i(\boldsymbol{w}_i^{t,s}; \xi_i^{t,s})\| + \|\nabla f_i(\boldsymbol{v}_i^{t,s}; \xi_i^{t,s})\| \right)$$

$$\leq \frac{\eta_t L}{\sqrt{N}} \sum_{k=0}^{K-1} \left( \|\boldsymbol{X}^{(t,0)}(\boldsymbol{I} - \boldsymbol{P})\|_F + \|\boldsymbol{Y}^{(t,0)}(\boldsymbol{I} - \boldsymbol{P})\|_F \right)$$

$$+ \frac{\eta_t^2 L}{N} \sum_{i=1}^{N} \sum_{k=1}^{K-1} \sum_{s=0}^{k-1} \left( \|\nabla f_i(\boldsymbol{w}_i^{t,s}; \xi_i^{t,s})\| + \|\nabla f_i(\boldsymbol{v}_i^{t,s}; \xi_i^{t,s})\| \right).$$

Then, we have

$$\frac{1}{N} \sum_{i=1}^{N} \|\nabla f_i(\boldsymbol{w}_i^{t,s}; \xi_i^{t,s})\| \leq \frac{1}{N} \sum_{i=1}^{N} \|\nabla f_i(\boldsymbol{w}_i^{t,s}; \xi_i^{t,s}) - \nabla f_i(\boldsymbol{w}_i^{t,s})\| + \|\nabla f_i(\boldsymbol{w}_i^{t,s})\|$$

$$\leq \frac{1}{\sqrt{N}} \left( \sum_{i=1}^{N} \|\nabla f_i(\boldsymbol{w}_i^{t,s}; \xi_i^{t,s}) - \nabla f_i(\boldsymbol{w}_i^{t,s})\|^2 \right)^{\frac{1}{2}} + B$$

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

$$\leq \frac{1}{\sqrt{N}} \left( N\sigma^2 \right)^{\frac{1}{2}} + B = \sigma + B,$$

so that with applying the Lemma A.3

$$\|A_1\| \leq 2\eta_t L K(\sigma + B) \left( \sum_{j=1}^{t} \eta_{t-j} \lambda^j \right) + 2\eta_t^2 L(\sigma + B) \sum_{k=1}^{K-1} k$$

$$= 2\eta_t LK(\sigma + B)\left(\sum_{j=1}^{t}\eta_{t-j}\lambda^j\right) + \eta_t^2 L(\sigma + B)K(K-1)$$

$$\leq \eta_t LK(\sigma + B)\left(2\sum_{j=1}^{t-1}\eta_{t-j}\lambda^j + \eta_t K\right).$$

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

$$\left\| \bar{\boldsymbol{w}}^t - \frac{\eta_t}{(1-\theta)^2}\sum_{k=0}^{K-1}p_k(\theta)\nabla f_i(\bar{\boldsymbol{w}}^t;\xi_i^{t,k}) - \left(\bar{\boldsymbol{v}}^t - \frac{\eta_t}{(1-\theta)^2}\sum_{k=0}^{K-1}p_k(\theta)\nabla f_i(\bar{\boldsymbol{v}}^t;\xi_i^{t,k})\right)\right\|$$

$$\leq \|\bar{\boldsymbol{w}}^t - \bar{\boldsymbol{v}}^t\| + \eta_t\sum_{k=0}^{K-1}\left\|\nabla f_i(\bar{\boldsymbol{w}}^t;\xi_i^{t,k}) - \nabla f_i(\bar{\boldsymbol{v}}^t;\xi_i^{t,k})\right\|$$

$$\leq (1 + \eta_t LK)\|\bar{\boldsymbol{w}}^t - \bar{\boldsymbol{v}}^t\|.$$

Plugging everything into (A.3)

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

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,

$$\bar{\boldsymbol{w}}^{t+1} - \bar{\boldsymbol{v}}^{t+1}$$

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

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

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

(A.5)

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

(A.6)

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

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

$$A_2 = \frac{1}{N}\sum_{i=1}^N\left[\bar{\boldsymbol{w}}^t - \bar{\boldsymbol{v}}^t + \frac{\eta_t}{(1-\theta)^2}\sum_{k=0}^{K-1}p_k(\theta)\left(\nabla f_i(\bar{\boldsymbol{w}}^t;\xi_i^{t,k}) - \nabla f_i(\bar{\boldsymbol{v}}^t;\tilde{\xi}_i^{t,k})\right)\right]$$

