IEEE INFOCOM 2024 - IEEE Conference on Computer Communications | 979-8-3503-8350-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/INFOCOM52122.2024.10621133

Taming Subnet-Drift in D2D-Enabled Fog Learning: A Hierarchical Gradient Tracking Approach

Evan Chen*, Shiqiang Wang†, and Christopher G. Brinton*
*School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907
†IBM T. J. Watson Research Center, Yorktown Heights, NY 10598
Email: {chen4388, cgb}@purdue.edu, wangshiq@us.ibm.com

***Abstract*—Federated learning (FL) encounters scalability challenges when implemented over fog networks. Semi-decentralized FL (SD-FL) proposes a solution that divides model cooperation into two stages: at the lower stage, device-to-device (D2D) communications is employed for local model aggregations within subnetworks (subnets), while the upper stage handles device-server (DS) communications for global model aggregations. However, existing SD-FL schemes are based on gradient diversity assumptions that become performance bottlenecks as data distributions become more heterogeneous. In this work, we develop semi-decentralized gradient tracking (SD-GT), the first SD-FL methodology that removes the need for such assumptions by incorporating tracking terms into device updates for each communication layer. Analytical characterization of SD-GT reveals convergence upper bounds for both non-convex and strongly-convex problems, for a suitable choice of step size. We employ the resulting bounds in the development of a co-optimization algorithm for optimizing subnet sampling rates and D2D rounds according to a performance-efficiency trade-off. Our subsequent numerical evaluations demonstrate that SD-GT obtains substantial improvements in trained model quality and communication cost relative to baselines in SD-FL and gradient tracking on several datasets.**

***Index Terms*—Fog Learning, Semi-decentralized FL, Device-to-device (D2D) communications, Federated Learning, Gradient Tracking, Communication Efficiency**

I. Introduction

Federated learning (FL) has emerged as a promising technique for distributed machine learning (ML) over networked systems [1], [2]. FL aims to solve problems of this form:

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

$$\text{where } f_i(x) = \mathbb{E}_{\xi_i\sim\mathcal{D}_i} f_i(x;\xi_i), \tag{2}$$

where n is the total number of clients (typically edge devices) in the system, $f_i(x)$ is the local ML loss function computed at client i for model parameters $x \in \mathbb{R}^d$, $\mathcal{D}_i$ is the local data distribution at client i, and ξ_i is a random sample from $\mathcal{D}_i$.

Conventionally, FL employs a two-step iterative algorithm to solve this optimization: (i) *local model update*, where gradient information computed on the local device dataset is used to update the local model, and (ii) *global model aggregation*, where a central server forms a consensus model across all devices. In wireless networks, however, device-server (DS) communications for the global aggregation step can be expensive, especially for large ML models over long DS distances. Much research in FL has been devoted to improving this communication efficiency, with typical approaches including model sparsification/quantization [3]–[5], device sampling [6], and aggregation frequency minimization [7], [8].

This work was supported by the National Science Foundation (NSF) under grants CPS-2313109 and CNS-2212565, by DARPA under grant D22AP00168, and by the Office of Naval Research (ONR) under grant N000142212305.

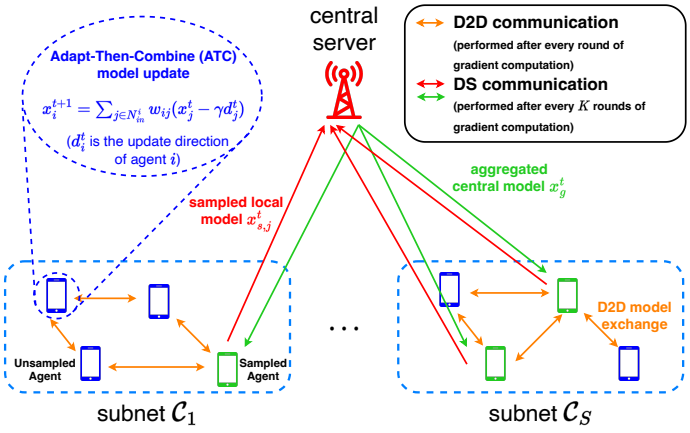

Fig. 1: Illustration of semi-decentralized FL. Clients in each subnet communicate via iterative low-cost D2D communications to conduct local aggregations. Once they have converged towards a consensus within the subnet, the central server conducts a global aggregation across sampled devices using DS communication.

Recent research has considered how decentralizing FL's client-server star topology can improve communication overhead, e.g., by introducing more localized communications wherever possible. In the extreme case of severless FL, model aggregations are conducted entirely through short range device-to-device (D2D) communications [9]–[11]. More generally, *fog learning* [12]–[14] considers distributing FL over fog computing architectures, where a hierarchy of network elements between the edge and cloud enables horizontal (i.e., intra-layer) and vertical (i.e., inter-layer) communications.

A. Semi-Decentralized FL and Subnet-Drift

Semi-decentralized FL (SD-FL) has emerged as an important implementation of fog learning [15], [16]. Its overall architecture is depicted in Fig. 1. Devices are grouped into subnetworks (subnets) of close physical proximity, according to their ability to form D2D connections. For example, con-

sider a set of 5G mobile devices in a cell aiming to learn an ML model: peer relationships can establish D2D-enabled subnets, with the main server located at the base station [17].

To enhance communication efficiency, the model aggregation in SD-FL is conducted in two stages: (i) *iterative cooperative consensus formation* of local models within subnets, and (ii) *DS communication among sampled devices* for global aggregation. The idea is that frequent, low-cost in-subnet model aggregations should reduce the burden placed on global, cross-subnet aggregations, as they can occur less frequently engaging fewer clients. However, a fundamental challenge in SD-FL is *managing gradient diversity across subnets:* the more local aggregations we perform, the more the global model drifts away from the global optimum towards a linear combination of local optimums of each subnet. This "subnet-drift" manifests from the client-drift problem in FL, due to non-i.i.d. local datasets across clients [7], [8], [18].

In this work, we are interested in addressing the subnet-drift challenge for SD-FL. Although some existing works alleviate client drift by letting clients share a portion of their datasets with their neighbors and/or the server [6], [19], [20], such approaches present privacy issues that FL aims to avoid. As a result, we turn to concepts in *gradient tracking*, which have been successful in mitigating data heterogeneity challenges in fully decentralized learning and do not require data sharing [9], [10]. In this respect, the hierarchical nature of SD-FL presents two key research challenges. First, the differing timescales of D2D and DS communications need careful consideration on how a client should employ gradient information from the server versus from its neighbors. Second, randomness in client participation for DS communication may create bias in aggregated gradient information. We thus pose the following question:

How do we alleviate subnet drift in semi-decentralized FL through gradient tracking while ensuring gradient information is well mixed throughout the system?

To address this, a key component of our design is to consider two separate gradient tracking terms, one for each communication stage of SD-FL, which treat the incoming information differently. Our convergence analysis and subsequent experiments demonstrate how this stabilizes the global learning process.

B. Outline and Summary of Contributions

- We propose Semi-Decentralized Gradient Tracking (SD-GT), the first work to integrate gradient tracking into SD-FL which is robust to data heterogeneity. SD-GT can tolerate a large number of D2D communications between two global aggregation rounds without risking convergence to a sub-optimal solution (Sec. III).
- We conduct a Lyapunov-based convergence analysis, showing convergence upper bounds for both non-convex and strongly convex functions. We employ these results in a co-optimization of convergence speed and communication efficiency based on the D2D communication rounds and subnet sampling rates (Sec. IV).
- Our experiments verify that SD-GT obtains substantial improvements in trained model quality and convergence speed relative to baselines in SD-FL and gradient tracking. Moreover, we verify the behavior of our co-optimization optimization in adapting to the relative cost of D2D vs. DS communications (Sec. V).

II. Related Works

Hierarchical FL. Hierarchical FL has received considerable attention for scaling up model training across large numbers of edge devices. Most of this work has considered a multi-stage tree extension of FL [14], [21]–[23], i.e., with each stage forming its own star topology for local aggregations. A commonly considered use case has been the three-tier hierarchy involving device, base station, and cloud encountered in cellular networks. Optimized hierarchical aggregations have demonstrated significant improvements in convergence speed and/or communication efficiency. In a separate domain, these concepts have been employed for model personalization in cross-silo FL [24].

Our work focuses specifically on semi-decentralized FL, where edge subnets conduct local aggregations via D2D-enabled cooperative consensus formation [15], [16], [25]. SD-FL is intended for settings where DS communications are costly, e.g., due to long distances. The authors of [15] were the first to formally study the convergence behavior of SD-FL, wherein they proposed a control algorithm to maintain convergence based on approximations of data-related parameters. [25] developed SD-FL based on more general models of subnet topologies that may be time-varying and directed. A main issue with all current SD-FL papers is that they assume that either the gradient, gradient diversity, or data-heterogeneity are bounded, while [15] and [25] even require knowledge on the connectivity of each subnet. In our work, we are able to remove the requirement of any knowledge on data distribution or subnet topology and still guarantee convergence.

