Energy-Efficient Connectivity-Aware Learning Over Time-Varying D2D Networks

Rohit Parasnis, Seyyedali Hosseinalipour, *Member, IEEE*, Yun-Wei Chu, Mung Chiang, and Christopher G. Brinton, *Senior Member, IEEE*

***Abstract*—Semi-decentralized federated learning blends the conventional device-to-server (D2S) interaction structure of federated model training with localized device-to-device (D2D) communications. We study this architecture over edge networks with multiple D2D clusters modeled as time-varying and directed communication graphs. Our investigation results in two algorithms: (a) a *connectivity-aware* learning algorithm that controls the fundamental trade-off between the convergence rate of the model training process and the number of energy-intensive D2S transmissions required for global aggregation, and (b) a *motion-planning* algorithm to enhance the densities and regularity levels of cluster digraphs so as to further reduce the number of D2S transmissions in connectivity-aware learning. Specifically, in our semi-decentralized methodology, weighted-averaging-based D2D updates are injected into the federated averaging framework based on column-stochastic weight matrices that encapsulate the connectivity within the clusters. To develop our algorithm, we show how the current expected optimality gap (i.e., the distance between the most recent global model computed by the server and the target/desired optimal model) depends on the greatest two singular values of the weighted adjacency matrices (and hence on the densities and degrees of digraph regularity) of the D2D clusters. We then derive tight bounds on these singular values in terms of the node degrees of the D2D clusters, and we use the resulting expressions to design our connectivity-aware learning algorithm. Simulations performed using real-world datasets and Random Direction Mobility Model (RDMM)-based time-varying D2D topologies reveal that our connectivity-aware algorithm significantly reduces the total communication energy required to reach a target accuracy level compared with baselines while achieving the accuracy level in nearly the same number of iterations as these baselines.**

***Index Terms*—Connectivity, semi-decentralized, federated learning, energy efficiency.**

Manuscript received 2 July 2023; revised 4 December 2023; accepted 17 February 2024. Date of publication 11 March 2024; date of current version 3 July 2024. This work was supported in part by ONR under Grant N000142112472 and Grant N000142212305, in part by NSF under Grant CNS-2146171 and Grant CNS-2212565, and in part by DARPA under Grant D22AP00168-00. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Shuvra S. Bhattacharyya. An earlier version of this paper was presented in part at the 2023 24th International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing [DOI: 10.1145/3565287.3610278]. *(Corresponding author: Rohit Parasnis.)*

Rohit Parasnis, Yun-Wei Chu, Mung Chiang, and Christopher G. Brinton are with Purdue University, West Lafayette, IN 47907 USA (e-mail: rohit100@mit.edu; chu198@purdue.edu; chiang@purdue.edu; cgb@purdue.edu).

Seyyedali Hosseinalipour is with the University at Buffalo–SUNY, Buffalo, NY 14260 USA (e-mail: alipour@buffalo.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/JSTSP.2024.3374591, provided by the authors.

Digital Object Identifier 10.1109/JSTSP.2024.3374591

I. Introduction

FEDERATED learning (FL) [2], [3] is a popular paradigm for distributing machine learning (ML) tasks over a network of centrally coordinated devices. By not requiring the devices to share any training data with the central coordinator (server), FL improves privacy and communication efficiency. The first FL technique, known as federated averaging (FedAvg), was proposed in [2], [3] as a distributed optimization algorithm for a "star" topology-based network architecture. In each iteration of the FedAvg algorithm, (i) devices individually perform a number of local stochastic gradient descent (SGD) iterations and transmit their cumulative stochastic gradients to the central server, which then (ii) aggregates a random subset of these gradients to estimate the globally optimal ML model. In recent years, several variants of FedAvg have been proposed to address the challenges encountered by FL at the wireless edge, including different dimensions of heterogeneity in dataset statistics (e.g., varying local data distributions) and in the network system itself (e.g., varying communication and computation capabilities).

An emerging arch of work has been exploring FL under edge networks that diverge from the star learning topology between the devices and the server. This had led to varying degrees of decentralization in FL, reaching fully decentralized, serverless settings that sit at the opposite extreme of the star topology [4], [5], [6], [7], [8], [9], [10]. In between these two extremes is *semi-decentralized FL*, where device-to-device (D2D) communications complement device-to-server (D2S) interactions [11], [12], [13], [14]. These D2D interactions occur locally within *clusters* of devices, with each cluster forming a connected component. In semi-decentralized FL, D2D transmissions are less energy-consuming than D2S interactions and can help reduce the frequency of D2S communications through localized synchronizations of the ML model updates.

Despite these recent investigations, we still do not have a clear understanding of how different D2D topology properties impact the learning process. For instance, the ratio of the number of D2D interactions to that of D2S interactions will impact the training efficiency (measured in terms of either the training speed or the maximum achievable training accuracy for a given energy budget) differently over different topologies. This becomes especially important in the presence of constraints such as upload/download bandwidths, and stochastic uncertainties such as data heterogeneity, client mobility, and communication link failures. On one hand, edge devices in clustered D2D networks

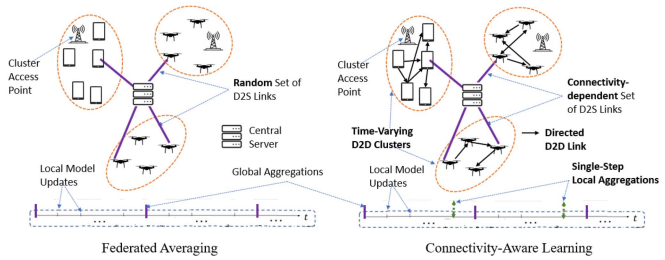

Fig. 1. Conventional federated learning vs. connectivity-aware semi-decentralized learning architecture.

that have little to no cross-cluster interactions are typically in contact with only a small fraction of the rest of the network at any given time instant (e.g., networks of unmanned aerial vehicle (UAV) swarms spread over geo-distributed regions separated by long distances). In such networks, if there is no central coordinator (implying zero D2S interactions) and if the training data are distributed heteregeneously among the edge devices, no practically feasible number of D2D interactions is likely to aggregate a set of local ML models that are diverse enough to approximate the global data distribution [15].

On the other hand, having a high number of D2D interactions is advantageous when D2S interactions take the form of high-energy, high-latency transmissions (e.g., if the UAV swarms in the previous example are miles away from the nearest base station). Moreover, classical star-topology-based FL architectures miss out on an important benefit of D2D cooperation: devices acting as information relays between other devices and the server, effectively sharing with the server more information than it would expect to receive.

We are thus motivated to conduct a formal study of semi-decentralized FL, and reveal the combined impact of D2S and D2D interactions on the training process. After building an understanding of the D2D topologies on which the D2D interactions occur, we propose a novel FL technique that enables us to take into account the degree distributions of the D2D clusters and use this knowledge to tune the number of energy-intensive D2S transmissions while simultaneously ensuring a minimum rate of global training convergence. As shown in Fig. 1, we incorporate two scales of model aggregations: on the first scale, the edge devices perform intra-cluster model aggregations with their one-hop neighbors via distributed averaging, and on the second scale, a central server samples a random set of clients (as in the classical FedAvg architecture [3]) for global aggregation. In our theoretical analysis of these model aggregations, our focus will be on the *expected optimality gap*, which will be defined formally in Lemma 2 and then used throughout the paper. Informally, the expected optimality gap is a quantity that captures on average the extent to which the global model computed by the central server differs from the target model (or the desired model that minimizes the global loss function) after a given number of iterations. Intuitively, therefore, the expected optimality gap quantifies the distance that the global model should traverse to reach optimality.

Our methodology has several potential use-cases, including the following that we will refer to as examples henceforth:

1) UAV Networks for ISR: UAVs are being increasingly deployed for intelligence, surveillance, and reconnaissance (ISR) operations in defense settings [16], [17]. With UAVs partitioned into D2D-enabled swarms deployed across different areas, our connectivity-aware algorithm can facilitate energy-efficient intra-cluster communications and reduce the over-reliance of the model training process on D2S transmissions.

2) Self-Driving Cars: Many learning tasks for self-driving cars call for vehicles to communicate over short distances. In such settings, geographical proximity can be used to partition the traffic network into clusters, enabling us to design intra-cluster D2S communications that turn out to be more energy-efficient than D2S communications with a far-away server.

A. Summary of Contributions

We summarize our key contributions below:

1) Analysis With Time-Varying and Directed Cluster Topologies: Our model is general in that each D2D cluster is assumed to be a time-varying directed graph (digraph). We show how the expected optimality gap of the learning process depends on the greatest two singular values of the weighted adjacency matrices used for local aggregations in the clusters. Our analysis is applicable to edge networks with asymmetric/unidirectional D2D communications subjected to link failures.

2) Singular Value Bounds in Terms of Node Degrees: We derive bounds on the singular values of the cluster-specific weighted adjacency matrices in terms of the degree distribution of every cluster. This introduces new technical challenges as described in Section I-B, since it is a stark departure from existing analyses of averaging-based FL algorithms that rely heavily on the spectral gaps of symmetric weight matrices (e.g., see [6], [11], [13], [18], [19], [20], [21]).

3) Connectivity-Aware Learning Algorithm: We use our singular value bounds to design a time-varying threshold on the number of clients required to be sampled by the central server for global aggregation so as to enforce a desired convergence rate while simultaneously reducing the number of D2S communications. This tradeoff results in a novel connectivity-aware algorithm with significant energy savings, as validated subsequently by our numerical results.

4) Motion-Planning Algorithm: We develop a motion-planning algorithm that enhances the regularity properties of the cluster digraphs, thereby enabling our connectivity-aware learning algorithm to further reduce the total energy consumption by further reducing the number of D2S transmissions, as can be seen from our latest set of simulation results.

5) Effect of Data Heterogeneity Under Mild Gradient Diversity Assumptions: We derive a bound on the expected optimality gap that captures the effects of cluster densities as well as the extent of data heterogeneity across the devices. In doing so, we employ a milder definition of gradient diversity [11] than what is typically assumed in literature.

B. Related Work

Several different FL approaches with varying levels of energy efficiency and different degrees of decentralization have been proposed to date. In this section, we focus on those which are most relevant to the present work.

Semi-decentralized FL: The closely related paper [11] also proposes a semi-decentralized learning methodology for clustered D2D networks. The key differences between [11] and the present work are (a) we do not assume the D2D communications to be bidirectional (equivalently, the cluster graphs in our model are not undirected), and (b) our analysis uses column-stochastic weight matrices that need not satisfy the standard but unrealistic assumptions of symmetry or double stochasticity (which may not hold if the cluster graphs are directed). This leads to two significant technical challenges. First, we cannot use standard eigenvalue results in our analysis since we must focus on singular values, which generally differ from eigenvalues for asymmetric matrices. Second, unlike doubly stochastic matrices, column-stochastic aggregation matrices in general do not ensure convergence to consensus in the absence of a central coordinator, which means our analysis must (and does) account for the combined effect of global aggregations and column-stochasticity.

Another closely related semi-decentralized learning methodology is [12], in which the goal is to enable edge devices to compute weighted sums of their neighbors' scaled cumulative gradients in order to reduce the dependence of the global training process on unreliable D2S links. [12], however, assumes the D2D communication network to be time-invariant and undirected, thereby disregarding potential communication link failures and client mobility.

Energy-Efficient FL: [22] proposes a computationally efficient iterative algorithm to minimize the overall energy consumption in IOT-based FL architectures by simultaneously optimizing a combination of objectives such as communication frequency, learning accuracy, and bandwidth allocation. A related work, [23], proposes a reinforcement learning-based algorithm called AutoFL, which jointly optimizes the convergence time and the energy efficiency by allowing only selected subsets of devices to participate in any given training round. In the context of FL over heterogeneous mobile networks, [24] develops an algorithm that sparsifies local gradients to varying extents depending on the total energy budget. Related works include [25], which explores energy-efficiency from the viewpoint of communication-computation trade-offs in 5G+ networks, [26], which uses Intelligent Reflecting Surfaces (IRS) to maximize resource utilization in wireless networks, and others such as [27], [28], [29], [30]. See [31] for a related survey. However, none of these works, unlike ours, consider time-varying D2D networks.

Learning over Clustered D2D Networks: Recently, [32] proposed fully decentralized (serverless) learning over a static D2D network (unlike our dynamic D2D topologies) equipped with bridge nodes to enable cross-cluster communications. Reference [14] also focuses on clustered networks, but it provides a semi-decentralized learning methodology where the basis for clustering is data similarity, whereas our methodology makes no such assumptions. Another relevant work, [33], proposes having one edge server per cluster so as to eliminate the need for a central server. Its learning algorithm assumes the edge network topology to be undirected, which gives rise to a symmetric adjacency matrix.

Other Averaging-based Algorithms: We remark that there exists abundant literature on distributed optimization over time-varying digraphs characterized by weight matrices that are not necessarily doubly stochastic (see [34], [35], [36], [37], [38], [39], [40]). However, the effects of both data heterogeneity and degree distributions of the studied communication digraphs on the convergence rates of these algorithms have remained largely unexplored.

II. Preliminaries: Singular Values of Digraph Adjacency Matrices

Before introducing the semi-decentralized learning setup, we explain the significance of an essential component of our proposed algorithm: the greatest two singular values of weighted adjacency matrices of digraphs.