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

$$+ \frac{\eta_t}{N(1-\theta)^2} \sum_{k=0}^{K-1} p_k(\theta) \left( \nabla f_{i^*}(\bar{\boldsymbol{w}}^t; \xi_{i^*}^{t,k}) - \nabla f_{i^*}(\bar{\boldsymbol{v}}^t; \tilde{\xi}_{i^*}^{t,k}) \right)$$

$$\implies \|A_2\| \leq \|\bar{\boldsymbol{w}}^t - \bar{\boldsymbol{v}}^t\| + \frac{\eta_t LK(m-1)}{N} \|\bar{\boldsymbol{w}}^t - \bar{\boldsymbol{v}}^t\| + \frac{\eta_t}{N} \sum_{k=0}^{K-1} \|\nabla f_{i^*}(\bar{\boldsymbol{w}}^t; \xi_{i^*}^{t,k}) - \nabla f_{i^*}(\bar{\boldsymbol{w}}^t)\|$$

$$+ \frac{\eta_t}{N} \sum_{k=0}^{K-1} \|\nabla f_{i^*}(\bar{\boldsymbol{w}}^t) - \nabla f_{i^*}(\bar{\boldsymbol{v}}^t)\| + \|\nabla f_{i^*}(\bar{\boldsymbol{v}}^t) - \nabla f_{i^*}(\bar{\boldsymbol{v}}^t; \tilde{\xi}_{i^*}^{t,k})\|$$

$$\implies \mathbb{E}\|A_2\| \leq \left( 1 + \frac{\eta_t LK(m-1)}{N} \right) \mathbb{E}\|\bar{\boldsymbol{w}}^t - \bar{\boldsymbol{v}}^t\| + \frac{\eta_t K}{N} \left( 2\sigma + L\mathbb{E}\|\bar{\boldsymbol{w}}^t - \bar{\boldsymbol{v}}^t\| \right)$$

$$= (1 + \eta_t LK) \mathbb{E}\|\bar{\boldsymbol{w}}^t - \bar{\boldsymbol{v}}^t\| + \frac{2\eta_t \sigma K}{N},$$

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.

Recalling the definition $\delta_t := \|\bar{\boldsymbol{w}}^t - \bar{\boldsymbol{v}}^t\|$, then we can combine both cases to obtain

$$\mathbb{E}(\delta_{t+1}|\delta_{t_0} = 0)$$

$$\leq \left( \frac{n-1}{n} \right)^K \left( (1 + \eta_t LK)\mathbb{E}(\delta_t|\delta_{t_0} = 0) + \eta_t LK(\sigma + B) \left( 2\sum_{j=1}^{t-1} \eta_{t-j}\lambda^j + \eta_t K \right) \right)$$

$$+ \left( 1 - \left( \frac{n-1}{n} \right)^K \right) \left( \eta_t LK(\sigma + B) \left( 2\sum_{j=1}^{t-1} \eta_{t-j}\lambda^j + \eta_t(K+1) \right) \right)$$

$$+ \left( 1 - \left( \frac{n-1}{n} \right)^K \right) \left( (1 + \eta_t LK)\mathbb{E}(\delta_t|\delta_{t_0} = 0) + \frac{2\eta_t \sigma K}{N} \right)$$

$$= (1 + \eta_t LK)\mathbb{E}(\delta_t|\delta_{t_0} = 0) + \eta_t LK(\sigma + B) \left( 2\sum_{j=1}^{t-1} \eta_{t-j}\lambda^j + \eta_t K \right)$$

$$+ \left( 1 - \left( \frac{n-1}{n} \right)^K \right) \frac{2\eta_t \sigma K}{N}.$$

With a similar result to bound the sum $\sum_{j=1}^{t-1} \eta_{t-j}\lambda^j$ to that of [41], if we set