Gradient tracking for communication efficiency. Gradient tracking (GT) methods [26]–[30] were proposed to mitigate data heterogeneity in decentralized optimization algorithms. The main idea is to track the gradient information from neighbors every time a communication is performed. GT has become particularly popular in settings where communication costs are high, as it enables algorithms to reach the optimum point using a large number of local updates and minimal communications [7], [8], [18], [31]–[34]. Assumptions on data heterogeneity can be lifted under proper initialization of gradient tracking variables.

Our work instead considers GT under a semi-decentralized network setting. In this respect, [35] discussed GT under a hierarchical network structure, where they assumed: (i) random edge activation within subgraphs and (ii) all subgraphs being connected by a higher layer graph that communicates after every gradient update. However, the hierarchical structure they consider is different from SD-FL, where in our setting D2D communication usually is cheaper than DS communication and thus occurs at a much higher frequency. In this paper, we

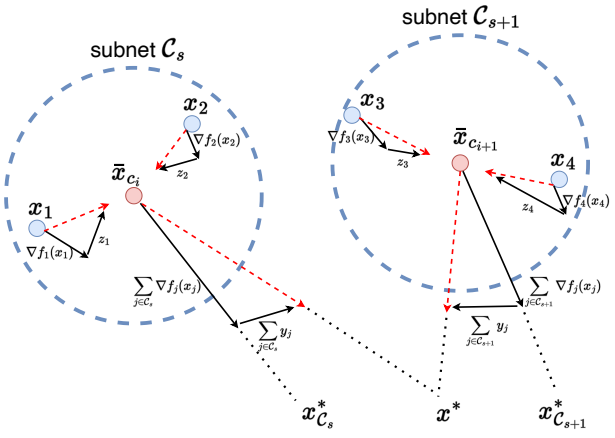

Fig. 2: An illustration of how SD-GT deals with subnet-drifting. With the introduction of in-subnet GT term z_i^t, all clients within each subnet are able to converge towards a consensual location of the subnet. And the inter-subnet GT term y_i^t corrects the update direction of the whole subnet so that it no longer converges towards the optimal solution $x^*_{\mathcal{C}_s}$ of the subnet $\mathcal{C}_s$ but the optimal solution x^* of the whole network.

develop a GT methodology that is aware of the diversity in information mixing speeds between D2D and DS, and track this difference by maintaining two separate GT terms.

III. Proposed Method

In this section, we first introduce the overall network structure of SD-FL (Sec. III-A). Then we develop our SD-GT algorithm, explaining the usage of each tracking variable and how they solve the subnet-drift problem (Sec. III-B). Finally, we show that our method encapsulates two existing methods under specific network topologies (Sec. III-C).

A. Network Model and Timescales

We consider a network containing a central server connected upstream from n clients (edge devices), indexed $i = 1, ..., n$. As shown in Figure 1, the devices are partitioned into S disjoint subnets $\mathcal{C}_1, \ldots, \mathcal{C}_S$. Subnet s contains $m_s = |\mathcal{C}_s|$ clients, where $\sum_{s=1}^{S} m_s = n$. Similar to existing works in SD-FL [15], [16], we do not presume any particular mechanism by which clients have been grouped into subnets, except that clients within the same subnet are capable of engaging in D2D communications according to a wireless protocol, e.g., devices within a 5G cell.

For every client $i \in \mathcal{C}_s$, we let $\mathcal{N}_i^{\text{in}} \subseteq \mathcal{C}_s$ be the set of input neighbors for D2D transmissions. Considering all clients $i, j \in \mathcal{C}_s$, we define $W_s = [w_{ij}] \in \mathbb{R}^{m_s \times m_s}$ to be the D2D communication matrix for subnet s, where $0 < w_{ij} \leq 1$ if $j \in \mathcal{N}_i^{\text{in}}$, and $w_{ij} = 0$ otherwise. As we will see in Sec. III-B, w_{ij} is the weight that client i will apply to information received from client j. We then can define the network-wide D2D matrix $W = \text{diag}(W_1, \ldots, W_S) \in \mathbb{R}^{n \times n}$, which is block-diagonal given that the subnets do not directly communicate. In Sec. IV-A, we will discuss further assumptions on the subnet matrices W_s for our convergence analysis.

The SD-GT training process consists of two timescales. The outer timescale, $t = 1, 2, \ldots, T$, indexes global aggregations carried out through DS communications. The inner timescale, $k = 1, \ldots, K$, indexes local training and aggregation rounds carried out via D2D communications. We assume a constant K local rounds occur between consecutive global aggregations.

B. Learning Model

As shown in Algorithm 1, each client maintains two gradient tracking terms, y_i^t and z_i^t, which track (i) the gradient information between different subnets and (ii) the gradient information inside each subnet, respectively. These two variables act as corrections to the local gradients so that the update direction can guarantee convergence towards global optimum as in Figure 2.

In-subnet Updates. We denote $x_i^{t,k}$ as the ML parameter vector stored at client i during the t^{th} global aggregation round and k^{th} D2D communication round. The local model updates are performed by first updating its local model, and then making a linear combination with models received from its neighbors, also known the Adapt-Then-Combine (ATC) scheme. ATC is known to have a better performance compared to other mixing schemes [36]. The update direction not only includes the gradient direction computed from the local function $\nabla f_i(\cdot)$ but also the gradient tracking terms z_i^t and y_i^t:

$$x_i^{t,k+\frac{1}{2}} = x_i^{t,k} - \gamma \left(\nabla f_i(x_i^{t,k}, \xi_i^{t,k}) + y_i^t + z_i^t \right), \quad \forall i, \quad (3)$$

$$x_i^{t,k+1} = \textstyle\sum_{j \in \mathcal{N}_i^{\text{in}} \cup \{i\}} w_{ij} x_j^{t,k+\frac{1}{2}}, \quad \forall i, \quad (4)$$

where $\nabla f_i(x_i^{t,k}, \xi_i^{t,k})$ denotes the stochastic gradient of $\nabla f_i(x_i^{t,k})$. After the K rounds of D2D communications, each client updates its own in-subnet gradient tracking term z_i^t using the measured difference of the variable $x_i^{t,k}$ every time a D2D communication within the subnet is performed:

$$z_i^{t,k} = x_i^{t,k+\frac{1}{2}} - x_i^{t,k} + \gamma y_i^t, \quad \forall i, \quad (5)$$

$$z_i^{t+1} = z_i^t + \tfrac{1}{K\gamma} \textstyle\sum_{k=1}^{K} (z_i^{t,k} - \sum_{j \in \mathcal{N}_i^{\text{in}} \cup \{i\}} w_{ij} z_j^{t,k}), \quad \forall i. \quad (6)$$

Global Aggregation. The central server is able to choose the total number of clients $h_s \in \{1, ..., m_s\}$ to sample from each subnet s, e.g., based the DS communication budget. These will be variables in our optimization considered in Sec. IV-C. Those clients that are not sampled by the central server for round t will not update their parameters, and maintain x_i^t and y_i^t into the next communication round.

We denote x_{g}^t as the global model that is stored at the server. At each global aggregation t, the server accumulates the gradient information collected from each subnet, and updates the inter-subnet gradient tracking terms that are stored on the server, which we denote ψ_s^t for subnet s.

Now, for each subnet s, let the sampled clients for round t be indexed by $j = 1, \ldots, h_s$. Before uploading the information to the server, each client conducts a cancellation of the inter-subnet gradient tracking information so that the global model receives unbiased gradient information. Formally, we have

Algorithm 1: SD-GT: Semi-Decentralized Gradient Tracking

Constants: step size $\gamma > 0$, initial model parameter x^0
Output: x_{g}^t

Local parameter initialization
$x_i^{1,1} \leftarrow x^0, \quad \forall i$
$y_1^1 = y_2^1 = \cdots = y_n^1 \leftarrow 0, \quad \forall i$
$z_1^1 = z_2^1 = \cdots = z_n^1 \leftarrow 0, \quad \forall i$
Server parameter initialization
$x_{\mathrm{g}}^1 \leftarrow x^0$
$\psi_s^1 \leftarrow 0, \quad \forall s$
for $t \leftarrow 1, \ldots, T$ **do**
  /* Step 1: In-Subnet Model Update */
  for $k \leftarrow 1, \ldots, K$ **do**
    each client $i \leftarrow 1, \ldots, n$ **in parallel:**
      Perform in-subnet update on the local model $x_i^{t,k}$ based on (3) and (4).
  /* Step 2: Local Tracking Term Update */
  each client $i \leftarrow 1, \cdots, n$ **in parallel:**
    Update the in-subnet gradient tracking variable z_i^t based on (5) and (6).
  /* Step 3: Global Aggregation */
  sample h_s random clients $x_{s,j} \sim \mathcal{C}_s$ from every subnet
  each subnet $s \leftarrow 1, \ldots, S$ **in parallel:**
    all sampled clients $j \leftarrow 1, \ldots, h_s$ **aggregate:**
      Use (7) to (10) to update variables x_{g}^t and ψ_s^t, $\forall s$ on the server, then broadcast them to all sampled clients.
  All clients that are not sampled:
  $x_i^{t+1,1} \leftarrow x_i^{t,K+1}$
  $y_i^{t+1} \leftarrow y_i^t$

the following sequence of updates, based on intermediate quantities $\tilde{x}_{s,j}^{t+1}$ at the client-side and $\tilde{x}_{\mathrm{g}}^{t+1}$ at server-side:

$$\tilde{x}_{s,j}^{t+1} = x_{s,j}^{t,K+1} - x_{s,j}^{t,1} + K\gamma y_{s,j}^t, \qquad \forall j, \forall s, \quad (7)$$

$$\tilde{x}_{\mathrm{g}}^{t+1} = \tfrac{1}{S \cdot h_s} \textstyle\sum_{s=1}^{S} \sum_{j=1}^{h_s} \tilde{x}_{s,j}^{t+1}, \quad (8)$$

$$x_{\mathrm{g}}^{t+1} = x_{\mathrm{g}}^t + \tilde{x}_{\mathrm{g}}^{t+1}, \quad (9)$$

$$\psi_s^{t+1} = \tfrac{1}{K\gamma}(\tfrac{1}{h_s} \textstyle\sum_{j=1}^{h_s} \tilde{x}_{s,j}^{t+1} - \tilde{x}_{\mathrm{g}}^{t+1}), \qquad \forall s. \quad (10)$$