Consider a network of n devices aiming to estimate the value of an unknown quantity of interest q (which, as we shall see, takes the form of gradients of the global loss function throughout this paper). The network forms a directed graph $G = (V, E)$, where V denotes the set of network nodes/devices and $E \subset V \times V$ denote the set of edges (i.e., the set of communication links). Suppose each device $i \in \{1, 2, \ldots, n\}$ stores a local estimate x_i of q and updates this estimate by computing a weighted sum of its own estimate as well as the estimates of its *in-neighbors*, i.e., devices it can receive information from. To model this distributed computation, we define the $n \times n$ *weighted adjacency matrix* A of G, where a_{ij} in the i-th row and the j-th column of A denotes the weight assigned by device i to the estimate of device j in its local weighted sum computation. $a_{ij} \neq 0$ only if $(i, j) \in E$.

The set of updated estimates of q is given by

$$x(k+1) = Ax(k), \tag{1}$$

where $x(k+1)$ (respectively, $x(k)$) is a vector whose i-th entry denotes the updated estimate (respectively, current estimate) of device i for each $i \in \{1, 2, \ldots, n\}$ over iterations $k \geq 0$. Hence, the computation of the local estimates $\{x_i(k)\}_{i=1}^n$ of the devices is governed by the structure of A, which in turn depends on G.

We now examine this connection between G, A and the distributed computation procedure (1) when the goal of the devices is to perform repeated weighted summations to let their local estimates converge to $\frac{1}{n}\sum_{i=1}^n x_i(0)$, the average of all the initial estimates. This goal imposes two requirements:

i) *(Average Preservation):* The average of all the local estimates should be preserved at each iteration, i.e., $\frac{1}{n}\sum_{i=1}^n x_i(k) = \frac{1}{n}\sum_{i=1}^n x_i(0)$ for each k.
ii) *(Consensus):* Every local estimate should converge to the same limit, i.e., there should exist a real number c such that $x_i(k) \to c$ as $k \to \infty$ for all $i \in \{1, 2, \ldots, n\}$. In other words, all the devices should reach a consensus on their estimate of q.

It is well-known in the literature on distributed algorithms for averaging and optimization [41] that (i) requires A to be *column-stochastic* (i.e., the entries in every column of A should sum to 1), while (ii) requires A to be *row-stochastic* (i.e., the entries in every row should sum to 1). In effect, A is required to be *doubly stochastic*, i.e., both row-stochastic and column-stochastic.

While there exist standard techniques to construct doubly stochastic adjacency matrices for *undirected graphs*, the existing algorithms for distributed construction of such matrices for *directed graphs* [42], [43], [44], [45], [46], [47] are known to either be computationally expensive or have poor convergence rates. This motivates us to consider choices for A that are either column-stochastic or row-stochastic, and control the impact of violations of double-stochasticity that manifest as a result.

To this end, for a given column-stochastic matrix A, we define its *average deviation from double stochasticity* $\delta_{\text{DS}}(A)$ as the average difference of the row sums from 1, i.e., $\delta_{\text{DS}}(A) := \frac{1}{n}\sum_{i=1}^{n} |\sum_{j=1}^{n} a_{ij} - 1|$. If A is row-stochastic, then we use its transpose as $\delta_{\text{DS}}(A) = \delta_{\text{DS}}(A^\top)$. Similarly, we define the *maximum deviation from double stochasticity* as $\Delta_{\text{DS}}(A) = \max_{i\in[n]} |\sum_{j=1}^{n} a_{ij} - 1|$ if A is column-stochastic and $\Delta_{\text{DS}}(A) = \Delta_{\text{DS}}(A^\top)$ if A is row-stochastic.

The greater the deviation of A from double stochasticity, the greater the extent to which (i) or (ii) is violated. We next show how the largest singular value σ_1 of A captures this deviation.

A. Interpreting the Largest Singular Value

We first note that $\sigma_1^2 - 1$, $\delta_{\text{DS}}(A)$, and $\Delta_{\text{DS}}(A)$ are all zero if A is a doubly stochastic matrix: $\delta_{\text{DS}}(A) = 0$ and $\Delta_{\text{DS}}(A) = 0$ follow from the definition of row-stochasticity, whereas $\sigma_1^2 - 1 = 0$ is a known result [48, Problem 8.7.P5].

On the other hand, if A is not doubly stochastic, then we have the following bounds on $\sigma_1^2 - 1$:

Lemma 1: Let A be either a row-stochastic or a column-stochastic matrix. Then $0 \le \sigma_1^2 - 1 \le \Delta_{\text{DS}}(A)$.

Proof: Let ρ denote the spectral radius of A. Then the lower bound is obtained as follows:

$$\sigma_1^2 \overset{(a)}{=} \|A\|_2^2 \overset{(b)}{\ge} \rho^2 \overset{(c)}{=} 1,$$

where (a) follows from [49, Problem 5.12.2], (b) follows from [49, Example 7.1.4], and (c) follows since every row-stochastic or column-stochastic matrix has $\rho = 1$.

For the upper bound, suppose first that A is column-stochastic. Then we have

$$\sigma_1^2 - 1 \overset{(a)}{\le} \max_{i\in n} \sum_{j=1}^{n} (AA^\top)_{ij} - 1 = \max_{i\in[n]} \sum_{j=1}^{n}\sum_{k=1}^{n} a_{ik}a_{jk} - 1$$

$$= \max_{i\in[n]} \sum_{k=1}^{n} a_{ik} \left(\sum_{j=1}^{n} a_{jk} \right) - 1 \overset{(b)}{=} \max_{i\in[n]} \sum_{k=1}^{n} a_{ik} - 1,$$

where (a) follows from [48, Theorem 8.1.22] and the fact that σ_1^2 is the spectral radius of $AA^\top$, and (b) follows from the column-stochasticity of A. Noting that $\max_{i\in[n]} \sum_{k=1}^{n} a_{ik} - 1 \le \max_{i\in[n]} |\sum_{k=1}^{n} a_{ik} - 1|$ yields the desired bound. □

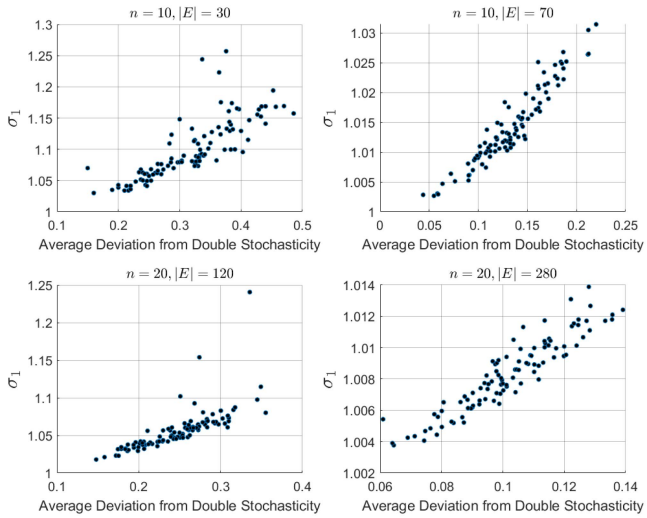

Fig. 2. Correlation between σ_1, the greatest singular value of A, and $\delta_{\text{DS}}(A)$, the average deviation from double stochasticity for different values of n and $|E|$. Each subplot displays 100 points, each of which corresponds to the *equal-neighbor* weighted adjacency matrix (see Section IV) of a random digraph, which we generate by sampling from the uniformly on n and $|E|$.

Thus, a smaller maximum deviation from row-stochasticity makes it less likely that σ_1 deviates significantly from 1. This means that for a fixed number of edges or communication links in G, a weight assignment that brings A closer to double stochasticity will tend to result in a smaller value of σ_1, as the example below shows.

Example 1: Consider the following column-stochastic matrices, and let their greatest singular values be σ_1 and σ_1'.

$$A = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{2} \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{2} \end{pmatrix}, \quad A' = \begin{pmatrix} \frac{1}{3} & \frac{1}{4} & 0 & 0 \\ \frac{1}{3} & \frac{5}{12} & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{4} \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{3}{4} \end{pmatrix}$$

Note that both A and A' have the same zero pattern. Hence, they are both candidate adjacency matrices for the same digraph. However $\Delta_{\text{DS}}(A) = \frac{1}{3} < \frac{5}{12} = \Delta_{\text{DS}}(A')$ and $\delta_{\text{DS}}(A) = \frac{1}{6} < \frac{1}{4} = \delta_{\text{DS}}(A')$, which means A is closer to being doubly stochastic than A'. This is consistent with $\sigma_1 = 1.03 < 1.06 = \sigma_1'$, i.e., the greatest singular value of A deviates less from 1 than the greatest singular value of A'.

Therefore, given a communication digraph G, the greatest singular value of its weighted adjacency matrix A provides an estimate of the extent to which the distributed computation protocol described by (1) deviates from its end goals of average preservation and consensus. Note, however, that the dependence of σ_1 on the structure of A is complex in general and cannot be captured fully by any measure of deviation from double stochasticity. Nevertheless, there is a significant correlation between σ_1 and $\delta_{\text{DS}}(A)$ for a class of weighted adjacency matrices called *equal-neighbor* adjacency matrices, as plotted in Fig. 2 for several randomly generated digraphs. We will discuss these matrices further in Section IV.

B. Interpreting the Second-Largest Singular Value

We now focus on σ_2. Note that σ_2^2 is by definition the second-largest *eigenvalue* of $A^\top A$. Furthermore, by symmetry of $A^\top A$, there exists an undirected graph $G^{(2)}$ that has $A^\top A$ as its weighted adjacency matrix.

These observations enable us to use the literature on weighted Cheeger's inequality (e.g., see [50]) to relate σ_2 to the concept of *isoperimetric numbers*, quantities that capture the severity of "bottlenecks" in the flow of information over networks. In essence, a large isoperimetric number means that every two-set partitioning of the network nodes has many links connecting the two subsets together. Therefore, in the case of approximately *regular graphs,*[1] i.e., graphs in which node degrees are approximately homogeneous across the network, we would expect the isoperimetric number to often increase with the total number of links in the graph. In the case of $G^{(2)}$, this would likely result in a decrease in the eigenvalue σ_2^2 of its weighted adjacency matrix $A^\top A$, because the greater the isoperimetric number, the lower the upper bound provided by the weighted Cheeger's inequality on the second-largest eigenvalue.

Now, the number of edges in $G^{(2)}$ (which equals half the number of non-zero entries in $A^\top A$) is non-decreasing in the number of links in the original graph G (which equals the number of non-zero entries in A). Along with the preceding discussion, this suggests that increasing the number of links in an approximately regular digraph G tends to increase the isoperimetric number of the corresponding augmented graph $G^{(2)}$, and in turn decrease the value of σ_2. Thus, σ_2 is an estimate of the lack of connectivity in G. This is consistent with Fig. 3, which exhibits a negative correlation between σ_2 and the number of links $|E|$ for several randomly generated graphs of different sizes.

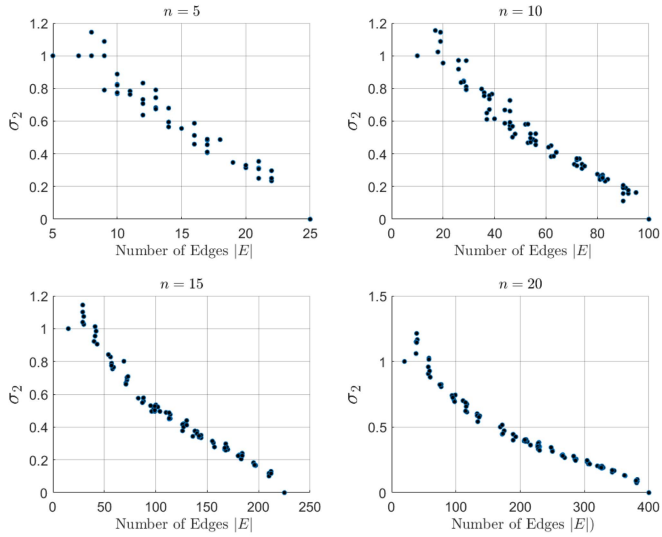

Fig. 3. Correlation between σ_2, the second greatest singular value of A, and $|E|$, for different values of n. Each subplot displays 100 points, each of which corresponds to the *equal-neighbor* weighted adjacency matrix of a digraph. These digraphs are generated by deleting up to one randomly selected edge per node from an n-vertex random regular digraph whose maximum out-degree is distributed uniformly on $\{1, 2, \ldots, n\}$.

III. Semi-Decentralized FL Setup

We now introduce the system model, the learning objective, and the network model in semi-decentralized FL.

A. System Model and Learning Objectives

We consider a collaborative learning environment consisting of n edge devices, or *clients*, and a central parameter server (PS) that is tasked with aggregating all the local model updates generated by the clients. We use $[n] := \{1, 2, \ldots, n\}$ to denote the set of clients.

Each client $i \in [n]$ has a local dataset $\mathcal{D}_i$, which is a collection of data samples of the form $\xi = (u, y)$ where $u \in \mathbb{R}^p$ is the *feature vector* of the sample and y is its *label*. On this basis, for any model $x \in \mathbb{R}^p$, we define the *loss function* $L : \mathbb{R}^p \times \cup_{i=1}^n \mathcal{D}_i \to \mathbb{R}$ so that $L(x; \xi)$ denotes the loss incurred by x on a sample $\xi \in \cup_{i=1}^n \mathcal{D}_i$ (where $\cup_{i=1}^n \mathcal{D}_i$ is the global dataset). The average loss incurred by x over the local dataset of client i is given by $f_i(x) := \frac{1}{|\mathcal{D}_i|} \sum_{\xi \in \mathcal{D}_i} L(x; \xi)$, where $f_i : \mathbb{R}^p \to \mathbb{R}$ denotes the *local loss function* of client i.