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

then we should be able to calculate

$$\mathbb{E}(\delta_{t+1}|\delta_{t_0} = 0) \leq \left( 1 + \frac{cLK}{t} \right) \mathbb{E}(\delta_t|\delta_{t_0} = 0) + \frac{cLK}{t}(\sigma + B) \left( 2\frac{C_\lambda}{t} + \frac{cK}{t} \right)$$

$$+ \left( 1 - \left( \frac{n-1}{n} \right)^K \right) \frac{2c\sigma K}{Nt}$$

$$\leq \left( 1 + \frac{cLK}{t} \right) \mathbb{E}(\delta_t|\delta_{t_0} = 0) + \underbrace{\frac{2cK\sigma}{N}}_{=:C_1} \frac{1}{t} + \underbrace{cLK(\sigma + B)(cK + 2C_\lambda)}_{=:C_2} \frac{1}{t^2}$$

$$\leq \exp\left(\frac{cLK}{t}\right)\mathbb{E}(\delta_t|\delta_{t_0}=0)+\frac{C_1}{t}+\frac{C_2}{t^2}.$$

Unraveling this recursion, we obtain

$$\mathbb{E}(\delta_T|\delta_{t_0}=0)\leq\sum_{t=t_0+1}^{T}\exp\left(cLK\sum_{k=t+1}^{T}\frac{1}{k}\right)\left(\frac{C_1}{t}+\frac{C_2}{t^2}\right)$$

$$\leq\sum_{t=t_0+1}^{T}\exp\left(cLK\ln\frac{T}{t}\right)\left(\frac{C_1}{t}+\frac{C_2}{t^2}\right)$$

$$=T^{cLK}\left(\sum_{t=t_0+1}^{T}\frac{1}{t^{cLK+1}}\left(C_1+\frac{C_2}{t}\right)\right)$$

$$\leq T^{cLK}\left(\frac{C_1}{cLKt_0^{cLK}}+\frac{C_2}{(cLK+1)t_0^{cLK+1}}\right)$$

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

Plugging in the definitions of $C_1, C_2$ we get

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

$$=\left(\frac{T}{t_0}\right)^{cLK}\left[\frac{2\sigma}{NL}+\frac{(\sigma+B)(cK+2C_\lambda)}{t_0}\right],$$

which gives by Lemma A.4

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

$$+B\left(\frac{T}{t_0}\right)^{cLK}\left[\frac{2\sigma}{NL}+\frac{(\sigma+B)(cK+2C_\lambda)}{t_0}\right]$$

$$\leq\frac{t_0K}{n}(\sup f)+\left(\frac{T}{t_0}\right)^{cLK}\left[\frac{2\sigma B}{NL}+\frac{B(\sigma+B)(cK+2C_\lambda)}{t_0}\right].$$

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

$$\mathbb{E}|f(\bar{\boldsymbol{w}}^T;\Xi)-f(\bar{\boldsymbol{v}}^T;\Xi)|\leq\frac{(\sup f)K(cLK)^{\frac{1}{1+cLK}}}{n}T^{\frac{cLK}{1+cLK}}+\frac{\frac{2\sigma B}{NL}}{(cLK)^{\frac{cLK}{1+cLK}}}T^{\frac{cLK}{1+cLK}}$$

$$+\frac{B(\sigma+B)(cK+2C_\lambda)T^{cLK}}{\left((cLK)^{\frac{1}{1+cLK}}T^{\frac{cLK}{1+cLK}}\right)^{cLK+1}}$$

$$=T^{\frac{cLK}{1+cLK}}\left(\frac{(\sup f)K(cLK)^{\frac{1}{1+cLK}}}{n}+\frac{\frac{2\sigma B}{NL}}{(cLK)^{\frac{cLK}{1+cLK}}}\right)$$

$$+ \frac{B(\sigma + B)(cK + 2C_\lambda)}{cLK},$$

as desired.                                                                                    □

LEMMA A.3. *Under Assumptions 1-4 and on the mixing matrix $W$, 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 $\sigma, B$ and $\lambda$ 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

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

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 $k = 0, \dots, K-1$, we have

$$\|\boldsymbol{X}^{(t,0)}(\boldsymbol{I} - \boldsymbol{P})\|_F$$

$$= \left\| \sum_{j=1}^t \frac{\eta_{t-j}}{(1-\theta)^2} \sum_{k=0}^{K-1} p_k(\theta)\boldsymbol{G}^{(t,k)}(\boldsymbol{X}^{(t,k)}; \Xi^{(t,k)}) \left(W^j - P\right) \right\|_F$$