(7) is a client-side computation, while (8)-(10) occur at the server. Finally, the server broadcasts the updated global model x_{g}^{t+1} and inter-subnet gradient tracking terms y_s^{t+1} to the sampled clients to complete the synchronization:

$$x_{s,j}^{t,1} = x_{\mathrm{g}}^{t+1}, \quad \forall j,$$
$$y_{s,j}^{t+1} = \psi_s^{t+1}, \quad \forall j.$$

Maintaining two gradient tracking terms is an essential feature of SD-GT for stabilizing convergence. In particular, if we only used the between-subnet measure y_i^t to track gradient information, then the gradient information within each subnet would deviate from the average of the subnet, preventing the system from converging towards the global minimum. On the other hand, if we only used the within-subnet measure z_i^t to track gradient information, then each subnet s will tend to converge towards its local minimum x_s^* instead.

C. Connection with Existing Methods

Some existing methods can be shown to be special cases of our algorithm under certain network structures.

Case 1: Every client is its own subnet ($S = n$). Under this setting, the server always samples the full subnet since each subnet contains only one client. Then we can see that with the initialization $z_1^1 = z_2^1 = \ldots = z_n^1 = 0$, the in-subnet gradient tracking terms are always zero:

$$z_i^{t+1} = z_i^t + \tfrac{1}{K\gamma} \textstyle\sum_{k=1}^{K} (\tilde{z}_i^{t,k} - \tilde{z}_i^{t,k}) = z_i^t = 0, \quad \forall i.$$

The global gradient tracking term $y_{s,j}^t$ can be formulated as:

$$\begin{aligned} y_{s,j}^{t+1} = \psi_s^{t+1} &= \tfrac{1}{K\gamma}(\textstyle\sum_{j=1}^{h_s} \tfrac{1}{h_s} \tilde{x}_{s,j}^{t+1} - \tilde{x}_{\mathrm{g}}^{t+1}) \\ &= y_{s,j}^t + \tfrac{1}{K\gamma}(x_s^{t,K+1} - x_{\mathrm{g}}^{t+1}), \quad \forall s. \end{aligned}$$

With the in-subnet gradient tracking term being zero and the global gradient tracking term in the form above, our algorithm aligns with Proxskip [8] under this setting.

Case 2: Single subnet with one D2D communication round ($S = 1, K = 1$). Under this setting, the global gradient tracking term is always zero since $\sum_{j=1}^{h_s} \frac{1}{h_s} \tilde{x}_{s,j}^{t+1} = \tilde{x}_{\mathrm{g}}^{t+1}$:

$$\psi_s^{t+1} = \psi_s^t + \tfrac{1}{\gamma}(\textstyle\sum_{j=1}^{h_s} \tfrac{1}{h_c} \tilde{x}_{s,j}^{t+1} - \tilde{x}_{\mathrm{g}}^{t+1}) = y_s^t = 0, \quad \forall s.$$

Since the D2D rounds are set to one, the update of z_i^t for every client i can be formulated as:

$$\begin{aligned} z_i^{t+1} =& z_i^t + \frac{1}{K\gamma} \sum_{k=1}^{K} (\tilde{z}_i^{t,k} - \sum_{j \in \mathcal{N}_i^{\mathrm{in}} \cup \{i\}} w_{ij} \tilde{z}_j^{t,k}) \\ =& \textstyle\sum_{j \in \mathcal{N}_i^{\mathrm{in}} \cup \{i\}} w_{ij}(z_j^t + \nabla f_j(x_j^t)) - \nabla f_i(x_i^t), \quad \forall i. \end{aligned}$$

If we define $\hat{z}_i^t = z_i^t + \nabla f_i(x_i^t)$, we are left with:

$$x_i^{t+1} = \textstyle\sum_{j \in \mathcal{N}_i^{\mathrm{in}} \cup \{i\}} w_{ij}(x_j^t - \gamma \hat{z}_i^t), \qquad \forall i,$$
$$\hat{z}_i^{t+1} = \sum_{j \in \mathcal{N}_i^{\mathrm{in}} \cup \{i\}} w_{ij} \hat{z}_j^t + \nabla f_i(x_i^{t+1}) - \nabla f_i(x_i^t), \quad \forall i,$$

which aligns with the gradient tracking algorithm [26], [37].

IV. Analysis and Optimization

We first show the assumptions used for the proof (Sec. IV-A), then provide the convergence analysis under non- and strong-convexity (Sec. IV-B). Finally, we derive a co-optimization algorithm that considers the trade-off between communication cost and performance (Sec. IV-C). We defer the proofs of supporting lemmas to the appendices.

A. Convergence Analysis Assumptions

The first three assumptions established are general assumptions [3], [8], [18] that are applied to both theorems in this paper, while the last assumption is a stricter condition [28] that guarantees a better convergence rate in Theorem 2.

Assumption 1. ***(L-smooth)*** *Each local objective function f_i: $\mathbb{R}^d \to \mathbb{R}$ is L-smooth:*

$$\|\nabla f_i(x) - \nabla f_i(y)\|_2 \leq L\|y - x\|_2, \quad \forall x, y \in \mathbb{R}^d.$$

Assumption 2. ***(Mixing Rate)*** *Each subnet $\mathcal{C}_1, \ldots, \mathcal{C}_S$ has a strongly connected graph, with doubly stochastic weight*

matrix $W_s \in \mathbb{R}^{m_s \times m_s}$. *Define* $J_s = \frac{\mathbf{1}\mathbf{1}^\mathbf{T}}{m_s}$, *then there exists a constant* $\rho_s \in (0,1]$ *s.t.*

$$\|X(W_s - J_s)\|_F^2 \leq (1-\rho_s)\|X(I-J_s)\|_F^2, \quad \forall X \in \mathbb{R}^{d \times m_s}.$$

Assumption 3. ***(Bounded Variance)*** *Variances of each client's stochastic gradients are uniformly bounded.*

$$\mathbb{E}_{\xi \sim \mathcal{D}_i}\|\nabla f_i(x;\xi) - \nabla f_i(x)\|_2^2 \leq \sigma^2, \forall i \in [1,n], \forall x \in \mathbb{R}^d.$$

Assumption 4. ***(μ-strongly-convex)*** *Every local objective function* $f_i : \mathbb{R}^d \to \mathbb{R}$ *is* μ*-strongly convex with* $0 < \mu \leq L$.

$$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \frac{\mu}{2}\|y-x\|^2, \quad \forall x, y \in \mathbb{R}^d.$$

B. Convergence Analysis Results

Here we provide the theoretical bounds for our algorithm in both non-convex and strongly convex settings without assumptions on data-heterogeneity. We define $\overline{x}_g^t = \mathbb{E}_{h_S}[x_g^t]$ to be the server model's expectation over the sample sets h_S.

Theorem 1. ***(Non-convex)*** *Under Assumptions 1, 2, and 3. Let* $\beta_s = \frac{m_s - h_s}{m_s}$ *be the ratio of unsampled clients from each subnet, and define the* ***sample-wise mixing rate*** *term* $\phi_s = (1-\beta_s^2)$. *Define* $p = \min(\phi_1, \ldots, \phi_S) \in (0,1]$, *and* $q = \min(\rho_1, \ldots, \rho_S) \in (0,1]$. *With a choice of constant step size* $\gamma < \mathcal{O}(\frac{p^2 q}{KL})$, *we have the following convergence rate:*

$$\frac{1}{T}\sum_{t=1}^T \mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^t)\|^2 \leq \mathcal{O}\left(\frac{\mathcal{H}^1}{TK\gamma} + \frac{LK\gamma^2\sigma^2}{2} + \frac{L^2K^2\gamma^3\sigma^2}{p^4q^2}\right). \tag{11}$$

Proof. If we define a Lyapunov function $\mathcal{H}^t = \mathbb{E}f(\overline{x}_{\mathrm{g}}^t) - \mathbb{E}f(x^*) + c_0K^3\gamma^3\left(\frac{1}{p}Y^t + \frac{1}{q}Z^t\right) + c_1\frac{K\gamma}{p}\Gamma^t$ and also combine Lemma 4 with a constant c_2 (see definition of Γ^t, Y^t, Z^t, and Δ^t in the appendix), we have:

$$\begin{aligned} 0 \leq & c_2\gamma L^2\Bigg(-\Delta^t + 3\rho_m^2 K \tfrac{1}{n}\textstyle\sum_{i=1}^n \mathbb{E}\|x_i^{t-1,K+1} - \overline{x}_{\mathrm{g}}^t\|^2 \\ & + 6K^3\gamma^2(4Y^t + 4Z^t + \mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^t)\|^2) + 3K^2\gamma^2\sigma^2\Bigg). \end{aligned}$$