[1]We will show results around approximately regular digraphs in Section VI.

In collaboration with the PS, the clients seek to minimize the *global loss function* $f : \mathbb{R}^p \to \mathbb{R}$, defined as the unweighted arithmetic mean $f(x) := \frac{1}{n} \sum_{i=1}^n f_i(x)$ of all the local loss functions. The learning objective, therefore, is to determine the *global optimum* $x^* := \arg\min_{x \in \mathbb{R}^p} f(x)$.

B. D2D and D2S Network Models

We model two types of interactions among the network elements: (i) D2S and (ii) D2D. For D2S interactions, the devices can engage in uplink communications to the PS if prompted by the server, which happens through a sampling procedure explained later.

We model the D2D network as a time-varying directed graph $G(t) = ([n], E(t))$, where $[n]$ denotes the *vertex set* and $E(t)$ the *edge set* of the digraph. The existence of a directed edge from a node $i \in [n]$ to another node $j \in [n]$ in $G(t)$ denotes the existence of a communication link from the i-th client to the j-th client in the D2D network. In this case, we refer to client i (respectively, client j) as the in-neighbor (respectively, out-neighbor) of client j (respectively, client i). The set of in-neighbors (respectively, out-neighbors) of a client $i \in [n]$ at time t is denoted by $\mathcal{N}_i^-(t)$ (respectively, $\mathcal{N}_i^+(t)$). The number of in-neighbors (respectively, out-neighbors) is called the in-degree (respectively, out-degree) and is denoted by $d_i^-(t)$ (respectively, $d_i^+(t)$). We let $d_{\max}^-(t)$, $d_{\min}^+(t)$, and $d_{\max}(t)$ denote the maximum in-degree, the minimum out-degree, and the maximum out-degree, respectively.

Unlike standard works on distributed learning [36], [37], [38], [39], we do not assume the D2D network to be strongly connected or even periodically strongly connected [38], [39] over time. This gives rise to a number $c > 1$ of strongly connected components of $G(t)$, denoted

$\{(V_1(t), E_1(t)), (V_2(t), E_2(t)), \ldots, (V_c(t), E_c(t))\}$ which we refer to as *clusters* of the D2D network. Here, we make the following mild assumptions that apply to most cellular networks:

1) There does not exist any communication link between any two clusters. In other words, $E(t) = \cup_{\ell=1}^{c} E_\ell(t)$.
2) Regardless of any client movement from one cluster to another, as of time t, the server has full knowledge of the vertex sets $\{V_\ell(t)\}_{\ell=1}^{c}$ of all the c clusters.

The second condition is satisfied in cellular communications since the base station knows the users in its coverage area.

IV. Proposed Method for Connectivity-Aware Learning

We now present our methodology for connectivity-aware learning over the semi-decentralized setup from Section II. Our technique will enable the central server to use limited knowledge of the cluster degree distributions to tune a communication-efficiency trade-off.

A. Local Model Updates

As in many FL schemes, we assume every client performs multiple rounds of local SGD iterations between any two consecutive rounds of global aggregation. Let $x^{(t)}$ denote the global model that all the clients possess at the end of the t-th round of global aggregation. Then, each client $i \in [n]$ performs $T \in \mathbb{N}$ iterations of local SGD. In other words, for each $k \in \{0, 1, \ldots, T-1\}$, we have

$$x_i^{(t,k+1)} = x_i^{(t,k)} - \eta_t \widetilde{\nabla} f_i(x_i^{(t,k)}), \tag{2}$$

where $\eta_t > 0$ is the learning rate or the step-size, and $\widetilde{\nabla} f_i(x) := \frac{1}{|\chi_i|} \sum_{\xi \in \chi_i} \nabla L(x; \xi)$ is the stochastic gradient computed by client i by sampling a *mini-batch* or a random subset $\chi_i \subset \mathcal{D}_i$ of its local samples. Note that $x_i^{(t,0)} := x^{(t)}$.

B. Intra-Cluster Model Aggregations

The next step involves all the clients aggregating their scaled cumulative gradients with their neighbors. This aggregation takes the form of weighted sum computations similar to (1). Every client $i \in [n]$ first transmits its *scaled cumulative stochastic gradient* $x_i^{(t,T)} - x^{(t)} = -\eta_t \sum_{k=0}^{T-1} \widetilde{\nabla} f_i(x_i^{(t,k)})$ to each of its out-neighbors $j \in \mathcal{N}_i^+(t)$ before the t-th global aggregation round. To facilitate this, we assume that every cluster $\ell \in [c]$ contains an access point to which every client $i \in V_\ell(t)$ sends a list of its in-neighbors (clients whose gradients i has received). The access point then announces the end of the concerned D2D communication round, determines the out-degree sequence $\{d_j^+(t) : j \in V_\ell(t)\}$ of the cluster, and broadcasts this sequence to every client in the cluster.

Subsequently, the client computes the following weighted sum of all the scaled cumulative gradients it receives from its in-neighbors:

$$\Delta_i(t) = \sum_{j \in \mathcal{N}_i^-(t)} \frac{1}{d_j^+(t)} \left(x_j^{(t,T)} - x^{(t)} \right). \tag{3}$$

This rule can be expressed compactly in matrix form as

$$\mathbf{\Delta}(t) = A(t) X_{\text{diff}}^\top(t), \tag{4}$$

where $\mathbf{\Delta}(t) := [\Delta_1(t)\ \Delta_2(t)\ \cdots\ \Delta_n(t)]^\top$, $X_{\text{diff}}(t) := [x_1^{(t,T)} - x^{(t)}\ \ x_2^{(t,T)} - x^{(t)}\ \ \cdots\ \ x_n^{(t,T)} - x^{(t)}]$, and $A(t) \in \mathbb{R}^{n \times n}$ is a matrix whose (i,j)-th entry equals $a_{ij}(t) = \frac{1}{d_j^+(t)}$ for all $i \in [n]$ and $j \in \mathcal{N}_i^-(t)$.

Fact 1: $A(t)$ is a column-stochastic matrix because the following holds for all $j \in [n]$:

$$\sum_{i=1}^{n} a_{ij}(t) = \sum_{i \in [n]: j \in \mathcal{N}_i^-(t)} \frac{1}{d_j^+(t)} = \sum_{i \in \mathcal{N}_j^+(t)} \frac{1}{|\mathcal{N}_j^+(t)|} = 1.$$

It can be verified that $A(t)$ is a block-diagonal matrix with its blocks $\{A_\ell(t)\}_{\ell=1}^{c}$ being the equal-neighbor adjacency matrices of the c clusters in the D2D network.

Henceforth, we refer to $A(t)$ as the *equal-neighbor adjacency matrix* of $G(t)$ because it represents every client $i \in [n]$ transmitting an equal share (a fraction $\frac{1}{d_i^+(t)}$) of its scaled cumulative gradient to its $d_i^+(t)$ out-neighbors.

C. Global Aggregation at the PS

For the global aggregation step, the PS samples a random subset of clients $\mathcal{S}(t) \subset [n]$. The cardinality $m(t) \leq n$ of this set is carefully chosen by our algorithm such that the resulting number of D2S interactions is just enough to complement the intra-cluster aggregations without excessively slowing down the training process.

Specifically, this involves three broad steps: (a) The PS first learns the degree distribution of each cluster. (b) It then computes an upper bound on an error quantity $\phi(t)$ that captures the combined effect of random sampling and the cluster degree distributions on the convergence rate. (c) It computes the minimum value of $m(t)$ required to keep $\phi(t)$ below a desired threshold. More specifically:

1) For the $(t+1)$-th round of global aggregation, the server uses $m(t)$ (computed in the previous iteration) to select $\left\lceil (\frac{m(t)}{n}) n_\ell(t) \right\rceil$ clients uniformly at random from the $n_\ell(t) := |V_\ell(t)|$ clients that constitute cluster $\ell \in [c]$. This ensures that every cluster has a representation in the global aggregation that is proportionate to its size. The resulting set of randomly sampled clients is denoted by $\mathcal{S}(t)$. The server then updates the global model as

$$x^{(t+1)} = x^{(t)} + \frac{1}{m(t)} \sum_{i \in \mathcal{S}(t)} \Delta_i(t) \tag{5}$$

$$= x^{(t)} + \frac{1}{m(t)} \sum_{i=1}^{n} \tau_i(t) \Delta_i(t), \tag{6}$$

where $\tau_i(t) := |\{i\} \cap \mathcal{S}(t)|$ is an indicator random variable that takes the value 1 when client i is sampled and the value 0 otherwise. Note that $\sum_{i=1}^{n} \tau_i(t) = |\mathcal{S}(t)| = m(t)$.
2) The current round is now $t \leftarrow t+1$. All the cluster access points send their respective out-degree sequences to the server. Using this information, the server computes

$\alpha_\ell(t) := \frac{1}{n_\ell(t)} \min_{i \in V_\ell(t)} d_i^+(t)$, the *minimum out-degree fraction* of cluster $\ell \in [c]$. The server then uses either of the two sets of singular value bounds that we later derive in Section V (either (12) and (13) or (17) and (18)) to compute an upper bound $\psi(m(t), \alpha_1(t), \ldots, \alpha_c(t))$ on the *connectivity factor* affecting the convergence rate. This connectivity factor, motivated in part by the discussion in Section II, is defined as

$$\phi(t) := \left(\frac{n}{m(t)} - 1\right) \sum_{\ell=1}^{c} \frac{n_\ell(t)}{n} \phi_\ell(t), \tag{7}$$

where $\phi_\ell(t) := \sigma_1^2(A_\ell(t)) + \sigma_2^2(A_\ell(t)) - 1$ depends on the greatest two singular values $\sigma_1(A_\ell(t)) \geq \sigma_2(A_\ell(t))$ of the equal-neighbor adjacency matrix $A_\ell(t)$ of cluster ℓ. For the upper bound, we will show that

$$\psi(m(t), \alpha_1(t), \ldots, \alpha_c(t)) = \left(\frac{n}{m(t)} - 1\right) \sum_{\ell=1}^{c} \frac{n_\ell(t)}{n} \psi_\ell(t),$$

where either of the following holds (with the indexing (t) on the right hand side omitted for brevity):

$$\psi_\ell(t) = 1 + \varepsilon_\ell + \left(\frac{1}{\alpha_\ell} - 1\right)^2 + 2\varepsilon_\ell \left(1 + \frac{2}{\alpha_\ell} - \frac{1}{\alpha_\ell^2}\right),$$

$$\psi_\ell(t) = 2 + 2\varphi_\ell - \frac{(1-\varepsilon_\ell)^2(1-\alpha_{-\ell}^2)\left((1-\varepsilon_\ell)^2(1-\alpha_{-\ell}^2) - \alpha_{-\ell}\right)}{n_\ell(\varepsilon_{\text{net},\ell}+1)\left(\varepsilon_{\text{net},\ell} - \alpha_{-\ell} + \frac{1}{\alpha_\ell n_\ell}\right)} \tag{8}$$

with $\varepsilon_\ell(t) := \frac{d^+_{\max}(t) - d^+_{\min}(t)}{d^+_{\max}(t)}$, $\varphi_\ell(t) := \frac{d^-_{\max}(t) - d^+_{\min}(t)}{d^+_{\min}(t)}$, $\alpha_{-\ell}(t) := \frac{1}{\alpha_\ell(t)} - 1$ and $\varepsilon_{\text{net},\ell}(t) = \varphi_\ell(t) + \frac{\varepsilon_\ell(t)}{\alpha_\ell(t)}$.

3) Finally, the server sets $m(t+1)$ equal to $\min\{r \in [n] : \psi(r, \alpha_1(t+1), \ldots, \alpha_c(t+1)) \leq \phi_{\max}\}$, where $\phi_{\max}$ is a threshold given as an input to the algorithm. This step ensures that $\phi(t)$ remains below the threshold $\phi_{\max}$, thereby preserving the convergence rate.

Our algorithm for $t_{\max}$ global rounds is given in Algorithm 1.

Algorithm 1: Connectivity-Aware Semi-Decentralized Learning.

Input: n, c, T, $\phi_{\max}$, $t_{\max}$, $m(0)$, $\{n_\ell(t)\}_{t=0}^{t_{\max}}$, $x^{(0)}$
Output: $x^{(t_{\max})}$

for $t \in \{0, 1, \ldots, t_{\max} - 1\}$ **do**
  Client $i \in [n]$ sets $x_i^{(t,0)} \leftarrow x^{(t)}$
  for $k \in \{0, 1, \ldots, T-1\}$ **do**
    Client $i \in [n]$ computes $x_i^{(t,k+1)} \leftarrow x_i^{(t,k)} - \eta_t \widetilde{\nabla} f_i(x_i^{(t,k)})$
  end
  Client $i \in [n]$ transmits its scaled cumulative local gradient $-\eta_t \sum_{k=0}^{T-1} \widetilde{\nabla} f_i(x_i^{(t,k)}) = x_i^{(t,T)} - x^{(t)}$ to its out-neighbors $\mathcal{N}_i^+(t)$
  Client $i \in [n]$ computes the following weighted sum of its in-neighbors' cumulative local gradients:

$$\Delta_i(t) \leftarrow \sum_{j \in \mathcal{N}_i^-(t)} \frac{1}{d_j^+(t)} \left(x_j^{(t,T)} - x^{(t)}\right)$$

  PS samples $m_\ell(t) = \frac{n_\ell(t)}{n} m(t)$ clients uniformly at random from cluster $\ell \in [c]$
  PS computes $x^{(t+1)} \leftarrow x^{(t)} + \frac{1}{m(t)} \sum_{i=1}^{n} \tau_i(t) \Delta_i(t)$ and broadcasts $x^{(t+1)}$ to all clients
  PS computes $m(t+1) \leftarrow \min\{r \in [n] : \psi(r, \alpha_1(t+1), \ldots, \alpha_c(t+1)) \leq \phi_{\max}\}$
end
return $x^{(t_{\max})}$

V. Convergence Analysis

We now provide theoretical performance guarantees for Algorithm 1. We also explain how the effect of D2D cluster connectivity on the convergence rate of the algorithm is captured by the singular values of the equal-neighbor adjacency matrices of the clusters. For all the calculations omitted from the proof sketches, please refer to the supplementary material.