$$\leq \sum_{j=1}^t \eta_{t-j} \left\|W^j - P\right\|_F \sum_{k=0}^{K-1} \left\|\boldsymbol{G}^{(t-j,k)}(\boldsymbol{X}^{(t-j,k)}; \Xi^{(t,k)}) - \nabla\boldsymbol{F}(\boldsymbol{X}^{(t-j,k)}) + \nabla\boldsymbol{F}(\boldsymbol{X}^{(t-j,k)})\right\|_F$$

$$\leq \sum_{j=1}^t \eta_{t-j}\lambda^j \sum_{k=0}^{K-1} \left(\sum_{i=1}^N \|\nabla f_i(\boldsymbol{w}_i^{t-j,k}; \xi_i^{t-j,k}) - \nabla f_i(\boldsymbol{w}_i^{t-j,k})\|^2\right)^{\frac{1}{2}} + \left(\sum_{i=1}^N \|\nabla f_i(\boldsymbol{w}_i^{t-j,k})\|^2\right)^{\frac{1}{2}}$$

$$\leq \sum_{j=1}^t \eta_{t-j}\lambda^j \sum_{k=0}^{K-1} \left[\left(\sum_{i=1}^N \|\nabla f_i(\boldsymbol{w}_i^{t-j,k}; \xi_i^{t-j,k}) - \nabla f_i(\boldsymbol{w}_i^{t-j,k})\|^2\right)^{\frac{1}{2}} + B\sqrt{N}\right],$$

$$\implies \quad \mathbb{E}\|\boldsymbol{X}^{(t,0)}(\boldsymbol{I} - \boldsymbol{P})\|_F$$

$$\leq \sum_{j=1}^t \eta_{t-j}\lambda^j \sum_{k=0}^{K-1} \left[\left(\sum_{i=1}^N \mathbb{E}\|\nabla f_i(\boldsymbol{w}_i^{t-j,k}; \xi_i^{t-j,k}) - \nabla f_i(\boldsymbol{w}_i^{t-j,k})\|^2\right)^{\frac{1}{2}} + B\sqrt{N}\right]$$

$$\leq \sum_{j=1}^t \eta_{t-j}\lambda^j \sum_{k=0}^{K-1} \left[\sigma\sqrt{N} + B\sqrt{N}\right]$$

$$= K\sqrt{N}(\sigma + B) \sum_{j=1}^t \eta_{t-j}\lambda^j,$$

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

LEMMA A.4. *Assume that the loss function $f(\cdot; \Xi)$ is nonnegative and $B$-Lipschitz for all $\Xi$. 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 $\Xi$ and every $t_0 \in \{0, 1, \ldots, n\}$, under the random selection rule, we have

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

*Proof.* Our proof closely follows that of [12], just with a small distinction. After obtaining the inequality

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

where the event $\mathcal{E}$ denotes the event that $\delta_{t_0} = 0$, we similarly need to 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 $\xi^*$, which occurs at node $i^* \in [N]$ and is located in the $j^* \in [n]$ entry of $\tilde{\mathcal{D}}_{i^*}$. We have

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

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 $k$), then we have that

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

from which we conclude the proof. $\qquad\square$

LEMMA A.5. *If $\eta_t \le \frac{c}{t}$ for $t = 1, 2, \ldots$, then*

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

*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.*

(A.7) $$\sum_{j=1}^{t} \eta_{t-j} \lambda^j$$

$$= \sum_{j=1}^{t} \eta_j \lambda^{t-j} \le \lambda^t \sum_{j=1}^{t} \frac{\lambda^{-j}}{j}$$

$$= \lambda^t + \lambda^t \sum_{j=2}^{t} \frac{\lambda^{-j}}{j}$$

(A.8) $$\le \lambda^t + \lambda^t \sum_{j=2}^{t} \int_{j-1}^{j} \frac{\lambda^{-x}}{x} dx$$

(A.9) $$= \lambda^t + \lambda^t \int_{1}^{t} \frac{\lambda^{-x}}{x} dx = \lambda^{t-1} + \lambda^t \left(\int_{1}^{t/2} \frac{\lambda^{-x}}{x} dx + \int_{t/2}^{t} \frac{\exp(x \ln(\frac{1}{\lambda}))}{x} dx\right),$$