Choose the constants as the following: $c_2 > 2$, $c_1 = 6L^2c_2$, $c_0 = \frac{1152}{p^2}L^2c_2$, $\gamma < \frac{p^2q}{945KL}$. Then by combining Lemmas 2, 4, 5, 6, 7, and with values $D_1, D_2, D_3, D_4, D_5 \geq 0$ and $D > 0$, we can obtain the following recursive form:

$$\begin{aligned} \mathcal{H}^t - \mathcal{H}^{t-1} \leq & -DK\gamma\mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^{t-1})\|^2 \\ & - D_1Y^{t-1} - D_2Z^{t-1} - D_3\Gamma^{t-1} - D_4\Delta^{t-1} \\ & + \tfrac{D_5L^2}{p^4q^2\cdot K}(K^3\gamma^3)\sigma^2 + \tfrac{L}{2K}(K^2\gamma^2)\sigma^2 \\ \leq & -DK\gamma\mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^{t-1})\|^2 \\ & + \tfrac{D_5L^2}{p^4q^2\cdot K}(K^3\gamma^3)\sigma^2 + \tfrac{L}{2K}(K^2\gamma^2)\sigma^2. \end{aligned}$$

We can then unpack the recursion of the Lyapunov function and get the final rate. □

In Theorem 1, p implies the sampling rate of the server and q implies the information mixing ability of D2D communications. Large p indicates the server samples a large amount of clients and a large q indicates the connectivity of every subnet is high. By carefully choosing the step size, we can achieve the following result:

Corollary 1. *Under the same conditions as in Theorem 1, there exists a constant step size such that:*

$$\frac{1}{T}\sum_{t=1}^T \mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^t)\|^2 = \mathcal{O}\left(\sqrt{\tfrac{\mathcal{H}^1\sigma^2L}{TK}} + (\tfrac{\mathcal{H}^1L\sigma}{KTp^2q})^{\frac{2}{3}} + \tfrac{\mathcal{H}^1L}{Tp^4q^2}\right). \tag{12}$$

Proof. Follows from plugging Lemma 1 into Theorem 1. □

From observing (12), the first two terms capture the effect of stochastic gradient variance on convergence. Choosing a larger K decreases the effect of stochasticity on the convergence. We can also observe that the bound will be better when the values of p and q are large. A key observation is that the network topology that gives the tightest bound is under $p = q = 1$, which means the every time DS communication is performed, the server samples all clients and the topology of every in-subnet D2D communication is a fully connected graph. By adding a stronger assumption on strong-convexity, we are able to get a stronger convergence result.

Theorem 2. ***(Strongly-convex)*** *Under Assumptions 1, 2,3, and 4. Define the Lyapunov function* $\mathcal{L}^t = \mathbb{E}\|\overline{x}_{\mathrm{g}}^t - x^*\|^2 + \Gamma^t + \gamma Y^t + \gamma Z^t$ *(see definition of* Γ^t, Y^t, Z^t *in the appendix), then for a sufficiently small constant step size* γ, *we have*

$$\mathbb{E}\|\overline{x}_{\mathrm{g}}^{T+1} - x^*\|^2 \leq (1 - \tfrac{\mu K\gamma}{4})^T\mathcal{L}^1 + \|(I-A)^{-1}b\|_1, \tag{13}$$

where I *is the identity matrix,*

$$A = \left(1 - \frac{p}{2}\right)I + \gamma KL\begin{bmatrix} \frac{p-\mu\gamma K}{2\gamma KL} & 9(1-p) & 72K^2\gamma & 72K^2\gamma \\ \frac{14\gamma KL}{p} & \frac{36\gamma L}{p} & \frac{14K}{pL} & \frac{14K}{pL} \\ \frac{72\gamma^2K^2L^3}{p} & \frac{30L}{p} & \frac{240K^2\gamma L}{p} & \frac{240K^2\gamma L}{p} \\ \frac{168\gamma^2K^2L^3}{q} & \frac{78L}{q} & \frac{624K^2\gamma L}{q} & \frac{624K^2\gamma L}{q} \end{bmatrix},$$

$$b = \begin{bmatrix} \frac{2\gamma^2K}{n} + 9K^2\gamma^3L \\ K\gamma^2 + 3K^3\gamma^4L^2 \\ \frac{2\gamma}{qK} + \frac{30K^3\gamma^3L^2}{q} \\ \frac{2\gamma}{qK} + \frac{78K^3\gamma^3L^2}{q} \end{bmatrix}\sigma^2.$$

Proof. By combining Lemmas 3, 4, 5, 6, 7, we can form the recursive form:

$$\begin{bmatrix} \mathbb{E}\|\overline{x}_{\mathrm{g}}^t - x^*\|^2 \\ \Gamma^t \\ \gamma Y^t \\ \gamma Z^t \end{bmatrix} \leq A\begin{bmatrix} \mathbb{E}\|\overline{x}_{\mathrm{g}}^{t-1} - x^*\|^2 \\ \Gamma^{t-1} \\ \gamma Y^{t-1} \\ \gamma Z^{t-1} \end{bmatrix} + b.$$

If we choose the our step size with the following inequality:

$$\gamma \leq \min\Big(\tfrac{\min(p,q)\mu}{K(14L^2+240L^3)}, \tfrac{1}{18KL}, \tfrac{4}{\mu K}, \tfrac{\min(p,q)p}{2(45KL+108KL^2+K\mu/4)}, \tfrac{\min(p,q)^2}{2(86K^2+864K^2L+K\mu/4)}\Big), \tag{14}$$

then we have $\rho(A) \leq \|A\|_1 \leq 1 - \frac{\mu K \gamma}{4} < 1$. Unrolling the stochastic noise part $\sum_{t=0}^{T-1} A^t b \leq (I-A)^{-1} b$, we have:

$$\mathcal{L}^{T+1} \leq (1 - \tfrac{\mu K \gamma}{4})^T \mathcal{L}^1 + \|(I-A)^{-1} b\|_1.$$

Lower bound $\mathcal{L}^{T+1}$ with $\mathbb{E}\|x_{\mathrm{g}}^{T+1} - x^*\|^2$, then the proof is complete. □

The second term on the RHS of (13) is related to stochastic gradient noise σ^2, which can be controlled by the step size. By choosing a specific step size, we obtain the following result.

Corollary 2. *Under the same conditions as in Theorem 2, if we define* $\zeta_0 = \mathbb{E}\|\overline{x}_{\mathrm{g}}^1 - x^*\|^2 + \Gamma^1$, $\zeta_1 = Y^1 + Z^1$, *and the RHS of* (14) *to be* $\overline{\gamma}$, *then there exists a constant step size* $\gamma = \min(\overline{\gamma}, \frac{\ln(\max(1, \mu K(\zeta_0 + \overline{\gamma}\zeta_1)T/\sigma^2))}{\mu K T})$ *such that:*

$$\mathbb{E}\|\overline{x}_{\mathrm{g}}^{T+1} - x^*\|^2 \leq \tilde{\mathcal{O}}\Big(\exp(-pqT)\cdot(\zeta_0+\zeta_1)\Big) + \mathcal{O}\left(\tfrac{\sigma^2}{\mu K T}\right) + \tilde{\mathcal{O}}\left(\tfrac{L^5\sigma^2}{\mu^5 T^5 pq}\right). \quad (15)$$

Proof. Starting from (13), we find see that:

$$\begin{aligned}\mathbb{E}\|\overline{x}_{\mathrm{g}}^{T+1} - x^*\|^2 &\leq (1 - \tfrac{\mu K\gamma}{4})^T \mathcal{L}^1 + \|(I-A)^{-1}b\|_1 \\ &\leq \exp(-\tfrac{\mu K \gamma}{2} T)\mathcal{L}^1 + \mathcal{O}\left(\tfrac{K^5 L^5 \sigma^2 \gamma^5}{pq}\right).\end{aligned}$$

If we define the RHS of (14) to be $\overline{\gamma} = \mathcal{O}(\frac{\mu pq}{L^3 K})$ and choose $\gamma = \min(\overline{\gamma}, \frac{\ln(\max(1, \mu K(\zeta_0 + \overline{\gamma}\zeta_1)T/\sigma^2))}{\mu K T})$, then we have:

$$\mathbb{E}\|\overline{x}_{\mathrm{g}}^{T+1} - x^*\|^2 \leq \mathcal{O}\Big(\exp(-\frac{\mu K \overline{\gamma}}{2} T)(\zeta_0 + \overline{\gamma}\zeta_1)\Big) + \mathcal{O}\left(\frac{\sigma^2}{\mu K T}\right) + \tilde{\mathcal{O}}\left(\frac{L^5\sigma^2}{\mu^5 T^5 pq}\right).$$

Plug in $\overline{\gamma}$ and the proof is complete. □

With the strong-convexity assumption, the theorem demonstrates that our algorithm has a linear convergence rate under $\sigma^2 = 0$. The terms Y^t and Z^t in both Lyapunov functions capture how the two gradient tracking terms y^t and z^t approximate the gradient information in the network, respectively. The term Γ^t tracks the distance between the aggregated model during DS communication and the local models before aggregation. In both corollaries , we observe the that these three terms converge to zero along with $\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^t)\|^2$ and $\mathbb{E}\|\overline{x}_{\mathrm{g}}^{T+1} - x^*\|^2$ when T goes to infinity.