A. Assumptions and Preliminaries

1) Loss Functions: We start by making the following standard assumptions on the local loss functions:

Assumption 1 (Strong Convexity): All the local loss functions $\{f_i\}_{i=1}^n$ are μ-strongly convex, i.e., there exists $\mu > 0$ such that $(\nabla f_i(x) - \nabla f_i(y))^\top (x - y) \geq \mu \|x - y\|^2$ for all $x, y \in \mathbb{R}^p$ and all $i \in [n]$.

Assumption 2 (Smoothness): All the local loss functions $\{f_i\}_{i=1}^n$ are β-smooth, i.e., there exists a finite β such that $\|\nabla f_i(x) - \nabla f_i(y)\| \leq \beta \|x - y\|$ for all $x, y \in \mathbb{R}^p$ and all $i \in [n]$.

As shown in [11], Assumptions 1 and 2 imply that the global loss function f is both μ-strongly convex and β-smooth.

2) SGD Iterations: Additionally, we make the following standard assumption on the stochastic gradients generated through the SGD procedure for each client:

Assumption 3 (Unbiasedness and Bounded Variance): The SGD noise associated with every client is unbiased, i.e., $\mathbb{E}[\widetilde{\nabla} f_i(x) - \nabla f_i(x) \mid x] = 0$, and it has a bounded variance, i.e., there exists a constant $\varrho > 0$ such that $\mathbb{E}\|\widetilde{\nabla} f_i(x) - \nabla f_i(x)\|^2 \leq \varrho^2$ for all models $x \in \mathbb{R}^p$ and all $i \in [n]$.

In addition, we assume that the SGD noise is independent across clients, i.e., for all $x \in \mathbb{R}^p$, the random vectors $\{\widetilde{\nabla} f_i(x) - \nabla f_i(x)\}_{i=1}^n$ are mutually conditionally independent given x.

3) Gradient Diversity: Furthermore, we assume that the training data are not distributed uniformly at random among the clients, which gives rise to data heterogeneity among the clients. Unlike the standard assumption on data heterogeneity that imposes a uniform upper bound on $\|\nabla f_i(x) - \nabla f(x)\|$ (see [51] for example), we make a weaker assumption on the diversity of local gradients. In fact, this assumption, which

was first proposed in [11], can be derived as a consequence of Assumptions 1 and 2, as shown in [11]. Below, we formally state this observation.

Lemma 2 (Gradient diversity [11]): For all $i \in [n]$ and $x \in \mathbb{R}^p$, we have $\|\nabla f_i(x) - \nabla f(x)\| \leq \delta + 2\beta\|x - x^*\|$, where

$$\delta := \beta \max_{i\in[n]} \|x^* - x_i^*\| = \beta \max_{i\in[n]} \|x^* - \arg\min_{y\in\mathbb{R}^p} f_i(y)\| \quad (9)$$

As argued in [11], the standard assumption (which is a special case of the above inequality with $\beta = 0$) is unrealistic as it does not apply to quadratic and super-quadratic loss functions unless the upper bound δ is chosen to be unreasonably large.

B. Results

We now quantify how the singular values of the equal-neighbor matrices and the number of clients sampled by the PS affect the efficiency of our algorithm in terms of its optimality gap.

We first show how the expected optimality gap of our algorithm depends on the expected deviation of the global average $x^{(t+1)} - x^{(t)}$ (i.e., the random vector computed by the PS using the aggregation rule (5)) from the true average of all the scaled cumulative gradients.

Lemma 3: At the end of the $(t+1)$-th round of global aggregation, the expected optimality gap of Algorithm 1 is given by

$$\mathbb{E}\left\|x^{(t+1)} - x^*\right\|^2 = \mathbb{E}\left\|x^{(t+1)} - \bar{x}^{(t+1)}\right\|^2 + \mathbb{E}\left\|\bar{x}^{(t+1)} - x^*\right\|^2,$$

where $\bar{x}^{(t+1)} := x^{(t)} + \frac{1}{n}\sum_{i=1}^n (x_i^{(t,T)} - x^{(t)})$ is a vector that would equal the global model if the PS were to sample all the n clients.

Proof: The key steps are to note that

$$\mathbb{E}\left\|x^{(t+1)} - x^*\right\|^2 = \mathbb{E}\left\|x^{(t+1)} - \bar{x}^{(t+1)}\right\|^2 + \mathbb{E}\left\|\bar{x}^{(t+1)} - x^*\right\|^2 + 2\mathbb{E}\left[\left(x^{(t+1)} - \bar{x}^{(t+1)}\right)^T \left(\bar{x}^{(t+1)} - x^*\right)\right]$$

and to show that the cross-term above vanishes. To this end, let $v(t) \in \mathbb{R}^n$ denote the vector of server-assigned weights, i.e., $v_i(t) = \frac{1}{m}$ if $i \in [n]$ is sampled by the PS at time t and $v_i(t) = 0$ otherwise. We then observe that $x^{(t+1)} - \bar{x}^{(t+1)} = X_{\text{diff}}(A^T(t)v(t) - \frac{1}{n}\mathbf{1})$. As a result, we can use (a) the independence of $v(t)$ and the matrix of local models $X_{\text{diff}}(t)$, (b) the column-stochasticity of $A(t)$, and (c) the fact that every node is sampled with the same probability to show that $\mathbb{E}[(x^{(t+1)} - \bar{x}^{(t+1)})^T(\bar{x}^{(t+1)} - x^*)] = 0$. □

Observe that the first term on the RHS depends on $x^{(t+1)} - \bar{x}^{(t+1)}$, which can be easily shown to be the difference between the random average $\frac{1}{m(t)}\sum_{i\in\mathcal{S}(t)} \Delta_i(t)$ and the true average $\frac{1}{n}\sum_{i=1}^n (x_i^{(t,T)} - x^{(t)})$. Thus, this term captures the error due to random sampling. As the next result shows, this difference depends on the network topology as well as on $m(t)$, the number of clients selected for global aggregation uniformly at random by the PS.

Proposition 1: Let δ be the constant defined in (9). Then Algorithm 1 satisfies the following for every $t \in \mathbb{N} \cup \{0\}$:

$$\mathbb{E}\left\|x^{(t+1)} - \bar{x}^{(t+1)}\right\|^2 \leq \Big(2T\varrho^2\eta_t^2 + 4eT(\varrho^2 + 2\delta^2)\eta_t^2 + 6\delta^2\, T^2\eta_t^2 + (27 + 4e)T^2\beta^2\eta_t^2\mathbb{E}\left\|x^{(t)} - x^*\right\|^2\Big)\phi(t),$$

where $\phi(t)$ is the connectivity factor defined in (7).

The proof of this proposition is based on the following key result that helps us connect the greatest two singular values of $A(t)$ with $\mathbb{E}\|x^{(t+1)} - \bar{x}^{(t+1)}\|^2$.

Lemma 4: For $s \in \mathbb{N}$, let $v \in \mathbb{R}^s$ be a stochastic vector, and let $A \in \mathbb{R}^{s\times s}$ be an irreducible column-stochastic matrix with positive diagonal entries. Then $\|A^T v - \frac{1}{s}\mathbf{1}\|^2 \leq (\sigma_1^2 + \sigma_2^2 - 1)\|v_\perp\|^2$, where $v_\perp := v - \frac{1}{s}\mathbf{1}$ is the component of v that is orthogonal to $\mathbf{1}$, and σ_1 and σ_2 are the largest and the second-largest singular values of A, respectively.

Proof: We first show that the quantity in question equals $v_\perp^T AA^T v_\perp$, derive an inequality connecting the principal eigenvector of AA^T with σ_1 and σ_2, and then use the results of each of these steps to obtain the desired upper bound.

For the first step, we use $A^T\mathbf{1} = \mathbf{1}$ repeatedly to show that $\|v^T A - \frac{1}{s}\mathbf{1}^T\|^2 = v_\perp^T AA^T v_\perp$. We call this Observation 1.

Next, we let $\hat{p}$ denote the unit-norm principal eigenvector of AA^T, and we relate σ_1 and σ_2 to $\hat{p}$. To do so, we first note that AA^T is irreducible by Lemma 6 in the supplementary material, and hence, the Perron-Frobenius theorem implies that $\hat{p}$ is unique up to scaling by a complex scalar of unit magnitude. We now let $\{\hat{v}_j\}_{j=2}^s$ denote the unit-norm eigenvectors of AA^T corresponding to its eigenvalues $\{\sigma_j^2\}_{j=2}^s$, where σ_j denotes the j-th largest singular value of A. Here, we apply the spectral decomposition theorem for symmetric matrices to $AA^\top$ to make the critical observation that $s = \sigma_1^2(\mathbf{1}^T\hat{p})^2 + \sigma_2^2(s - (\mathbf{1}^T\hat{p})^2)$. This enables us to obtain $\frac{1}{s}(\mathbf{1}^T\hat{p})^2 \geq \frac{1-\sigma_2^2}{\sigma_1^2-\sigma_2^2}$. The final step is to upper bound $v_\perp^T AA^T v_\perp$. To this end, let $\hat{p}$ denote the unit-norm principal eigenvector of AA^T, let $\hat{p}_\perp := \hat{p} - \frac{1}{s}(\hat{p}^T\mathbf{1})\mathbf{1}$ denote the component of $\hat{p}$ that is orthogonal to $\mathbf{1}$, and let $v_\perp := v_\perp - (v_\perp^T\hat{p})\hat{p}$ denote the component of $v_\perp$ that is orthogonal to $\hat{p}$. We then have

$$\begin{aligned} v_\perp^T AA^T v_\perp &= \left(v_\perp + (v_\perp^T\hat{p})\hat{p}\right) AA^T \left(v_\perp + (v_\perp^T\hat{p})\hat{p}\right) \\ &\overset{(a)}{=} v_\perp^T AA^T v_\perp + (v_\perp^T\hat{p})^2\hat{p}^T AA^T\hat{p} \\ &\overset{(b)}{\leq} \sigma_2^2\|v_\perp\|^2 + (v_\perp^T\hat{p})^2\sigma_1^2 \\ &\overset{(c)}{\leq} (\sigma_1^2 + \sigma_2^2 - 1)\|v_\perp\|^2, \end{aligned} \quad (10)$$

where (a) holds because $v_\perp^T\hat{p} = 0$, (b) follows from the Courant-Fischer theorem and the fact that $v_\perp^T$ is orthogonal to the principal eigenspace $\{\beta\hat{p} : \beta \in \mathbb{R}\}$ of AA^T, and (c) is derived using the Pythagoras' theorem and the orthogonalities of v and $\hat{p}_\perp$ with $\mathbf{1}$,

Cauchy-Schwarz inequality, and from our preceding observation that $\frac{1}{s}(\mathbf{1}^T\hat{p})^2 \geq \frac{1-\sigma_2^2}{\sigma_1^2-\sigma_2^2}$. Combining (10) with Observation 1 now yields the required upper bound. □

We are now ready to prove Proposition 1.

Proof: Reusing $x^{(t+1)} - \bar{x}^{(t+1)} = X_{\text{diff}}(A^T(t)v(t) - \frac{1}{n}\mathbf{1})$ (an observation made in the proof of Lemma 3) yields

$$\mathbb{E}\left\|x^{(t+1)} - \bar{x}^{(t+1)}\right\|^2 = \mathbb{E}\left\|X_{\text{diff}}(t)\left(A^T(t)v(t) - \frac{1}{n}\mathbf{1}\right)\right\|^2$$

$$\leq \mathbb{E}\left\|X_{\text{diff}}(t)\right\|^2 \mathbb{E}\left\|A^T(t)v(t) - \frac{1}{n}\mathbf{1}\right\|^2$$

$$\overset{(a)}{\leq} \left(\sum_{i=1}^{n}\mathbb{E}\left\|x_i^{(t,T)} - x^{(t)}\right\|^2\right)\left(\frac{1}{m(t)} - \frac{1}{n}\right)\sum_{\ell=1}^{c}\frac{n_\ell}{n}\phi_\ell(t),$$

where (a) follows from the fact that every $n \times n$ matrix has its squared spectral norm upper bounded by the sum of the squares of its column norms (Lemma 4 in the supplementary material) and Lemma 3. Using Lemma 11 (which is proved in the supplementary material using standard upper bounding techniques) now yields the desired result. □

In other words, $\mathbb{E}\|x^{(t+1)} - \bar{x}^{(t+1)}\|$ depends on the previous optimality gap $\mathbb{E}\|x^{(t)} - x^*\|^2$ via $\phi(t)$, i.e., the connectivity factor that captures the combined effect of global aggregation (via $m(t)$) and the D2D network topology within each cluster (via $\phi_\ell(\alpha_\ell(t))$).