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

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

and

$$\int_{t/2}^t \frac{\exp(x \ln(\frac{1}{\lambda}))}{x} dx$$

$$= \int_1^{t/2} \frac{1}{x} \sum_{k=0}^\infty \frac{(\ln \frac{1}{\lambda})^k x^k}{k!} dx$$

$$= \int_1^{t/2} \frac{1}{x} dx + (\ln \frac{1}{\lambda}) \int_1^{t/2} 1 dx + \sum_{k=2}^\infty \frac{(\ln \frac{1}{\lambda})^k}{k!} \int_1^{t/2} x^{k-1} dx$$

$$= 1 - \frac{4}{t^2} + \frac{1}{2}(t-2) \ln \frac{1}{\lambda} + \sum_{k=2}^\infty \frac{(\ln \frac{1}{\lambda})^k}{(k)k!} \left( \left( \frac{t}{2} \right)^k - 1 \right)$$

$$\le 1 - \frac{4}{t^2} + \frac{1}{2}(t-2) \ln \frac{1}{\lambda} + \frac{1}{2} \sum_{k=2}^\infty \frac{(\ln \frac{1}{\lambda})^k}{k!} \left( \left( \frac{t}{2} \right)^k - 1 \right)$$

$$= 1 - \frac{4}{t^2} + \frac{1}{2}(t-2) \ln \frac{1}{\lambda} + \frac{1}{2} \left( \frac{1}{\lambda^{t/2}} - 1 + \frac{t}{2} \ln \frac{1}{\lambda} - \left[ \frac{1}{\lambda} - 1 + \ln \frac{1}{\lambda} \right] \right)$$

$$= 1 - \frac{4}{t^2} + \frac{1}{2}(t-2) \ln \frac{1}{\lambda} + \frac{1}{2} \left( \frac{1}{\lambda^{t/2}} - \frac{1}{\lambda} + \frac{1}{2}(t-2) \ln \frac{1}{\lambda} \right)$$

$$= 1 - \frac{4}{t^2} + \frac{3}{4}(t-2) \ln \frac{1}{\lambda} + \frac{1}{2\lambda^{t/2}} - \frac{1}{2\lambda}.$$

Plugging this result into (A.9), we obtain

$$\sum_{j=1}^t \eta_{t-j} \lambda^j \le \lambda^{t-1} + \lambda^t \left( 1 - \frac{4}{t^2} + \frac{3}{4}(t-2) \ln \frac{1}{\lambda} + \frac{1}{2\lambda^{t/2}} - \frac{1}{2\lambda} + \frac{2\lambda^{-t/2}}{t \ln \frac{1}{\lambda}} \right)$$

$$= \frac{\lambda^{t-1}}{2} + \lambda^t \left( 1 - \frac{4}{t^2} + \frac{3}{4}(t-2) \ln \frac{1}{\lambda} \right) + \lambda^{t/2} \left( \frac{1}{2} + \frac{2}{t \ln \frac{1}{\lambda}} \right)$$

(A.10) $$\le \frac{1}{t} \left[ \lambda^t \left( t + \ln \frac{1}{\lambda} t^2 \right) + \lambda^{t/2} \left( t + \frac{2}{\ln \frac{1}{\lambda}} \right) \right],$$

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

$$t\lambda^t \le \min \left\{ 2\lambda^2, \frac{1}{\ln \frac{1}{\lambda}} \lambda^{\frac{1}{\ln \frac{1}{\lambda}}} \right\},$$

$$t^2 \lambda^t \le \min \left\{ 4\lambda^2, \frac{4}{(\ln \frac{1}{\lambda})^2} \lambda^{\frac{2}{\ln \frac{1}{\lambda}}} \right\},$$

$$t\lambda^{t/2} \le \min \left\{ 2\lambda, \frac{1}{\ln \frac{1}{\lambda}} \lambda^{\frac{1}{\ln \frac{1}{\lambda}}} \right\},$$

where we have noted that each of these functions are decreasing functions of $t$. Therefore, our bound becomes:

$$\sum_{j=1}^{t} \eta_{t-j} \lambda^j$$

$$\leq \frac{1}{t}\left[\min\left\{2\lambda^2, \frac{1}{\ln\frac{1}{\lambda}}\lambda^{\frac{1}{\ln\frac{1}{\lambda}}}\right\} + \min\left\{4\lambda^2 \ln\frac{1}{\lambda}, \frac{4}{\ln\frac{1}{\lambda}}\lambda^{\frac{2}{\ln\frac{1}{\lambda}}}\right\} + \min\left\{2\lambda, \frac{1}{\ln\frac{1}{\lambda}}\lambda^{\frac{1}{\ln\frac{1}{\lambda}}}\right\}\right]$$