C. Learning-Efficiency Co-Optimization

Based on Corollary 1, we can see that the convergence speed is determined by $\frac{1}{p^4 q^2}$. Without any additional conditions, the best choice to maximize the convergence speed is setting $p = q = 1$. However, recall that the main objective of employing a semi-decentralized FL setting is to save on the communication costs between clients and the FL server, which is contradicted by choosing $p = 1$ (because the communication cost between devices and server would be maximized). We therefore propose a method for the central server to trade-off between convergence speed and communication cost, e.g., the energy or monetary cost for wireless bandwidth usage.

For each subnet $\mathcal{C}_s$, the server estimates a communication cost of E_s for pulling and pushing variables from the subnet $\mathcal{C}_s$ and a communication cost of E_s^{D2D} for every round of D2D communication performed by the subnet $\mathcal{C}_s$. As in the case of Theorem 1, we define $\beta_s = \frac{m_s - h_s}{m_s}$ to be the ratio of unsampled clients from each subnet. The server can solve the following minimization problem with a given the balance terms $\lambda_1, \lambda_2, \lambda_3, \lambda_4 > 0$.

$$\begin{aligned}\min_{\beta_1,\ldots,\beta_S,p,K} \quad & \tfrac{1}{p^4} + \lambda_1\sqrt{\tfrac{1}{K}} + \lambda_2(\tfrac{1}{Kp^2})^{\frac{2}{3}} \\ & + \lambda_3 \textstyle\sum_{s=1}^{S}(1-\beta_s)\cdot E_s \\ & + \lambda_4 K \textstyle\sum_{s=1}^{S} E_s^{D2D}, \\ \text{subject to} \quad & 0 \leq \beta_s \leq \tfrac{m_s - 1}{m_s}, \quad 1 \leq i \leq l, \\ & p = \min(1-\beta_1^2, \ldots, 1-\beta_s^2).\end{aligned} \quad (16)$$

The ratio between E_s^{D2D} and E_s is set to $E_s^{D2D}/E_s = \delta$. A small δ means that the cost for a client to communicate with its neighbors using D2D communication is much cheaper than to perform a centralized aggregation.

Optimization (16) has a similar form to a geometric program (GP), with the exception of the last constraint $p = \min(1 - \beta_1^2, \ldots, 1-\beta_S^2)$. By relaxing it to $p \leq \min(1-\beta_1^2, \ldots, 1-\beta_S^2)$, it becomes a posynomial inequality constraint which can be handled by GP [38]. The solution of the relaxed problem will remain the same as (16), which is provable via a straightforward contradiction. It is worth comparing this procedure to the more complex control algorithm proposed in [15], where one must adaptively choose a smaller K when client models in a subnet deviate from each other too quickly. With our GT methodology, the subnet drift is inherently controlled, allowing a constant K throughout the whole training process.

V. Numerical Evaluation

A. Experimental Setup

System Setup. We set the system to have $n = 30$ clients, the only exception being Figure 3b, where the system contains 60 clients. All system have $S = 6$ subnets, each with same number of clients. Each subnet's graph structure is a generated as a random geometric graph with radius between 0.5 and 3.5. The doubly stochastic weight matrices $W_1, ..., W_S$ for each subnet are generated using the Metropolis-Hasting rule.

Datasets. There are in total four datasets that we use for testing. The first three are all image classification datasets, and last one one is a synthetic dataset.

(i) Real-world Datasets: We first consider a neural network classification task using a cross entropy loss function on image datasets (MNIST [39], CIFAR10, CIFAR100 [40]). For all three datasets we use the training set for training and testing set for evaluation. Let $\mathcal{D}_i$ be the dataset allocated on client i; we set $|\mathcal{D}_i|$ to be the same $\forall i$. To simulate high data heterogeneity, all three datasets are partitioned in a non-i.i.d manner such that each client in the MNIST and CIFAR10 experiments only holds data of one class and each client in the CIFAR100 experiments only holds data from four classes.

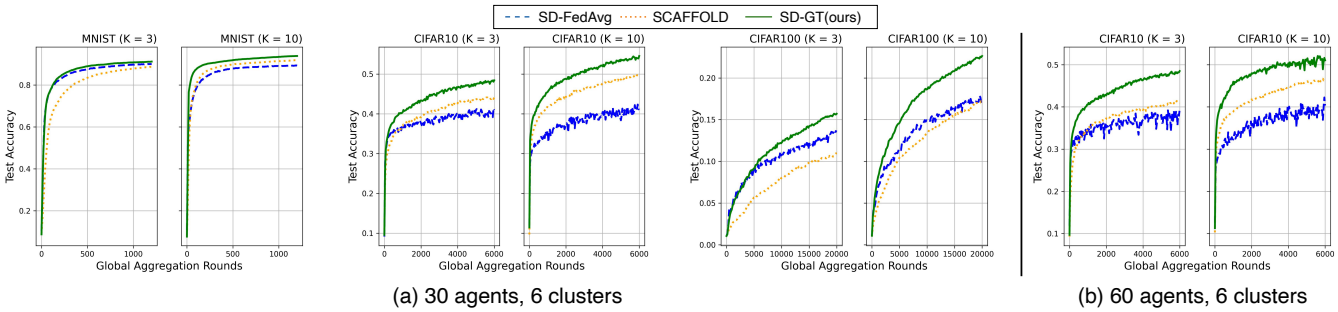

Fig. 3: Experimental results on real-world datasets ($\frac{h_s}{m_s} = 40\%$). By fixing the sampling rate for each subnet to 40 percent and observe the effect of performing multiple D2D communication rounds, we see that our algorithm SD-GT gains the most improvement from increasing the D2D rounds. The advantage of our method can still be observed even if we double the number of clients in the system.

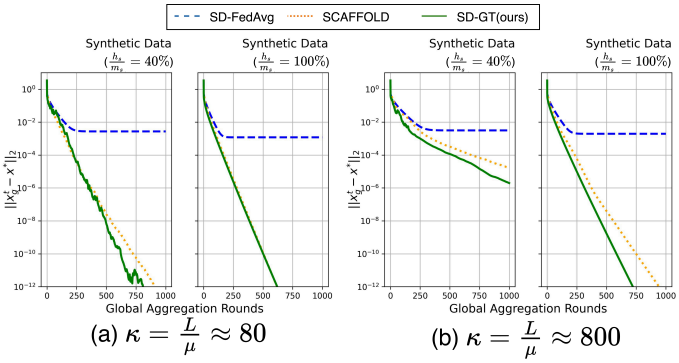

Fig. 4: Experimental results from synthetic dataset ($K = 40$). By fixing the number of D2D rounds performed between two global aggregations, we see that our algorithm SD-GT is able to improve from increasing the sampling rate of each subnet while maintaining linear rate.

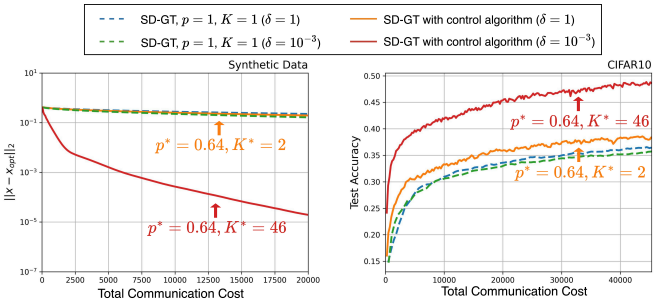

Fig. 5: Experiments on the proposed co-optimization algorithm. When δ is small (D2D communication is cheap), the co-optimization algorithm is able to choose the sample rate and the number of D2D communication rounds such that more improvement is obtained using the same amount of communication.

(ii) Synthetic Dataset: We consider a strongly convex Least Square (LS) problem using synthetic data. The reason that we show results on this dataset is to demonstrate the linear convergence of our algorithm under strong-convexity. In the LS problem, each client i estimates an unknown signal $x_0 \in \mathbb{R}^d$ through linear measurements $b_i = A_i x_0 + n_i$, where $A_i \in \mathbb{R}^{|\mathcal{D}_i| \times d}$ is the sensing matrix, and $n_i \in \mathbb{R}^{|\mathcal{D}_i|}$ is the additive noise. The value of d is set to 200 and x_0 is a vector of i.i.d. random variables drawn from a normal distribution $\mathcal{N}(0, 1)$. Each client i receives $|\mathcal{D}_i| = 30$ observations. All additive noise is sampled from the same distribution $\mathcal{N}(0, 0.04)$. The sensing matrix A_i is generated as follows: For the j^{th} row of the sensing matrix $a_{ij} \in \mathbb{R}^d$, let $z_1, ..., z_d$ be an i.i.d. sequence of $\mathcal{N}(0, 1)$ variables, and fix some correlation parameter $\omega \in [0, 1)$. We initialize by setting the first element of the j^{th} row $a_{ij,1} = \frac{z_1}{\sqrt{1-\omega^2}}$, and generate the remaining entries by applying the recursive update $a_{ij,t+1} = \omega a_{ij,t} + z_{t+1}$, for $t = 1, 2, \ldots, d-1$.