Moreover, Lemma 3 and Proposition 1 together show that the singular values of the equal-neighbor adjacency matrices can be used to derive an upper bound on the expected optimality gap (and ultimately establish theoretical performance guarantees) for our connectivity-aware algorithm. Doing so yields the following.

Proposition 2: Let δ be as defined in (7), let $\phi(t)$ be the connectivity factor defined in (7), let $\Gamma := f(x^*) - \frac{1}{n}\sum_{i=1}^{n}\min_{x\in\mathbb{R}^p} f_i(x)$, and let e denote the exponential constant. Then the expected optimality gap of Algorithm 1 satisfies the following for all $t \in \mathbb{N}_0$:

$$\mathbb{E}\left\|x^{(t+1)} - x^*\right\|^2$$

$$\leq \left((1-\mu\eta_t)^T + (27+4e)T^2\beta^2\eta_t^2(2\,T+\phi(t))\right)\mathbb{E}\left\|x^{(t)} - x^*\right\|^2$$

$$+ T\left(\frac{\varrho^2}{n} + 6\beta\Gamma + 4T\varrho^2 + 8eT(\varrho^2+2\delta^2) + 12\delta^2\,T^2\right)\eta_t^2$$

$$+ \left(2T\varrho^2 + 4eT(\varrho^2+2\delta^2) + 6\delta^2\,T^2\right)\phi(t)\eta_t^2.$$

Proof: We quantify the difference between the iterates $x_i^{(t,k)}$ resulting from stochastic gradient descent and the iterates $\beta^{(t,k)}$ resulting from centralized non-stochastic gradient descent by first defining $\beta^{(t,k)} := \beta^{(t,k-1)} - \eta_t\nabla f(\beta^{(t,k-1)})$ (with $\beta^{(t,0)} := x^{(t)}$) and then by using Cauchy-Schwarz and Young's inequalities along with standard inductive arguments based on strong convexity and smoothness in order to obtain an upper bound on $\sum_{k=0}^{T-1}\mathbb{E}\|x_i^{(t,k)} - \beta^{(t,k)}\|^2$ in terms of $\mathbb{E}\|x^{(t)} - x^*\|^2$, which results in Lemma 9 in the supplementary material. We then use the resulting expressions along with the independence of the zero-mean random vectors $\{\widetilde{\nabla} f_i(x_i^{(t,q)}) - \nabla f_i(x^{(t,q)})\}_{i=1}^n$ to upper bound $\mathbb{E}\|x_i^{(t,k)} - x^{(t)}\|^2$ in terms of $\mathbb{E}\|x^{(t)} - x^*\|^2$. This results in Lemma 11 in the supplementary material. Next, we use the resulting inequality along with the property of averages that the arithmetic mean of a finite set of vectors $\{z_i\}_{i=1}^n$ is the minimizer of the unweighted mean-square loss function $\mathbb{R} \ni y \to \tilde{\ell}(y) = \frac{1}{n}\sum_{i=1}^{n}\|z_i - y\|^2$ in order to obtain Lemma 12 in the supplementary material. A simple induction on k enables us to extend the result to an upper bound on the k-independent quantity $\mathbb{E}\|\bar{x}^{(t+1)} - x^*\|^2$ as shown in Lemma 13 in the supplementary material. As the last step, we combine the resulting expressions with Lemma 2 and Proposition 1 in order to obtain the desired inequality. □

A recursive expansion on the inequality in Proposition 2 results in our main theoretical result, which we state below.

Theorem 3: Consider a connectivity factor threshold $\phi_{\max} \geq 0$, and suppose that $\phi(t) \leq \phi_{\max}$ for all times $t \geq 0$. In addition, suppose $\eta_t = \frac{4}{T\mu(t+t_1)}$, where $t_1 := \lfloor 4(1-\frac{1}{T}) + (16\,T + 8\phi_{\max})(\frac{\beta}{\mu})^2 + 1\rfloor$. Then the expected optimality gap of Algorithm 1 satisfies the following for all $t \geq 0$:

$$\mathbb{E}\left\|x^{(t)} - x^*\right\|^2$$

$$\leq \left(\frac{t_1}{t+t_1}\right)^2\mathbb{E}\left\|x^{(0)} - x^*\right\|^2 + \frac{16\left(\frac{1}{nT}\left(\frac{\varrho}{\mu}\right)^2 + 6\frac{\beta\Gamma}{T\mu^2}\right)}{t+t_1}$$

$$+ (32T + 16\phi_{\max})\left(\frac{2}{T}\left(\frac{\varrho}{\mu}\right)^2\right.$$

$$\left. + \frac{4e}{T}\left(\left(\frac{\varrho}{\mu}\right)^2 + 2\left(\frac{\delta}{\mu}\right)^2\right) + 6\left(\frac{\delta}{\mu}\right)^2\right)t + t_1. \qquad (11)$$

Theorem 3 reveals that the convergence rate of our algorithm is $\mathcal{O}(1/t)$, which coincides with that of FedAvg and its semi-decentralized variants such as [11]. In fact, $\mathcal{O}(1/t)$ resembles the convergence rate of vanilla centralized SGD. It also shows that suitably tuning the connectivity factor (by choosing an appropriate value of $\phi_{\max}$) is critical to the efficiency of the algorithm: as $\phi_{\max}$ increases the bound gets worse/larger; however, $\phi_{\max}$, by its definition, is non-negative, which means it can at best be made equal to 0, which forces $m = n$, in which case the inequality boils down to an upper bound on the convergence rate of FedAvg with full device sampling. At the other extreme, setting $\phi_{\max}$ to ∞ results in $m = 1$, which happens when our semi-decentralized FL architecture collapses to full decentralization.

Moreover, Theorem 3 jointly captures the effect of the following factors on the expected instantaneous optimality gap and hence on the convergence rate: (i) the initial optimality gap $\mathbb{E}\|x^{(0)} - x^*\|^2$ (via the first term), (ii) The SGD noise variance ϱ^2 and the strong convexity μ and smoothness parameters β (via the second term), and finally, (iii) the combined effect of cluster connectivity levels and random sampling-based global aggregations (via the third term, which depends on $\phi_{\max}$, which in turn prevents the connectivity factor $\phi(t)$ from becoming too

large). It can be seen that higher values of the SGD noise variance ϱ^2 and the data heterogeneity measure Γ lead to a larger value of the bound, implying that our algorithm is sensitive to the size of the mini-batches used for computing the stochastic gradients as well as to the non-i.i.d.-ness of the local datasets.

VI. Singular Value Bounds

Having established the role of the connectivity factor $\phi(t)$ in the performance of Algorithm 1, we now analyze two important quantities associated with $\phi(t)$: the top two singular values of the equal-neighbor adjacency matrices of the clusters. Since a precise estimation of these singular values requires full knowledge of the cluster topologies, which is challenging to obtain in practice, we are motivated to derive a set of novel upper bounds on these values in terms of the node degrees of the cluster digraphs, which are easy to obtain/measure in practice. To the best of our knowledge, this is one of the first attempts at connecting the singular values of adjacency matrices with minimal topological information such as node degrees of the digraphs.

To conduct our analysis, for any digraph $G=([s],E)$, we first define $\varepsilon=\varepsilon_G:=\frac{d^+_{\max}(G)-d^+_{\min}(G)}{d^+_{\max}(G)}$, which quantifies the heterogeneity of out-degree of the nodes across the digraph. We also let $\alpha(G):=\frac{d^+_{\min}(G)}{s}$ capture the minimum fraction of the node population that any node is out-connected to. In addition, we let $W(G)=(w_{ij})$ and $D^+(G):=\mathrm{diag}([d_1^+\ d_2^+\ \cdots\ d_s^+]^\top)$ denote the binary adjacency matrix and the out-degree matrix of G, respectively. In the sequel, we drop the indexing (G) for brevity.

We are now equipped to state our first set of bounds on the greatest two singular values of G under certain regularity assumptions on the digraph.

Proposition 4: Suppose $G=([s],E)$ is a directed graph in which every node has its in-degree equal to its out-degree, i.e., $d_i^+=d_i^-$ for all $i\in[n]$. Then the greatest two singular values σ_1 and σ_2 of the equal-neighbor adjacency matrix A of G satisfy the following inequalities for $\alpha>\frac{1}{2}$ and $\varepsilon\ll 1$:

$$\sigma_1^2\le 1+\varepsilon+\mathcal{O}(\varepsilon^2), \tag{12}$$

$$\sigma_2^2\le\left(\frac{1}{\alpha}-1\right)^2+2\varepsilon\left(1+\frac{2}{\alpha}-\frac{1}{\alpha^2}\right)+\mathcal{O}(\varepsilon^2), \tag{13}$$

where $\mathcal{O}(\cdot)$ is the big-O notation used in the context of $\varepsilon\to 0$.

Proof: To simplify our notation, we define $D:=D^+$ (where D^+ is the diagonal matrix of out-degrees) for the remainder of this proof. Observe that $A^\top=D^{-1}W=D^{-\frac{1}{2}}(D^{-\frac{1}{2}}WD^{-\frac{1}{2}})D^{\frac{1}{2}}$, which means $A^\top$ is similar to the normalized adjacency matrix defined as $A_N:=D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$.

On the other hand, we have $d^+_{\min}=d^+_{\max}(1-\varepsilon)\le d_i^+\le d^+_{\max}$ for all $i\in[s]$, which implies the existence of a diagonal matrix E_3 such that $O\le E_3\le I$ and $D=d^+_{\max}((1-\varepsilon)I+\varepsilon E_3)$. Using similar arguments, it can be easily shown that there exist diagonal matrices E_1 and E_2 such that $O\le E_1,E_2\le I$, $D^{\frac{1}{2}}=\sqrt{d^+_{\max}}((1-\frac{\varepsilon}{2})I+\frac{\varepsilon}{2}E_1)+\mathcal{O}(\varepsilon^2)$, and $D^{-\frac{1}{2}}=\frac{1}{\sqrt{d^+_{\max}}}(I+\frac{\varepsilon}{2}E_2)+\mathcal{O}(\varepsilon^2)$. As a result of this and some simplification, we obtain $A^\top-A_N=-\frac{\varepsilon}{2}((E_1-I)A^\top+A^\top E_2)+\mathcal{O}(\varepsilon^2)$. In conjunction with standard bounds on singular value perturbations (e.g., see [49]) and in light of the similarity of $D^{-1}WD^{-1}W^\top$ and $D^{-\frac{1}{2}}WD^{-1}W^\top D^{-\frac{1}{2}}$, this implies the following:

$$\sigma_j(A)=\sigma_j(A^\top)=\sqrt{\lambda_j(D^{-1}WD^{-1}W^\top)}+\varepsilon\sigma_1(A^\top)+\mathcal{O}(\varepsilon^2). \tag{14}$$

We now bound $\sigma_1(A)$ and $\sigma_2(A)$ individually. As for $\sigma_1(A)$, the derivation (14) and the fact that $\sigma_1(A)=\sigma_1(A^\top)$ imply

$$\sigma_1(A)\le(1-\varepsilon)^{-1}\sqrt{\lambda_1(D^{-1}WD^{-1}W^T)}=\sqrt{\lambda_1(D^{-1}WD^{-1}W^T)}(1+\varepsilon)+\mathcal{O}(\varepsilon^2). \tag{15}$$

So, it is enough to bound $\lambda_1(D^{-1}WD^{-1}W^T)$. For this purpose, note that A being column-stochastic implies that $D^{-1}W\mathbf{1}=\mathbf{1}$ and hence also that $W\mathbf{1}=D\mathbf{1}$. Besides, our assumption on in-degrees and out-degrees can be expressed as $\sum_{j=1}^s w_{ij}=\sum_{j=1}^s w_{ji}$ for each $i\in[s]$, or equivalently, $W^\top\mathbf{1}=W\mathbf{1}=D\mathbf{1}$. As a result, we have $D^{-1}W^\top\mathbf{1}=\mathbf{1}$. Thus, $D^{-1}WD^{-1}W^\top=A^\top D^{-1}W^\top$ is a product of row-stochastic matrices and hence, it is row-stochastic in itself. Thus, $\lambda_1(D^{-1}WD^{-1}W^T)=1$ and (15) implies (12).

It remains to prove (13). We do this by using Theorem 2.2 of [52], which helps derive a bound in terms of σ_1 and the minimum positive entry δ of the matrix $D^{-1}WD^{-1}W^T$. We first note that $(D^{-1}WD^{-1}W^T)_{ij}\ge\frac{(2\alpha-1)s}{(d^+_{\max})^2}$, which is derived using the fact that the number of common out-neighbors of any two nodes $i,j\in[s]$ is at least $(2\alpha-1)s$. We can now apply Theorem 2.2 of [52] by setting $x=\frac{1}{\sqrt{s}}$ in the theorem (because $\frac{1}{\sqrt{s}}\mathbf{1}$, as explained above, is the unit-norm principal eigenvector of $D^{-1}WD^{-1}W^\top$). Thus,

$$\lambda_2(D^{-1}WD^{-1}W^\top)\le\lambda_1(D^{-1}WD^{-1}W^\top)-\frac{(2\alpha-1)s^2}{(d^+_{\max})^2}=1-\left(\frac{2}{\alpha}-\frac{1}{\alpha^2}\right)(1-2\varepsilon)+\mathcal{O}(\varepsilon^2), \tag{16}$$

where the last step holds because $d^+_{\max}=\frac{\alpha s}{1-\varepsilon}$.