$$\qquad + \frac{2}{t\ln\frac{1}{\lambda}}$$

$$= \frac{1}{t}\left[2\lambda^2 + 4\lambda^2\ln\frac{1}{\lambda} + \frac{2}{\ln\frac{1}{\lambda}} + \min\left\{2\lambda, \frac{1}{\ln\frac{1}{\lambda}}\lambda^{\frac{1}{\ln\frac{1}{\lambda}}}\right\}\right]$$

$$=: \frac{C_\lambda}{t}. \qquad\qquad\qquad\qquad \square$$

We note that all terms of $C_\lambda$ 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 $C_\lambda$ 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 International Conference on Trust and Privacy in Digital Business, Springer, 2020, pp. 205–220.

[2] M. F. Balcan, A. Blum, S. Fine, and Y. Mansour, *Distributed learning, communication complexity and privacy*, in Conference on Learning Theory, JMLR Workshop and Conference Proceedings, 2012, pp. 26–1.

[3] 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*, arXiv preprint arXiv:1611.04482, (2016).

[4] S. Boyd, P. Diaconis, and L. Xiao, *Fastest mixing markov chain on a graph*, SIAM review, 46 (2004), pp. 667–689.

[5] O. Choudhury, A. Gkoulalas-Divanis, T. Salonidis, I. Sylla, Y. Park, G. Hsu, and A. Das, *Anonymizing data for privacy-preserving federated learning*, arXiv preprint arXiv:2002.09096, (2020).

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

[7] A. Elisseeff, T. Evgeniou, M. Pontil, and L. P. Kaelbing, *Stability of randomized learning algorithms.*, Journal of Machine Learning Research, 6 (2005).

[8] P. Erdös and A. Rényi, *On the evolution of random graphs*, in PUBLICATION OF THE MATHEMATICAL INSTITUTE OF THE HUNGARIAN ACADEMY OF SCIENCES, 1960, pp. 17–61.

[9] 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 Proc. of IEEE FOCS, 2006.

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

[11] R. C. Geyer, T. Klein, and M. Nabi, *Differentially private federated learning: A client level perspective*, arXiv preprint arXiv:1712.07557, (2017).

[12] M. Hardt, B. Recht, and Y. Singer, *Train faster, generalize better: Stability of stochastic gradient descent*, in International Conference on International Conference on Machine Learning, 2016.

[13] J. HEIDEMANN, M. KLIER, AND F. PROBST, *Online social networks: A survey of a global phenomenon*, Computer Networks, (2012).

[14] 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., vol. 31, Curran Associates, Inc., 2018, https://proceedings.neurips.cc/paper/2018/file/17326d10d511828f6b34fa6d751739e2-Paper.pdf.

[15] 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*, arXiv preprint arXiv:1912.04977, (2019).

[16] 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.

[17] S. S. LAM AND C. QIAN, *Geographic Routing in d-dimensional Spaces with Guaranteed Delivery and Low Stretch*, in Proceedings of ACM SIGMETRICS, 2011.

[18] C. LAW AND K.-Y. SIU, *Distributed construction of random expander networks*, in Proc. of IEEE INFOCOM, 2003.

[19] 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.

[20] X. LI, K. HUANG, W. YANG, S. WANG, AND Z. ZHANG, *On the Convergence of FedAvg on Non-IID Data*, Apr. 2020, https://iclr.cc/virtual_2020/poster_HJxNAnVtDS.html (accessed 2021-04-22).

[21] Z. LIANG, B. WANG, Q. GU, S. OSHER, AND Y. YAO, *Exploring private federated learning with Laplacian smoothing*, arXiv preprint arXiv:2005.00218, (2020).

[22] D. LIU AND O. SIMEONE, *Privacy for free: Wireless federated learning via uncoded transmission with adaptive power control*, IEEE Journal on Selected Areas in Communications, 39 (2020), pp. 170–185.

[23] 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, 2020, pp. 485–501.

[24] Z. LIU, T. LI, V. SMITH, AND V. SEKAR, *Enhancing the privacy of federated learning with sketching*, arXiv preprint arXiv:1911.01812, (2019).

[25] E. K. LUA, J. CROWCROFT, M. PIAS, R. SHARMA, AND S. LIM, *A survey and comparison of peer-to-peer overlay network schemes*, IEEE Communications Surveys and Tutorials, (2005).