Baselines. We consider two baselines for comparison with SD-GT.

(i) SD-FedAvg: Our first baseline is a semi-decentralized version of FedAvg [15] (denoted as SD-FedAvg). Each client updates using only its local gradient, and communicates with its nearby neighbors within each subnet using D2D communication after every gradient computation. A global aggregation is conducted after every K rounds of gradient computation. This baseline does not contain gradient tracking, thus allowing us to assess this component of our methodology.

(ii) SCAFFOLD: We also run a comparison with SCAFFOLD [7], which is a centralized gradient tracking algorithm that samples the network for aggregation. This baseline does not contain semi-decentralized local model updates, which means while SD-FedAvg and SD-GT performs K rounds of D2D communication and model update, SCAFFOLD computes K rounds of local on-device updates. Comparing with SCAFFOLD allows us to assess the benefit of our methodology co-designing semi-decentralized updates with gradient tracking.

Parameter Settings. For experiments using the synthetic dataset, we tuned the value of ω to create two settings: (i) $\kappa \approx 80$, which is a simpler learning task, and (ii) $\kappa \approx 800$, which is more complicated. The step size for synthetic dataset experiments is set to 10^{-4}. The step size is set to $\gamma = 1 \times 10^{-2}$ for MNIST and $\gamma = 3 \times 10^{-3}$ for both CIFAR10 and CIFAR100. The neural network that we use for ML-tasks is a two layer fully-connected neural network with ReLU activations. We use deterministic gradients for the synthetic dataset and a batch-size of 512 for the three ML datasets.

For experiments on our co-optimization algorithm, the communication cost E_s for each subnet is a randomly drawn number between 1 and 100. The balance terms are set to $\lambda_1 = \lambda_2 = 1$, $\lambda_3 = 10^{-1}$ and $\lambda_4 = 10^{-2}$.

B. Results and Discussion

Experiment 1: Convergence results for varying D2D rounds. In Figure 3, we compare the model training perfor-

mance of the algorithms on the image classification datasets. In these experiments, the central server samples 40% of the clients from each subnet. We change the number of D2D rounds from $K = 3$ to $K = 10$ to observe the effect of performing multiple rounds of in-subnet consensus operations. In the MNIST and CIFAR10 experiments, we can see that SD-FedAvg struggles to gain improvement from increasing the number of D2D communications, while SD-GT is able to track the gradient information from other parts of the network and *obtains better results when using larger number of D2D rounds*. Since in our setting we consider D2D communications to be relatively low cost compare to DS communications, we also see that *the convergence of SCAFFOLD, which does not include D2D communication, is slower than SD-GT* even though both algorithms uses gradient tracking to correct the update direction.

In experiments utilizing CIFAR100, unlike the results in MNIST and CIFAR10 experiments, SD-FedAvg is able to gain improvements from increasing the D2D rounds in the CIFAR100 experiments. This result is similar to how [18] compares between D-SGD and their method by running experiments on a evenly distributed MNIST and a non-i.i.d. MNIST. GT based methods do not gain much improvement over non-GT based methods when all clients are holding i.i.d. datasets, but gradient tracking is very important when dealing with networks with non-i.i.d. data. *Our algorithm deals with data-heterogeneity better than algorithms that do not use gradient tracking, gaining more improvement from increasing the number of in-subnet D2D communication rounds.*

Experiment 2: Convergence results for varying DS sample rates. Figure 4 compares the learning performance between the algorithms on the synthetic dataset. We set the number of D2D rounds to $K = 40$ for all experiments and compare between different sampling rates $\frac{h_s}{m_s}$. We see that SD-GT has a linear convergence to the global optimal solution under strong-convexity for any sampling rate and under different κ values, which aligns with our theoretical results. When κ is large, D2D communication improves the overall convergence speed, which gives our algorithm a faster linear convergence speed compared to SCAFFOLD, and SD-FedAvg converges to a radius away from the optimal solution related to the step size and the data-hetorogeneity assumption [41], [42]. We also see that SCAFFOLD, while converging linearly, does so at a slower rate than SD-GT when κ is large. The lack of D2D communication in SCAFFOLD makes the model aggregated from device to server contain gradient information of only a single device instead of the whole subnet. *This emphasizes the performance benefits of combining D2D communications with gradient tracking in our algorithm when compared to the baselines.*

Experiment 3: Co-optimization algorithm under different communication costs. Figure 5 evaluates the impact of `SD-GT`'s learning-efficiency co-optimization procedure. Overall, we see that the optimization leads to improved SD-GT performance in terms of communication efficiency for the same learning quality. We compare the improvement gained from our co-optimization algorithm under a larger delta ($\delta = 1$) and a smaller delta ($\delta = 10^{-3}$). When δ is large, which means that for each client, DS and D2D communications have similar cost, there is no need to perform multiple D2D communication rounds because the communication efficiency of using the co-optimization algorithm is similar to simply aggregating all clients at the central server after every gradient computation. However, when δ is small (i.e., in-subnet D2D communication is much cheaper than DS communication), the co-optimization algorithm's balancing between convergence speed and communication efficiency results in better convergence while incurring the same communication cost.

VI. Conclusion

We developed SD-GT, the first gradient-tracking based semi-decentralized federated learning (FL) methodology. SD-GT incorporates dual gradient tracking terms to mitigate the subnet-drift challenge. We provided a convergence analysis of our algorithm under non-convex settings and strongly-convex settings, revealing conditions under which linear and sub-linear convergence rates are obtained. Based on our convergence results, we developed a low-complexity co-optimization algorithm that trades-off between learning quality and communication cost. Our subsequent experimental results demonstrated the improvements provided by SD-GT over baselines in SD-FL and gradient tracking literature.

Appendix

A. Notation

For the proof, we define several matrix form notations. Let $y^t = [y_1^t, \ldots, y_n^t]$, $z^t = [z_1^t, \ldots, z_n^t]$, and $x^{t,k} = [x_1^{t,k}, \ldots, x_n^{t,k}]$ be the matrix form of the variables. Let $\nabla F(x^{t,k}) = [\nabla f_1(x_1^{t,k}), \ldots, \nabla f_n(x_n^{t,k})]$ be the matrix form of the gradient. Define $J = \frac{\mathbf{1_n}\mathbf{1_n}^T}{n}$ be the averaging matrix of all clients and $J_c = \text{diag}(\frac{\mathbf{1_{m_1}}\mathbf{1_{m_1}}^T}{m_1}, \ldots, \frac{\mathbf{1_{m_S}}\mathbf{1_{m_S}}^T}{m_S})$ to be the block diagonal matrix of averaging matrices for each subnet. Let $W = \text{diag}(W_1, \ldots, W_S)$ be the block diagonal matrix of all weighted matrices for each subnet. Let p and q be constants that represents the mixing ability of inter-subnet and in-subnet (defined in Theorem 1). Define two block diagonal matrix $B = \text{diag}(\beta_1 I_{\mathcal{C}_1}, \ldots, \beta_S I_{\mathcal{C}_S})$ and $B^{'} = \text{diag}((1-\beta_1)I_1, \ldots, (1-\beta_S)I_{\mathcal{C}_S})$ for Lemma 5.

We define $\Delta^t = \frac{1}{n}\sum_{i=1}^n \sum_{k=1}^K \mathbb{E}\|x_i^{t,k} - x_{\text{g}}^t\|^2$ to be the **client drift** term, $\Gamma^t = \frac{1}{n}\sum_{i=1}^n \mathbb{E}\|x_i^{t-1,K+1} - x_{\text{g}}^t\|^2$ be the **sampling error** term, and $Z^t = \frac{1}{n}\mathbb{E}\|z^t + \nabla F(x_{\text{g}}^t)(I - J_c)\|_F^2$, $Y^t = \frac{1}{n}\mathbb{E}\|y^t + \nabla F(x_{\text{g}}^t)(J_c - J)\|_F^2$ to be the **in-subnet** and **inter-subnet correction** terms, respectively.