Combining (12), (14) and (16) now gives

$$\sigma_2(A)\le\sqrt{1-\left(\frac{2}{\alpha}-\frac{1}{\alpha^2}\right)(1-2\varepsilon)+\mathcal{O}(\varepsilon^2)}+\varepsilon+\mathcal{O}(\varepsilon^2).$$

Squaring both sides and rearranging the terms results in (13).□

Remark 1: Observing the bounds in Proposition 4, we can see that setting $\alpha=1$, which corresponds to G being a clique, in the bounds yields $\sigma_1\le 1+\mathcal{O}(\varepsilon)$ and $\sigma_2=\mathcal{O}(\varepsilon)$. These inequalities, for $\varepsilon\ll 1$, are tight with respect to the well-known lower bounds $\sigma_1\ge 1$ and $\sigma_2\ge 0$. This implies that the bounds (12) and (13) can be expected to be reasonably tight for high edge density (i.e., whenever $\alpha\approx 1$). Another implication of the bounds is decreasing ε, which measures how irregular the digraph is, leads to (13) becoming sharper.

The above singular value bounds are especially tight for digraphs that are approximately regular (or digraphs that do not exhibit significant variations in their in-degrees and out-degrees) since such digraphs satisfy $\varepsilon \ll 1$. This happens in practice, when the D2D clusters are dense, such as in the wireless setting where the nodes are close to each other or when they can move and communicate over time). Furthermore, the same holds for the condition on α in Proposition 4 (i.e., $\alpha > \frac{1}{2}$), which is always met when the clusters are dense. Thus, we expect Proposition 4 to apply, for example, to local wireless mesh networks where devices are in close proximity with rich connectivity [19] and also to swarms of UAV networks where the UAVs communicate while traveling together in close groups [53].

However, the bounds (12) and (13) are obtained under the assumption that every node has its in-degree equal to its out-degree, which can be restrictive in practical settings. This observation further motivates us to find a new set of singular value bounds that work well under milder assumptions. We thus provide the following bounds, which not only relax the said restrictive assumption, but also apply to digraphs with more general out-degree distributions (and hence subsume digraphs with wider out-degree variations).

Proposition 5: Let $\varphi = \frac{d^{\text{in}}_{\max} - d_{\min}}{d_{\min}}$, where $d^{in}_{\max}$ denotes the maximum in-degree of the digraph G. If $\alpha \geq \frac{1}{2}$, we have the following bounds:

$$\sigma_1^2 \leq 1 + \varphi, \tag{17}$$

$$\sigma_2^2 \leq 1 + \varphi - \frac{(1-\varepsilon)^2(1-\alpha_{-1}^2)\left((1-\varepsilon)^2(1-\alpha_{-1}^2) - \alpha_{-1}\right)}{s(\varepsilon_{\text{net}} + 1)\left(\varepsilon_{\text{net}} - \alpha_{-1} + \frac{1}{\alpha s}\right)}, \tag{18}$$

where $\varepsilon_{\text{net}} := \varphi + \frac{\varepsilon}{\alpha}$ and $\alpha_{-1} := \frac{1}{\alpha} - 1$.

Proof: The proof is carried out based on lengthy computations and upper bounding techniques that involve using bounds on the minimum and maximum entries of the Perron eigenvector of certain positive matrices associated with the digraph G. For the detailed proof, please see the supplementary material. □

The bounds obtained in Proposition 5 are particularly effective when the D2D cluster digraphs are dense but moderately irregular. This is often the case in practical systems, when there is communication heterogeneity (e.g., in wireless sensor networks consisting of sensors with different radii).

In conjunction with Theorem 3, the bounds derived in Propositions 4 and 5 capture the inherent dependence of the expected optimality gap, and hence that of the convergence rate, on the degree distributions of the D2D clusters. In particular, upon having approximately regular D2D clusters, the bounds in Proposition 4 along with the result of Theorem 3 determine the convergence rate of Algorithm 1. The same holds when using the result of Proposition 5 with Theorem 3, which will characterize the convergence rate of Algorithm 1 upon having irregular D2D clusters.

VII. Motion-Planning Algorithm

Having characterized the dependence of the convergence rate on the deviation of the cluster topologies from digraph regularity, we now develop a method aimed at making cluster digraphs more regular. This method is a motion-planning algorithm that assumes that some of the mobile clients in the network have physical trajectories that can be controlled with a high probability. Such clients will be referred to as *control nodes* or *C-nodes* and the remaining will be called *uncontrollable nodes* or *U-nodes*. The objective of our algorithm is to reduce the spread in the out-degree distribution of each cluster digraph by steering the C-nodes within the cluster to U-nodes with low out-degrees, thereby enabling link formations between the concerned C-nodes and U-nodes.

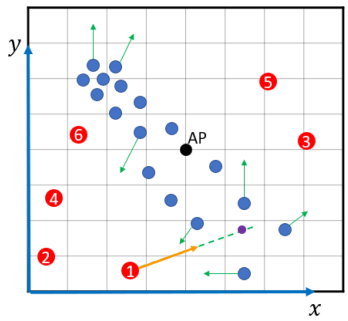

Fig. 4. Schematic diagram of a typical cluster region and our motion-planning configuration. The U-nodes are blue and the C-nodes in red. The purple solid circle is the geometric center of the four closest U-nodes.

Our algorithm makes the following assumptions.

1) Every U-node changes its velocities arbitrarily but periodically with a period $\Delta > 0$.
2) Between consecutive velocity changes, every U-node makes a sojourn at its current location for a short duration of time $\delta < \Delta$, during which it may make measurements or record observations pertaining to tasks that are independent of the model training process of interest. During such sojourns, every C-node approaches (i.e., enters the sensing radius of) the U-node it is assigned to with a high success probability p_A, i.e., $1 - p_A \ll 1$.
3) Suppose client i is a C-node assigned to client j, which is a U-node. Then, given that i has entered the sensing radius of j, we assume that i and j transmit to each other with a high probability p_T (in the sense that $1 - p_T \ll 1$).

On the basis of these assumptions, we define a control *failure* as the event that a given C-node fails to either (a) enter the sensing radius of the U-node that it is assigned to, or (b) to transmit and receive local scaled cumulative gradients to and from the concerned U-node. This failure probability p_F is easily seen to be given by $p_F = (1 - p_A) + p_A(1 - p_T) \ll 1$.

To describe our algorithm in detail, we first need to collect all the U-nodes with the minimum out-degree into the set $S_{\min}$, and we need to define a function that arranges all the subsets of $S_{\min}$ satisfying a certain condition in a specific order. To this end, for every natural number $s \leq |S_{\min}|$, we let $\mathcal{S}_s : \{1, 2, \ldots, \binom{|S_{\min}|}{s}\} \to 2^{S_{\min}}$ be the function that maps every index $i \leq \binom{|S_{\min}|}{s}$ to the i-th set in the lexicographic ordering of all the s-sized subsets of $S_{\min}$.

We now describe the steps of our motion-planning algorithm, each iteration of which corresponds to one sojourn period of

the uncontrollable nodes. As before, the algorithm assumes for each cluster the presence of an access point (AP) that is located in the same cluster region as the mobile clients comprising the cluster (see Fig. 4) and is therefore available to the clients for short-distance communications.[2] For any $j \leq n_C^{(\ell)}$ where $n_C^{(\ell)}$ is the number of control nodes in cluster ℓ, we now describe the steps performed by the j-th node within cluster ℓ as part of any one iteration τ.

1) The AP assigned to cluster ℓ measures and collects the velocities and the positions[3] (location coordinates $\{(x_i(\tau), y_i(\tau)) : c < i \leq n_C^{(\ell)}\}$, where $x_i(\tau)$ and $y_i(\tau)$ are the x- and y-coordinates of the U-node i) of all the U-nodes in the cluster, which we index by $\{i \in V_\ell : c < i \leq n_C^{(\ell)}\}$.
2) It then computes the out-degrees of the U-nodes using its knowledge of the sensing radius and then transmits these out-degrees as well as the location coordinates computed in the previous step to every control node (C-node).
3) C-node j performs the following steps to determine its destination (the physical location within its cluster region that it should travel to during the sojourn times of the U-nodes):

a) It identifies the set $S_{\min}$ of U-nodes with the minimum out-degrees.
b) It sets $S := S_{\min}$, $s := |S|$, and determines whether the geometric center $(x_S(\tau), y_S(\tau))$ (defined as the arithmetic mean of the location coordinates $\{(x_i(\tau), y_i(\tau)) : i \in S\}$) of the s U-nodes within S lies in the intersection I_S of the sensing regions (defined by the sensing radius) of all of the s U-nodes. If $(x_S(\tau), y_S(\tau)) \in I_S$, C-node j chooses $(x_S(\tau), y_S(\tau))$ as its destination.
c) If $(x_S(\tau), y_S(\tau)) \notin I_S$, the C-node decrements the value of s by 1, generates a set $S \subset S_{\min}$ such that S is the first set of size s to appear in the lexicographic ordering of the subsets of $S_{\min}$. It then updates $(x_S(\tau), y_S(\tau))$ and I_S accordingly and chooses $(x_S(\tau), y_S(\tau))$ as its destination under the same condition as in the previous step.
d) If it still finds $(x_S(\tau), y_S(\tau)) \notin S$, then it keeps repeating the previous step until the repetition eventually yields $(x_S(\tau), y_S(\tau)) \in S$, each time choosing S as the next subset in the lexicographic ordering of subsets of size s. In doing so, it decrements the value of s by 1 whenever all the subsets of higher sizes are found to be exhausted.

4) Node j then travels to the destination $(x_S(\tau), y_S(\tau))$ during the sojourn period with probability p_A and establishes bidirectional links with the concerned set of U-nodes with probability p_T.

[2]These communications do not constitute D2S interactions as the server is assumed to be a distinct entity located far away from every cluster region.

[3]We assume this location-sharing between the clients and the AP introduces negligible energy overhead compared with model transmissions. In applications such as mobile edge computing, intelligent traffic systems [54], and cellular communications, such information may already be collected (e.g., through Global Positioning Systems (GPS) [55]), and can be shared with the concerned APs (e.g., base stations, road-side units) using established signalling protocols [56], [57].

Algorithm 2: Motion-Planning for Control Nodes.

Inputs: n, c, number of iterations T_I, sensing radius r, $\{(x_i(\tau), y_i(\tau)) : c < i \leq n, \tau \in \{0, \Delta, \ldots, (T_I - 1)\Delta\}\}$
Output: $\{(x_S(\tau), y_S(\tau)) : \tau \in \{0, \Delta, \ldots, (T_I - 1)\Delta\}\}$

for $\ell \in \{1, 2, \ldots, c\}$ **do**
  Set $S \leftarrow S_{\min} := \{j : c < j \leq n, d_j^+ = \min_{c<j'\leq n} d_{j'}^+\}$
  Set $s \leftarrow |S_{\min}|$
  while $s > 0$ **do**
    Set $(x_S(\tau), y_S(\tau)) \leftarrow \frac{1}{|S|} \sum_{i \in S} (x_i(\tau), y_i(\tau))$,
    $I_S \leftarrow \cup_{i \in S} \{(x, y) : (x - x_i(\tau))^2 + (y - y_i(\tau))^2 \leq r^2\}$
    if $(x_S(\tau), y_S(\tau)) \in I_S$: **then**
      Break
    if $(x_S(\tau), y_S(\tau)) \notin I_S$: **then**
      Set $i_S \leftarrow \mathcal{S}_s^{-1}(S)$
      if $i_S < \binom{|S_{\min}|}{s}$: **then**
        Set $S \leftarrow \mathcal{S}(i_S + 1)$
    else
      Set $s \leftarrow s - 1$
      Set $S \leftarrow \mathcal{S}_s(1)$
  end
  return $(x_S(\tau), y_S(\tau))$

Using the above notation, Algorithm 2 provides a pseudo-code for our motion-planning method, which we simulate in the next section to show how it effectively contributes to savings in the total communication energy by reducing the number of D2S transmissions required to ensure any given convergence rate.

VIII. Numerical Validation

We now perform numerical experiments to demonstrate that Algorithm 1 achieves substantial reductions in the total communication energy when compared to certain baselines, whereas Algorithm 2 enhances these energy reductions by further reducing the number of devices required to be sampled by the server in Algorithm 1. Moreover, these reductions are achieved without sacrificing testing accuracy.

A. Implementation

1) Network Architecture: We simulate a network consisting of $n = 70$ edge devices partitioned into $c = 7$ clusters with $n_\ell = 10$ nodes per cluster. In every global aggregation round, the digraph for each cluster $\ell \in [c]$ is constructed using the Random Direction Mobility Model (RDMM) [58] as follows: we assume that every client has a sensing radius[4] of $r = 15$ meters and that all the clients belonging to any given cluster are restricted to move within a square region having dimensions 45 m $\times$ 45 m. Each client performs $T_I = 20$ rounds of motion, with each new round starting with the client changing its direction of motion uniformly at random independently of other clients

[4]This means that the client detects the presence of another client in its vicinity and is able to establish a communication link with the latter if and only if the distance separating the clients is at most r.

and its own past directions of motion. In other words, if we use a 2-dimensional rectangular coordinate system to specify client locations within the cluster region, then regardless of our choice of the coordinate axes, the angle $\theta_i(\tau)$ made by the velocity vector of any client i with the x-axis in the τ-th round of motion is a random angle uniformly distributed over $\{0^\circ, 1^\circ, \ldots, 359^\circ\}$, and $\{\theta_i(\tau) : i \in [n]\}$ are independent random variables. An exception to the above update rule is encountered when the client hits any of the four boundaries of the cluster region during round $\tau - 1$, in which case $\theta_i(\tau)$ is chosen uniformly at random from the set of all whole number angles (measured in degrees) that enable the client to remain within the cluster region. For example, if the client hits a boundary parallel to the x-axis, then $\theta_i(\tau)$ is uniformly distributed over $\{0^\circ, 1^\circ, \ldots, 180^\circ\}$.