[26] O. MARFOQ, C. XU, G. NEGLIA, AND R. VIDAL, *Throughput-optimal topology design for cross-silo federated learning*, in 34th Conference on Neural Information Processing Systems (NeurIPS 2020), 2020.

[27] 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.

[28] 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, A. Singh and J. Zhu, eds., vol. 54 of Proceedings of Machine Learning Research, Fort Lauderdale, FL, USA, 20–22 Apr 2017, PMLR, pp. 1273–1282, http://proceedings.mlr.press/v54/mcmahan17a.html.

[29] 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, 2019, pp. 691–706.

[30] A. NEDIC AND A. OZDAGLAR, *Distributed subgradient methods for multi-agent optimization*, IEEE Transactions on Automatic Control, 54 (2009), pp. 48–61.

[31] M. A. OGLEARI, Y. YU, C. QIAN, E. MILLER, AND J. ZHAO, *String figure: A scalable and elastic memory network architecture*, in Proc. of IEEE HPCA, 2019.

[32] T. OREKONDY, S. J. OH, Y. ZHANG, B. SCHIELE, AND M. FRITZ, *Gradient-leaks: Understanding and controlling deanonymization in federated learning*, arXiv preprint arXiv:1805.05838, (2018).

[33] R. PATHAK AND M. J. WAINWRIGHT, *Fedsplit: An algorithmic framework for fast federated optimization*, arXiv preprint arXiv:2005.05238, (2020).

[34] D. POVEY, X. ZHANG, AND S. KHUDANPUR, *Parallel training of dnns with natural gradient and parameter averaging*, arXiv preprint arXiv:1410.7455, (2014).

[35] C. QIAN AND S. LAM, *A Scalable and Resilient Layer-2 Network with Ethernet Compatibility*,

IEEE/ACM Transactions on Networking, (2016).

[36] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, *Adaptive federated optimization*, arXiv preprint arXiv:2003.00295, (2020).

[37] O. Shamir and N. Srebro, *Distributed stochastic optimization and learning*, in 2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), IEEE, 2014, pp. 850–857.

[38] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, *Jellyfish: Networking data centers randomly*, in Proc. of USENIX NSDI, 2012.

[39] S. U. Stich and S. P. Karimireddy, *The Error-Feedback framework: SGD with Delayed Gradients*, Journal of Machine Learning Research, 21 (2020), pp. 1–36, http://jmlr.org/papers/v21/19-748.html (accessed 2021-04-22).

[40] T. Sun, D. Li, , and B. Wang, *Stability and generalization of the decentralized stochastic gradient descent*, in Thirty-Fourth AAAI Conference on Artificial Intelligence, 2021.

[41] T. Sun, D. Li, and B. Wang, *Decentralized federated averaging*, arXiv preprint arXiv:2104.11375, (2021), https://arxiv.org/abs/2104.11375.

[42] A. Triastcyn and B. Faltings, *Federated learning with bayesian differential privacy*, in 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 2587–2596.

[43] 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, 2019, pp. 1–11.

[44] 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 Third ACM International Workshop on Edge Systems, Analytics and Networking, 2020, pp. 61–66.

[45] J. Wang and G. Joshi, *Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms*, arXiv preprint arXiv:1808.07576, (2018).

[46] J. Wang, A. K. Sahu, Z. Yang, G. Joshi, and S. Kar, *Matcha: Speeding up decentralized sgd via matching decomposition sampling*, in 2019 Sixth Indian Control Conference (ICC), IEEE, 2019, pp. 299–300.

[47] 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 Transactions on Information Forensics and Security, 15 (2020), pp. 3454–3469.

[48] 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, 2019, pp. 13–23.

[49] 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., vol. 97 of Proceedings of Machine Learning Research, PMLR, 09–15 Jun 2019, pp. 7184–7193.

[50] Y. Yu and C. Qian, *Space Shuffle: A Scalable, Flexible, and High-Bandwidth Data Center Network*, IEEE Transactions on Parallel and Distributed Systems, (2016).

[51] S. Zhang, A. Choromanska, and Y. LeCun, *Deep learning with elastic averaging sgd*, arXiv preprint arXiv:1412.6651, (2014).

[52] 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*, arXiv preprint arXiv:2005.11418, (2020).