B. Supporting Lemmas

Lemma 1. *(Unroll recursion lemma) For any parameters $r_0 \geq 0, b \geq 0, e \geq 0, u \geq 0$, there exists a constant step size $\gamma < \frac{1}{u}$ s.t.*

$$\Psi_T := \tfrac{r_0}{T}\tfrac{1}{\gamma} + b\gamma + e\gamma^2 \leq 2\sqrt{\tfrac{br_0}{T}} + 2e^{\frac{1}{3}}(\tfrac{r_0}{T})^{\frac{2}{3}} + \tfrac{ur_0}{T}. \quad (17)$$

Proof. See Lemma C.5 in [18] or Lemma 15 in [43]. □

Lemma 2. *(Non-convex descent lemma) Under Assumption 1, if we choose step size $\gamma < \frac{1}{4KL}$, then we have:*

$$\mathbb{E}f(\overline{x}_{\mathrm{g}}^{t+1}) \leq \mathbb{E}f(\overline{x}_{\mathrm{g}}^{t}) - \tfrac{\gamma K}{4}\mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^{t})\|^2 + \tfrac{\gamma L^2}{n}\textstyle\sum_{i,k}\mathbb{E}\|x_i^{t,k} - \overline{x}_{\mathrm{g}}^{t}\|^2. \tag{18}$$

Proof. By injecting L-smoothness, we can get:

$$\begin{aligned}\mathbb{E}f(\overline{x}_{\mathrm{g}}^{t+1}) \leq& \mathbb{E}f(\overline{x}_{\mathrm{g}}^{t}) + \mathbb{E}\langle\nabla f(\overline{x}_{\mathrm{g}}^{t}), \overline{x}_{\mathrm{g}}^{t+1} - \overline{x}_{\mathrm{g}}^{t}\rangle + \tfrac{L}{2}\mathbb{E}\|\overline{x}_{\mathrm{g}}^{t+1} - \overline{x}_{\mathrm{g}}^{t}\|^2\\ \leq& \mathbb{E}f(\overline{x}_{\mathrm{g}}^{t}) + (L^2K^2\gamma^2 - \tfrac{\gamma K}{2})\mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^{t})\|^2\\ &+ (L^3K\gamma^2 + \tfrac{\gamma L^2}{2})\tfrac{1}{n}\textstyle\sum_{i,k}\mathbb{E}\|x_i^{t,k} - \overline{x}_{\mathrm{g}}^{t}\|^2\\ \leq& \mathbb{E}f(\overline{x}_{g}^{t}) - \tfrac{\gamma K}{4}\mathbb{E}\|\nabla f(\overline{x}_{g}^{t})\|^2\\ &+ \gamma L^2\tfrac{1}{n}\textstyle\sum_{i,k}\mathbb{E}\|x_i^{t,k} - \overline{x}_{g}^{t}\|^2.\end{aligned}$$

Let $\gamma < \frac{1}{KL}$ completes the proof. □

Lemma 3. *(Strongly-convex descent lemma) Under Assumptions 1 and 2, if we choose $\gamma < \frac{1}{18KL}$, then we have:*

$$\begin{aligned}\mathbb{E}\|\overline{x}_{\mathrm{g}}^{t} - x^*\|^2 \leq& (1 - \frac{\mu K\gamma}{2})\mathbb{E}\|\overline{x}_{\mathrm{g}}^{t-1} - x^*\|^2\\ &+ 9\gamma KL(1-p)\Gamma^{t-1} + 72K^3L\gamma^3Y^{t-1}\\ &+ 72K^3L\gamma^3Z^{t-1} + \frac{2\gamma^2K\sigma^2}{n} + 9K^2\gamma^3L\sigma^2.\end{aligned} \tag{19}$$

Proof. By injecting L-smoothness and μ-strong-convexity at the same time, we can have:

$$\begin{aligned}\mathbb{E}\|\overline{x}_{\mathrm{g}}^{t} - x^*\|^2 =& \mathbb{E}\|\overline{x}_{\mathrm{g}}^{t-1} - \gamma\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\sum_{k=1}^{K}\nabla f_i(x_i^{t,k}, \xi_i^{t,k}) - x^*\|^2\\ \leq& (1 - \tfrac{\mu K\gamma}{2})\mathbb{E}\|\overline{x}_{\mathrm{g}}^{t-1} - x^*\|^2\\ &- 2\gamma K(\mathbb{E}f(\overline{x}_{\mathrm{g}}^{t-1}) - f(x^*))\\ &+ 4\gamma^2K^2\mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^{t-1})\|^2\\ &+ 3\gamma L\Delta^{t-1} + \tfrac{2\gamma^2K\sigma^2}{n}.\end{aligned}$$

By plugging in Lemma 4 and letting $\gamma < \frac{1}{18KL}$ completes the proof. □

Lemma 4. *(Deviation lemma) If we choose step size $\gamma < \frac{1}{8KL}$, then we have:*

$$\begin{aligned}\Delta^t \leq& 3\rho_m^2K\tfrac{1}{n}\Gamma^t + 24K^3\gamma^2Y^r\\ &+ 24K^3\gamma^2Z^r + 6K^3\gamma^2\mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^{t})\|^2 + 3K^2\gamma^2\sigma^2.\end{aligned} \tag{20}$$

Proof. For any given D2D communication round k, simply plug in the iteration $x_i^{t,k} = \sum_{j\in\mathcal{N}_i^{\mathrm{in}}} w_{ij}(x_j^{t,k-1} - \gamma(\nabla f_j(x^{t,k-1}, \xi_j^{t,k-1}) + y_j^t + z_j^t))$:

$$\begin{aligned}&\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\mathbb{E}\|x_i^{t,k} - \overline{x}_{\mathrm{g}}^{t}\|^2\\ =&\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\mathbb{E}\Big\|\sum_{j\in\mathcal{N}_i^{\mathrm{in}}} w_{ij}\Big(x_j^{t,k-1} - \gamma(\nabla f_j(x^{t,k-1}, \xi_j^{t,k-1})\\ &+ y_j^t + z_j^t)\Big) - \overline{x}_{\mathrm{g}}^{t}\Big\|^2\\ \leq&(1 + \tfrac{1}{K-1} + 4K\gamma^2L^2)\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\mathbb{E}\|x_i^{t,k-1} - \overline{x}_{\mathrm{g}}^{t}\|^2\\ &+ 8K\gamma^2Y^r + 8K\gamma^2Z^r + 2K\gamma^2\mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^{t})\|^2 + \gamma^2\sigma^2.\end{aligned} \tag{21}$$

If we let $(1 + \frac{1}{K-1} + 4K\gamma^2L^2) = \alpha$ and $\gamma < \frac{1}{8KL}$, we can have $\alpha^{k-1} \leq \alpha^K \leq 3$ and $\sum_{k'=0}^{k-2}\alpha^{k'} \leq 3K$. Plugging them into the equation (21) completes the proof. □

Lemma 5. *(Inter-subnet GT lemma) With constant step size $\gamma < \frac{1}{\sqrt{6}KL}$, we have:*

$$Y^t \leq (1 - \tfrac{p}{2})Y^{t-1} + \tfrac{10L^2}{p}\Delta^{t-1} + \tfrac{12}{p}\gamma^2L^2K^2\mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^{t-1})\|^2 + \tfrac{2\sigma^2}{pK}. \tag{22}$$

Proof. We have

$$\begin{aligned}nY^t =&\mathbb{E}\|y^t + \nabla F(\overline{x}_{\mathrm{g}}^{t})(J_c - J)\|_F^2\\ =&\mathbb{E}\|(y^{t-1} + \nabla F(\overline{x}_{\mathrm{g}}^{t-1})(J_c - J))B\\ &+ (\tfrac{1}{K}\textstyle\sum_{k=1}^{K}\nabla F(x^{t-1,k}) - \nabla F(\overline{x}_{\mathrm{g}}^{t-1}))(J - J_c)B'\\ &+ (\nabla F(\overline{x}_{\mathrm{g}}^{t}) - \nabla F(\overline{x}_{\mathrm{g}}^{t-1}))(J_c - J)\|_F^2 + n\tfrac{\sigma^2}{K}\\ \leq&(1 - \tfrac{p}{2})nY^{t-1} + \tfrac{6}{p}(\tfrac{(1-\rho_m)^2nL^2}{K}\Delta^{t-1}\\ &+ 2\gamma^2nL^4K\Delta^{t-1} + 2\gamma^2L^2K^2n\mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^{t-1})\|^2)\\ &+ \tfrac{6\gamma^2L^2Kn}{p}\sigma^2 + n\tfrac{\sigma^2}{K}\end{aligned}$$

Letting $\gamma < \frac{1}{\sqrt{6}KL}$ completes the proof. □

Lemma 6. *(In-subnet GT lemma) With constant step size $\gamma < \frac{1}{\sqrt{6}KL}$, we have:*

$$Z^t \leq (1 - \tfrac{q}{2})Z^{t-1} + \tfrac{26L^2}{q}\Delta^{t-1} + \tfrac{12K^2L^2\gamma^2}{q}\mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^{t})\|^2 + \tfrac{2\sigma^2}{qK} \tag{23}$$

Proof. We have:

$$\begin{aligned}nZ^t =&\mathbb{E}\|z^t + \nabla F(\overline{x}_{\mathrm{g}}^{t})(I - J_c)\|_F^2\\ =&\mathbb{E}\|z^{t-1}W + \nabla F(\overline{x}_{\mathrm{g}}^{t-1})(W - J_c)\\ &+ \tfrac{1}{K}\textstyle\sum_{k=1}^{K}(\nabla F(\mathbf{x}^{t-1,k}) - \nabla F(\overline{x}_{\mathrm{g}}^{t-1}))(W - I)\\ &+ (\nabla F(\overline{x}_{\mathrm{g}}^{t}) - \nabla F(\overline{x}_{\mathrm{g}}^{t-1}))(I - J_c)\|_F^2 + \tfrac{n\sigma^2}{K}\\ \leq&(1 - \tfrac{p}{2})nZ^{t-1} + \tfrac{6}{p}(\tfrac{4nL^2}{K}\Delta^{t-1}\\ &+ \mathbb{E}\|\nabla F(x_g^t) - \nabla F(x_g^{t-1})\|_F^2) + \tfrac{\sigma^2}{K}\\ \leq&(1 - \tfrac{q}{2})nZ^{t-1} + \tfrac{6n}{q}(\tfrac{4L^2}{K}\Delta^{t-1} + 2K\gamma^2L^4\Delta^{t-1}\\ &+ 2K^2L^2\gamma^2\mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^{t})\|^2 + L^2K\gamma^2n\sigma^2) + \tfrac{n\sigma^2}{K}\end{aligned}$$