At the end of each round of motion, any two clients i and j that are separated by r or a distance smaller than r establish either of the two unidirectional links (i, j) or (j, i), each independently of the other with probability 0.5.

2) Datasets: All our simulations are performed on MNIST [59], Fashion-MNIST (F-MNIST) [60], and CIFAR-10 [61] datasets. The MNIST dataset consists of 70 K images (60K for training and 10 K for testing), and each image is a hand-written digit between 0 to 9 (i.e., the dataset has 10 labels). The same applies to the FMNIST dataset, the only difference being that it consists of images of fashion products. On the other hand, CIFAR-10 consists of 60 K images (50 K for training and 10 K for testing), and each image belongs to one of 10 classes (e.g., automobiles, frogs, etc.).

3) ML Models and Implementation: We use the neural network model from [3] along with one of its variants in our simulations. In particular, for all the experiments performed using the MNIST and FMNIST datasets we use a convolutional neural network (CNN) with two 5×5 convolution layers, the first of which has 32 channels and the second 64 channels, where each of these layers precedes a 2×2 max pooling, resulting in a total model dimension of 1,663,370. On the other hand, for all the experiments performed using CIFAR-10, we use a more complex CNN with four 5×5 convolution layers, which we obtain by adding two extra layers to the aforementioned two-layer CNN. We use the PyTorch implementation of this setup provided in [62] with cross-entropy loss. Each dataset is distributed among the clients in a non-i.i.d. manner: the samples (from either of the two datasets) are first sorted by their labels, partitioned into chunks of equal size, and each of the 70 clients is assigned only two chunks (i.e., each client will end up having only two labels). This results in extreme data heterogeneity, which leads to strong empirical guarantees for our approach.

All of our simulations are performed using the following hyperparameter values/ranges: $T = 5$, $t_{\max} \in \{15, 30\}$, and $\eta_t = 0.01(0.1)^t$ where t is the global aggregation index.

B. Results

We compare the energy vs. accuracy trade-offs associated with Algorithm 1 with those associated with two baselines, FedAvg [3] and collaborative relaying (COLREL) [12]. The second baseline is a recently proposed semi-decentralized FL algorithm that incorporates single-step averaging-based updates. Under the D2D and D2S connectivity constraints introduced in Section III, COLREL is a variant of FedAvg that incorporates one round of column-stochastic D2D aggregations before every global aggregation round but does not provide any criterion to control the sampling size m, which we assume to be fixed throughout its implementation. The fundamental difference between our method and COLREL is that our method takes into account the change in the connectivity of D2D clusters, optimally tuning the value of m according to the set of novel upper bounds on the singular values we obtained in Section VI.

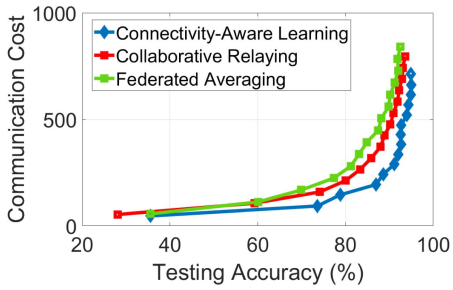

Fig. 5. Total communication energy/cost vs. testing accuracy under high D2S connectivity (Dataset: MNIST).

We consider these tradeoffs under different D2S connectivity levels. Intuitively speaking, on one hand, as the D2S connectivity improves, we expect to see that our algorithm leads to a lower energy and cost savings as compared to FedAvg. This is because our algorithm will naturally collapse to FedAvg and D2D communications will become less useful since more devices would engage in uplink communications, which by itself degrades the benefit of D2D local aggregations. On the other hand, as D2S connectivity improves, we expect to see that our algorithm achieves significant energy savings as compared to COLREL. This is because the impact of tuning m becomes more prominent when there is a possibility of D2S communications.

All of the following plots and discussion are based on the assumption that the ratio of the energy required for D2D communication to that of up-link (D2S) transmission, denoted by $\frac{E_{\text{D2D}}}{E_{\text{Glob}}}$, equals 0.1. This is a pessimistic estimate in favor of D2S considering that most ratios reported in the literature [11], [63], [64] take values less than 0.1. Thus, the communication costs reported are $(\#\text{D2S transmissions}) + 0.1 \times (\#\text{D2D transmissions})$.

1) Case 1. Cost Savings Under High D2S Connectivity: When the PS has a high downlink bandwidth and the connectivity between the devices and the PS is reliable, implementing FedAvg or COLREL has the effect of setting m to a value close to n. As an example, we implement FedAvg and COLREL with $m = 57$ and $m = 52$, respectively (note that COLREL requires fewer up-link transmissions because it uses D2D aggregations in addition to global aggregations). The results for MNIST are shown in Fig. 5: choosing $\phi_{\max} = 0.06$ results in Algorithm 1 achieving a testing accuracy of 90% while consuming about 46% less energy than FedAvg (thereby incurring proportionately lower communication costs). With respect to COLREL, the energy saving is even higher because COLREL also expends

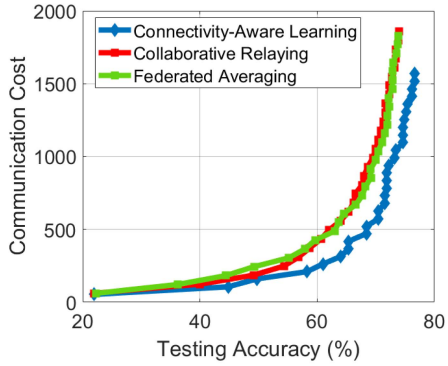

Fig. 6. Total communication energy/cost vs. testing accuracy under high D2S connectivity (Dataset: F-MNIST).

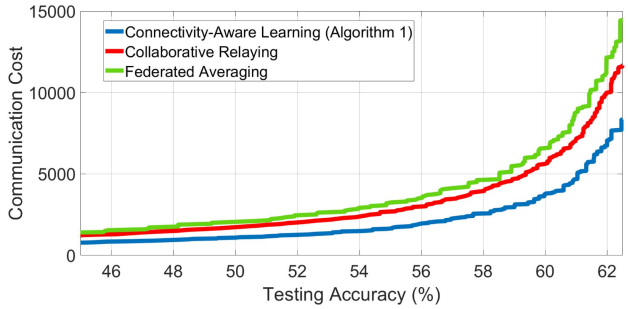

Fig. 7. Total communication energy/cost vs. testing accuracy under high D2S connectivity (Dataset: CIFAR-10).

energy on D2D aggregations with relatively little gain in testing accuracy.

Repeating a similar experiment on FMNIST results in a similar performance, depicted in Fig. 6. We see that Algorithm 1 (with $\phi_{\max} = 0.02$) consumes about 30% less energy than COLREL for achieving a testing accuracy of 70%. We also repeat this experiment on CIFAR-10 with the more complex four-layer CNN described in Section VIII-A. Here, we observe that Algorithm 1 (with $\phi_{\max} = 0.06$) consumes 40% less energy than COLREL for achieving a testing accuracy of 63%.

2) Case 2. Cost Savings Under Low D2S Connectivity: When the connectivity between the devices and the PS is poor, implementing FedAvg or COLREL has the effect of setting m to a value significantly smaller than n. As an example, we implement FedAvg and COLREL with $m = 26$ and $m = 15$, respectively. Choosing $\phi_{\max} = 0.2$ results in Algorithm 1 consuming about 30% less energy than FedAvg for achieving a testing accuracy of 90% on MNIST, as shown in Fig. 8. Moreover, the figure also shows that our algorithm does not compromise on the convergence rate and attains testing accuracy levels that are comparable to the baselines after each global aggregation. The cost saving is lower than in Case 1 as we expect because the singular value bounds incorporated by our algorithm into its choice of $m(t)$ are looser for higher values of the link failure probability p. Repeating a similar experiment on FMNIST results in a similar performance, depicted in Fig. 9. We also repeat this experiment on CIFAR-10 with the more complex four-layer CNN described in Section VIII-A. As shown in

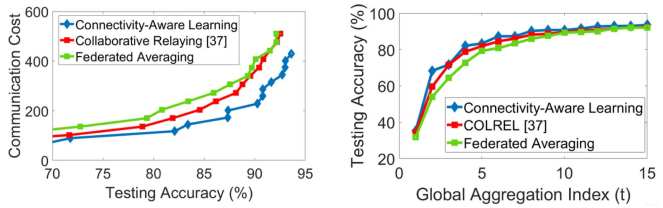

Fig. 8. Left-subplot: Total communication energy/cost vs. testing accuracy under low D2S connectivity (Dataset: MNIST). Right-subplot: The plot shows that our algorithm attains testing accuracy levels comparable to those of our baselines after each global aggregation.

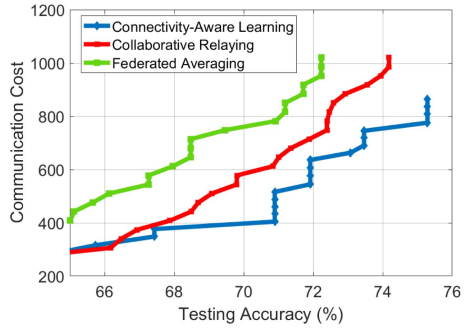

Fig. 9. Total communication energy/cost vs. testing accuracy under low D2S connectivity (Dataset: F-MNIST).

Fig. 7, we observe that Algorithm 1 (with $\phi_{\max} = 0.2$) consumes 30% less energy than COLREL for achieving a testing accuracy of 63%.

C. Motion-Planning Algorithm

Having shown how our connectivity-aware learning algorithm achieves significant energy savings by reducing the value of m, which denotes the number of D2S transmissions, we now show how Algorithm 2, our motion-planning method, significantly reduces the required value of m as determined by Step 11 of Algorithm 1.

Specifically, we assume that the trajectories of the U-nodes evolve according to the Random Direction Mobility Model as described in Section VIII-A, with each U-node pausing for a sojourn at the end of every round of motion. We then compute the required value of m using the expression provided in Step 11 of Algorithm 1. We perform 100 iterations of this procedure with $n = 140$ nodes and plot the average value of m against c, the number of control nodes, for different values of p_F, the control failure probability.

We observe from Fig. 11 that the value of m decreases as the number of control nodes increases, which implies that our motion-planning method contributes to energy savings by reducing the required number of energy-intensive D2S communications. We also observe that if 10% of the nodes are controlled, the value of m decreases by about 12%, and if 20% of them are controlled, the value of m decreases by about 14.6%. Moreover, we observe that our algorithm is robust to an increase in the probability of control failures.

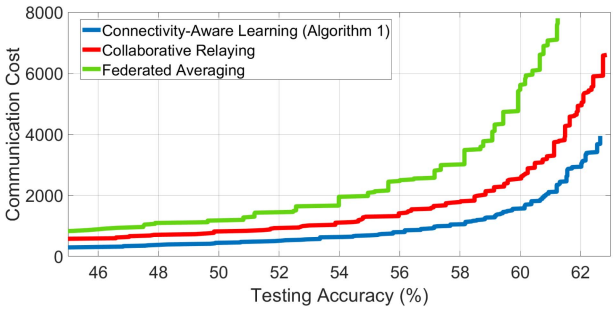

Fig. 10. Total communication energy/cost vs. testing accuracy under low D2S connectivity (Dataset: CIFAR-10).

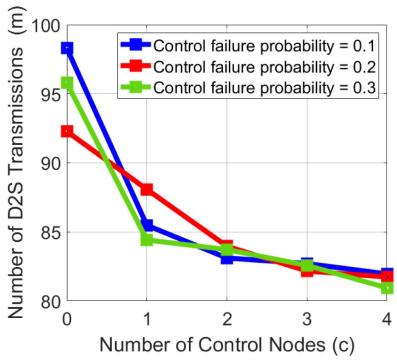

Fig. 11. Number of D2S transmissions m versus the number of control nodes c for a threshold of $\phi_{\max} = 0.02$ and $T_I = 5$ rounds of motion.

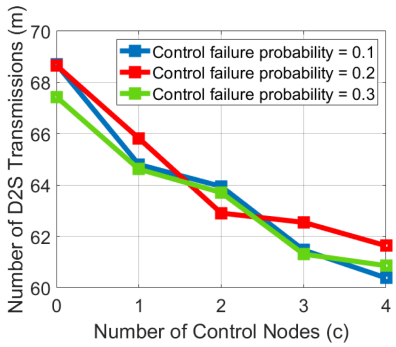

Fig. 12. Number of D2S transmissions m versus the number of control nodes c for a threshold of $\phi_{\max} = 0.02$ and $T_I = 10$ rounds of motion.

The reduction in m is slightly less remarkable when the value of T_I is increased from 5 to 10 (see Fig. 12). This may be because having greater rounds of motion result in the uncontrolled cluster digraph being less sparse and irregular, thereby creating less room for improvement in digraph regularity using our motion-planning of C-nodes. Nevertheless, we still reduce the number of D2S transmissions by 11% (respectively, 7%), by using only $c = 4$ (respectively, $c = 2$) C-nodes.

D. Scalability and Generality

We conclude this section with a few remarks on the scalability and generality of our proposed methods. We also comment briefly on the potential shortcomings of our approach.

First, note that neither the connectivity-aware algorithm (Algorithm 1) nor the motion-planning algorithm (Algorithm 2) imposes any restriction on c, the number of clusters in the D2D network, or on n_ℓ, the number of clients within any cluster. Moreover, the numerical values of the singular value bounds derived in Section VI are independent of n, c, and $\{n_\ell : \ell \in [n]\}$, because these bounds can be seen to depend only on the *ratios* of the node degrees and not on the absolute values of these node degrees. As such, we expect both algorithms are scalable with the number of devices.

Second, the computational steps that define both of our proposed algorithms do not impose restrictions on the dataset or the ML model employed. Also, our bounds are simple polynomials requiring low complexity computations. Hence, we expect both the algorithms to retain their energy-saving capabilities in other, real-world network settings. This is consistent with our simulation results, which show that Algorithm 1 continues to achieve energy saving levels superior to both the baselines even when we switch to a deeper CNN and a more complex dataset (namely, CIFAR-10). These results are plotted in Fig. 10.

On the contrary, the development of Algorithm 1 is based on theoretical analysis that is not intended for highly irregular D2D networks (i.e., communication digraphs in which the minimum out-degree is negligible compared to the maximum out-degree). This motivated our motion-planning algorithm (Algorithm 2), which uses client mobility to increase the regularity of the communication digraphs associated with each D2D cluster. Nevertheless, it is worth exploring whether Algorithm 1 can be enhanced by replacing the current singular value bounds in Step 10 with improved singular value bounds that can account for greater levels of digraph irregularity. The derivation of such improved bounds is likely to introduce significant technical difficulties arising from the heterogeneity of the edges inherent in irregular digraphs. To address these challenges, one possible starting point is to partition the network into multiple approximately regular digraphs, analyze each of these separately, and then examine the properties of the isoperimetric numbers of the resulting edge cuts.

Another potential drawback of Algorithms 1 and 2 is that they may not perform as well in applications involving sparse D2D networks. This shortcoming arises due to the fundamental limits on communication that are imposed by the lack of D2D links in sparse networks. In other words, it is not possible to design energy-efficient algorithms for sparse D2D networks by merely exploiting the graph-theoretic properties of such networks. One possible approach to address this is to partition the D2D network into a larger number of clusters so as to make each of the clusters small and dense (since smaller networks can achieve higher edge densities with smaller node degrees).

IX. Conclusion

We have investigated averaging-based semi-decentralized learning over clustered D2D networks modeled as time-varying digraphs. We first revealed the connection between the singular values of the column-stochastic matrices used for D2D model aggregations and the convergence rate of the learning process.

We then derived a set of upper bounds on these singular values in terms of the degree distributions of the cluster digraphs, and we used the resulting bounds to design a novel connectivity-aware FL algorithm that enables the central parameter server to tune the number of up-link transmissions by using its knowledge of the time-varying degree distributions of clusters. We also used the bounds to motivate a motion-planning algorithm that aids our former algorithm in further reducing the number of energy-intensive D2S transmissions.

Future works include obtaining upper bounds on singular values under more general assumptions, and obtaining optimal device sampling schemes for irregular clusters. An empirical assessment of our algorithm with the baselines functioning over distinct D2S connectivity regimes is also remained open.

ACKNOWLEDGMENT

The material includes the proofs of the theoretical results not proved in the main manuscript. Contact Rohit Parasnis (rohit100@mit.edu) for further questions about this work.

REFERENCES

[1] R. Parasnis, S. Hosseinalipour, Y.-W. Chu, M. Chiang, and C. G. Brinton, "Connectivity-aware semi-decentralized federated learning over time-varying D2D networks," in *Proc. 24th Int. Symp. Theory, Algorithmic Found., Protocol Des. Mobile Netw. Mobile Comput.*, 2023, pp. 31–40.

[2] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–82.

[4] A. Lalitha et al., "Fully decentralized federated learning," in *Proc. 3rd Workshop Bayesian Deep*, 2018.

[5] T. Sun, D. Li, and B. Wang, "Decentralized federated averaging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4289–4301, Apr. 2023.

[6] S. Zehtabi, S. Hosseinalipour, and C. G. Brinton, "Event-triggered decentralized federated learning over resource-constrained edge devices," 2022, *arXiv:2211.12640*.

[7] E. Beltrán et al., "Decentralized federated learning: Fundamentals, state-of-the-art, frameworks, trends, and challenges," *IEEE Commun. Surveys Tut.*, 2023.

[8] Y. Hua, K. Miller, A. L. Bertozzi, C. Qian, and B. Wang, "Efficient and reliable overlay networks for decentralized federated learning," *SIAM J. Appl. Math.*, vol. 82, no. 4, pp. 1558–1586, 2022.

[9] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[10] F. P.-C. Lin, S. Hosseinalipour, S. S. Azam, C. G. Brinton, and N. Michelusi, "Federated learning beyond the star: Local D2D model consensus with global cluster sampling," in *Proc. IEEE Glob. Commun. Conf.*, 2021, pp. 1–6.

[11] F. P.-C. Lin, S. Hosseinalipour, S. S. Azam, C. G. Brinton, and N. Michelusi, "Semi-decentralized federated learning with cooperative D2D local model aggregations," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3851–3869, Dec. 2021.

[12] M. Yemini, R. Saha, E. Ozfatura, D. Gündüz, and A. J. Goldsmith, "Semi-decentralized federated learning with collaborative relaying," in *Proc. IEEE Int. Symp. Inf. Theory*, 2022, pp. 1471–1476.

[13] S. Hosseinalipour et al., "Multi-stage hybrid federated learning over large-scale D2D-enabled fog networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 4, pp. 1569–1584, Aug. 2022.

[14] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-IID data," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2020, pp. 1–9.

[15] A. Bellet, A.-M. Kermarrec, and E. Lavoie, "D-cliques: Compensating for data heterogeneity with topology in decentralized federated learning," in *Proc. IEEE 41st Int. Symp. Reliable Distrib. Syst.*, 2022, pp. 1–11.

[16] Z. Wang, M. Zheng, J. Guo, and H. Huang, "Uncertain UAV ISR mission planning problem with multiple correlated objectives," *J. Intell. Fuzzy Syst.*, vol. 32, no. 1, pp. 321–335, 2017.

[17] D. Shen, G. Chen, J. B. Cruz, and E. Blasch, "A game theoretic data fusion aided path planning approach for cooperative UAV ISR," in *Proc. IEEE Aerosp. Conf.*, 2008, pp. 1–9.

[18] A. Koloskova, T. Lin, and S. U. Stich, "An improved analysis of gradient tracking for decentralized machine learning," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2021, pp. 11422–11435.

[19] H. Xing, O. Simeone, and S. Bi, "Federated learning over wireless device-to-device networks: Algorithms and convergence analysis," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3723–3741, Dec. 2021.

[20] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized SGD with changing topology and local updates," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5381–5393.

[21] A. Beznosikov, P. Dvurechensky, A. Koloskova, V. Samokhin, S. U. Stich, and A. Gasnikov, "Decentralized local stochastic extra-gradient for variational inequalities," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 38116–38133.

[22] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.

[23] Y. G. Kim and C.-J. Wu, "AutoFL: Enabling heterogeneity-aware energy efficient federated learning," in *Proc. IEEE/ACM 54th Annu. Int. Symp. Microarchitecture*, 2021, pp. 183–198.

[24] L. Li, D. Shi, R. Hou, H. Li, M. Pan, and Z. Han, "To talk or to work: Flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[25] D. Shi, L. Li, R. Chen, P. Prakash, M. Pan, and Y. Fang, "Toward energy-efficient federated learning over 5G+ mobile devices," *IEEE Wireless Commun.*, vol. 29, no. 5, pp. 44–51, Oct. 2022.

[26] T. Zhang and S. Mao, "Energy-efficient federated learning with intelligent reflecting surface," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 2, pp. 845–858, Jun. 2022.

[27] Q.-V. Pham,, M. Le, T. Huynh-The, Z. Han, and W. -J. Hwang, "Energy-efficient federated learning over UAV-enabled wireless powered communications," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4977–4990, May 2022.

[28] I. Varlamis et al., "Using Big Data and federated learning for generating energy efficiency recommendations," *Int. J. Data Sci. Analytics*, vol. 16, pp. 353–369, 2023.

[29] S. A. Khowaja, K. Dev, P. Khowaja, and P. Bellavista, "Toward energy-efficient distributed federated learning for 6G networks," *IEEE Wireless Commun.*, vol. 28, no. 6, pp. 34–40, Dec. 2021.

[30] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[31] Z. Zhao et al., "Towards efficient communications in federated learning: A contemporary survey," *J. Franklin Inst.*, vol. 360, pp. 8669–8703, 2023.

[32] M. S. Al-Abiad, M. Obeed, M. J. Hossain, and A. Chaaban, "Decentralized aggregation for energy-efficient federated learning via overlapped clustering and D2D communications," 2022, *arXiv:2206.02981*.

[33] B. Wang, J. Fang, H. Li, X. Yuan, and Q. Ling, "Confederated learning: Federated learning with decentralized edge servers," *IEEE Trans. Signal Process.*, vol. 71, pp. 248–263, 2023,.

[34] S. Liang and G. Yin, "Dual averaging push for distributed convex optimization over time-varying directed graph," *IEEE Trans. Autom. Control*, vol. 65, no. 4, pp. 1785–1791, Apr. 2020.

[35] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 315–320, Jul. 2018.

[36] M. Akbari, B. Gharesifard, and T. Linder, "Distributed online convex optimization on time-varying directed graphs," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 3, pp. 417–428, Sep. 2017.

[37] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.

[38] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 601–615, Mar. 2015.

[39] A. Nedić and A. Olshevsky, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 3936–3947, Dec. 2016.
[40] Z. Wang and H. Li, "Edge-based stochastic gradient algorithm for distributed optimization," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 1421–1430, Jul.–Sep. 2020.
[41] R. Xin, S. Pu, A. Nedić, and U. A. Khan, "A general framework for decentralized optimization with first-order methods," *Proc. IEEE*, vol. 108, no. 11, pp. 1869–1889, Nov. 2020.
[42] B. Gharesifard and J. Cortés, "Distributed strategies for generating weight-balanced and doubly stochastic digraphs," *Eur. J. Control*, vol. 18, no. 6, pp. 539–557, 2012.
[43] T. Charalambous, Y. Yuan, T. Yang, W. Pan, C. N. Hadjicostis, and M. Johansson, "Decentralised minimum-time average consensus in digraphs," in *Proc. IEEE 52nd Conf. Decis. Control*, 2013, pp. 2617–2622.
[44] T. Charalambous, C. N. Hadjicostis, and M. Johansson, "Distributed minimum-time weight balancing over digraphs," in *Proc. IEEE 6th Int. Symp. Commun., Control Signal Process.*, 2014, pp. 190–193.
[45] C.-S. Lee, N. Michelusi, and G. Scutari, "Topology-agnostic average consensus in sensor networks with limited data rate," in *Proc. IEEE 51st Asilomar Conf. Signals, Syst., Comput.*, 2017, pp. 553–557.
[46] A. I. Rikos and C. N. Hadjicostis, "Distributed integer weight balancing in the presence of time delays in directed graphs," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1300–1309, Sep. 2018.
[47] C.-S. Lee, N. Michelusi, and G. Scutari, "Finite rate distributed weight-balancing and average consensus over digraphs," *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4530–4545, Oct. 2021.
[48] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York, NY, USA: Cambridge Univ. Press, 2012.
[49] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, vol. 71. Philadelphia, PA, USA: SIAM, 2000.
[50] S. Friedland and R. Nabben, "On cheeger-type inequalities for weighted graphs," *J. Graph Theory*, vol. 41, no. 1, pp. 1–17, 2002.
[51] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.
[52] M. S. Lynn and W. P. Timlake, "Bounds for perron eigenvectors and subdominant eigenvalues of positive matrices," *Linear Algebra Appl.*, vol. 2, no. 2, pp. 143–152, 1969.
[53] S. Wang, S. Hosseinalipour, M. Gorlatova, C. G. Brinton, and M. Chiang, "UAV-assisted online machine learning over multi-tiered networks: A hierarchical nested personalized federated learning approach," *IEEE Trans. Netw. Serv. Manage.*, vol. 20, no. 2, pp. 1847–1865, Jun. 2023.
[54] C. Huang, R. Lu, and K.-K. R. Choo, "Vehicular fog computing: Architecture, use case, and security and forensic challenges," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 105–111, Nov. 2017.
[55] H. Gao, C. Liu, Y. Li, and X. Yang, "V2VR: Reliable hybrid-network-oriented V2V data transmission and routing considering RSUs and connectivity probability," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3533–3546, Jun. 2021.
[56] Y. Sun et al., "Adaptive learning-based task offloading for vehicular edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3061–3074, Apr. 2019.
[57] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.
[58] M. Waqas et al., "A comprehensive survey on mobility-aware D2D communications: Principles, practice and challenges," *IEEE Commun. Surv. Tut.*, vol. 22, no. 3, pp. 1863–1886, thirdquarter 2020.
[59] Y. LeCun, "The MNIST dataset of handwritten digits," Sep. 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/
[60] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[61] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.
[62] S. Ji, "A PyTorch implementation of federated learning," *Zenodo*, 2018.
[63] A. Zhang and X. Lin, "Security-aware and privacy-preserving D2D communications in 5G," *IEEE Netw.*, vol. 31, no. 4, pp. 70–77, Jul./Aug. 2017.
[64] M. Hmila, M. Fernández-Veiga, M. Rodríguez-Pérez, and S. Herrería-Alonso, "Energy efficient power and channel allocation in underlay device to multi device communications," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5817–5832, Aug. 2019.