Letting $\gamma < \frac{1}{\sqrt{6}KL}$ completes the proof. □

Lemma 7. *(Sample Gap lemma)*

$$\begin{aligned}\Gamma^t \leq&(1 - \tfrac{p}{2})\Gamma^{t-1} + \tfrac{12}{p}\gamma^2KL^2\Delta^{t-1} + \tfrac{12}{p}\gamma^2K^2Y^{t-1}\\ &+ \tfrac{12}{p}\gamma^2K^2Z^{t-1} + \tfrac{12}{p}\gamma^2K^2\mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^{t-1})\| + K\gamma^2\sigma^2\end{aligned} \tag{24}$$

Proof. We have:

$$\begin{aligned}\Gamma^t =&\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\mathbb{E}\|x_i^{t-1,K+1} - \overline{x}_{\mathrm{g}}^{t}\|^2\\ =&\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\mathbb{E}\|x_i^{t-1,K+1} - \overline{x}_{\mathrm{g}}^{t-1}\\ &+ \gamma\textstyle\sum_{k=1}^{K}\tfrac{1}{n}\sum_{i=1}^{n}\nabla f_i(x_i^{t,k}, \xi_i^{t,k})\|^2\\ \leq&(1 - \tfrac{p}{2})\Gamma^{t-1} + \tfrac{12}{p}\gamma^2KL^2\Delta^{t-1} + \tfrac{12}{p}\gamma^2K^2Y^{t-1}\\ &+ \tfrac{12}{p}\gamma^2K^2Z^{t-1} + \tfrac{12}{p}\gamma^2K^2\mathbb{E}\|\nabla f(\overline{x}_{\mathrm{g}}^{t-1})\|^2 + K\gamma^2\sigma^2\end{aligned}$$

□

REFERENCES

[1] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

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

[3] S. Wang, J. Perazzone, M. Ji, and K. S. Chan, "Federated learning with flexible control," in *IEEE INFOCOM*. IEEE, 2023, pp. 1–10.

[4] M. M. Amiri, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Federated learning with quantized global model updates," *arXiv preprint arXiv:2006.10672*, 2020.

[5] 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 *IEEE INFOCOM 2021*. IEEE, 2021, pp. 1–10.

[6] S. Wang, M. Lee, S. Hosseinalipour, R. Morabito, M. Chiang, and C. G. Brinton, "Device sampling for heterogeneous federated learning: Theory, algorithms, and implementation," in *IEEE INFOCOM 2021*. IEEE, 2021, pp. 1–10.

[7] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5132–5143.

[8] K. Mishchenko, G. Malinovsky, S. Stich, and P. Richtárik, "Proxskip: Yes! local gradient steps provably lead to communication acceleration! finally!" in *International Conference on Machine Learning*. PMLR, 2022, pp. 15 750–15 769.

[9] A. Koloskova, T. Lin, S. U. Stich, and M. Jaggi, "Decentralized deep learning with arbitrary communication compression," *arXiv preprint arXiv:1907.09356*, 2019.

[10] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," *Neural Information Processing Systems*, vol. 30, 2017.

[11] S. Zehtabi, S. Hosseinalipour, and C. G. Brinton, "Decentralized event-triggered federated learning with heterogeneous communication thresholds," in *IEEE CDC*, 2022, pp. 4680–4687.

[12] S. Hosseinalipour, C. G. Brinton, V. Aggarwal, H. Dai, and M. Chiang, "From federated to fog learning: Distributed machine learning over heterogeneous wireless networks," *IEEE Communications Magazine*, vol. 58, no. 12, pp. 41–47, 2020.

[13] V.-D. Nguyen, S. Chatzinotas, B. Ottersten, and T. Q. Duong, "Fedfog: Network-aware optimization of federated learning over wireless fog-cloud systems," *IEEE Transactions on Wireless Communications*, vol. 21, no. 10, pp. 8581–8599, 2022.

[14] S. Hosseinalipour, S. S. Azam, C. G. Brinton, N. Michelusi, V. Aggarwal, D. J. Love, and H. Dai, "Multi-stage hybrid federated learning over large-scale d2d-enabled fog networks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1569–1584, 2022.

[15] 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 Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3851–3869, 2021.

[16] M. Yemini, R. Saha, E. Ozfatura, D. Gündüz, and A. J. Goldsmith, "Semi-decentralized federated learning with collaborative relaying," in *2022 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2022, pp. 1471–1476.

[17] C. Suraci, S. Pizzi, D. Garompolo, G. Araniti, A. Molinaro, and A. Iera, "Trusted and secured d2d-aided communications in 5g networks," *Ad Hoc Networks*, vol. 114, p. 102403, 2021.

[18] Y. Liu, T. Lin, A. Koloskova, and S. U. Stich, "Decentralized gradient tracking with local steps," *arXiv preprint arXiv:2301.01313*, 2023.

[19] Y. Tu, Y. Ruan, S. Wagle, C. G. Brinton, and C. Joe-Wong, "Network-aware optimization of distributed learning for fog computing," in *IEEE INFOCOM 2020*. IEEE, 2020, pp. 2509–2518.

[20] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[21] Z. Wang, H. Xu, J. Liu, H. Huang, C. Qiao, and Y. Zhao, "Resource-efficient federated learning with hierarchical aggregation in edge computing," in *IEEE INFOCOM 2021*. IEEE, 2021, pp. 1–10.

[22] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *IEEE ICC*. IEEE, 2020, pp. 1–6.

[23] X. Wang, Y. Zhao, C. Qiu, Z. Liu, J. Nie, and V. C. Leung, "Infedge: A blockchain-based incentive mechanism in hierarchical federated learning for end-edge-cloud communications," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 12, pp. 3325–3342, 2022.

[24] X. Zhou, X. Ye, I. Kevin, K. Wang, W. Liang, N. K. C. Nair, S. Shimizu, Z. Yan, and Q. Jin, "Hierarchical federated learning with social context clustering-based participant selection for internet of medical things applications," *IEEE Transactions on Computational Social Systems*, 2023.

[25] R. Parasnis, S. Hosseinalipour, Y.-W. Chu, C. G. Brinton, and M. Chiang, "Connectivity-aware semi-decentralized federated learning over time-varying d2d networks," *International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc)*, 2023.

[26] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

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

[28] Y. Tian, Y. Sun, and G. Scutari, "Asy-sonata: Achieving linear convergence in distributed asynchronous multiagent optimization," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2018, pp. 543–551.

[29] A. Koloskova, T. Lin, and S. U. Stich, "An improved analysis of gradient tracking for decentralized machine learning," *Neural Information Processing Systems*, vol. 34, pp. 11 422–11 435, 2021.

[30] Y. Sun, G. Scutari, and A. Daneshmand, "Distributed optimization based on gradient tracking revisited: Enhancing convergence rate via surrogation," *SIAM Journal on Optimization*, vol. 32, no. 2, pp. 354–385, 2022.

[31] S. A. Alghunaim, "Local exact-diffusion for decentralized optimization and learning," *arXiv preprint arXiv:2302.00620*, 2023.

[32] S. Ge and T.-H. Chang, "Gradient and variable tracking with multiple local SGD for decentralized non-convex learning," *arXiv preprint arXiv:2302.01537*, 2023.

[33] A. S. Berahas, R. Bollapragada, and S. Gupta, "Balancing communication and computation in gradient tracking algorithms for decentralized optimization," *arXiv preprint arXiv:2303.14289*, 2023.

[34] X. Zhang, J. Liu, Z. Zhu, and E. S. Bentley, "Low sample and communication complexities in decentralized learning: A triple hybrid approach," in *IEEE INFOCOM 2021*. IEEE, 2021, pp. 1–10.

[35] Y. Huang, Y. Sun, Z. Zhu, C. Yan, and J. Xu, "Tackling data heterogeneity: A new unified framework for decentralized SGD with sample-induced topology," *arXiv preprint arXiv:2207.03730*, 2022.

[36] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6217–6234, 2012.

[37] Y. Sun, G. Scutari, and D. Palomar, "Distributed nonconvex multiagent optimization over time-varying networks," in *2016 50th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2016, pp. 788–794.

[38] S. Boyd, S.-J. Kim, L. Vandenberghe, and A. Hassibi, "A tutorial on geometric programming," *Optimization and engineering*, vol. 8, pp. 67–127, 2007.

[39] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[40] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[41] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on optimization*, vol. 19, no. 4, pp. 1574–1609, 2009.

[42] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[43] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized SGD with changing topology and local updates," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5381–5393.