Robust Decentralized Learning With Local Updates and Gradient Tracking

Sajjad Ghiasvand, Amirhossein Reisizadeh, Mahnoosh Alizadeh, and Ramtin Pedarsani, *Senior Member, IEEE*

***Abstract*—As distributed learning applications such as Federated Learning, the Internet of Things (IoT), and Edge Computing grow, it is critical to address the shortcomings of such technologies from a theoretical perspective. As an abstraction, we consider decentralized learning over a network of communicating clients or nodes and tackle two major challenges: *data heterogeneity* and *adversarial robustness*. We propose a decentralized minimax optimization method that employs two important modules: local updates and gradient tracking. Minimax optimization is the key tool to enable adversarial training for ensuring robustness. Having local updates is essential in Federated Learning (FL) applications to mitigate the communication bottleneck, and utilizing gradient tracking is essential to proving convergence in the case of data heterogeneity. We analyze the performance of the proposed algorithm, Dec-FedTrack, in the case of nonconvex-strongly-concave minimax optimization, and prove that it converges a stationary point. We also conduct numerical experiments to support our theoretical findings.**

***Index Terms*—Decentralized learning, robust federated learning, universal adversarial perturbation, gradient tracking, local updates.**

I. Introduction

LEARNING from distributed data is at the core of modern and successful technologies such as Internet of Things (IoT), Edge Computing, fleet learning, etc., where massive amounts of data are generated across dispersed users. Depending on the application, there are two main architectures for the learning paradigm: (i) A *distributed* setting with a central parameter server that is responsible for aggregating the model and is able to communicate to all the computing nodes or workers; (ii) A *decentralized* setting for which there is no central coordinating node, and all the nodes communicate to their neighbors through a connected communicating graph. In this work, we focus on the latter.

Received 6 May 2024; revised 21 January 2025; accepted 11 March 2025; approved by IEEE TRANSACTIONS ON NETWORKING Editor S. Ioannidis. This work was supported by the National Science Foundation under Grant 2419982, Grant 2342253, Grant 2236483, and Grant 2330154. A preliminary version of this work appeared in [DOI: 10.1109/Allerton63246.2024.10735328]. *(Corresponding author: Sajjad Ghiasvand.)*

Sajjad Ghiasvand, Mahnoosh Alizadeh, and Ramtin Pedarsani are with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106 USA (e-mail: sajjad@ucsb.edu; alizadeh@ece.ucsb.edu; ramtin@ece.ucsb.edu).

Amirhossein Reisizadeh is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: amirr@mit.edu).

Digital Object Identifier 10.1109/TON.2025.3552423

Federated learning (FL) is a novel and promising distributed learning paradigm mostly employed using the main-secondary architecture that aims to find accurate models across distributed nodes [2], [3]. The main premise of FL framework is user data privacy, that is, locally stored data on each entity remains local during the training procedure, which is in contrast to traditional distributed learning paradigms. In the peer-to-peer or decentralized implementation of FL methods which is the focus of this work, distributed nodes update model parameters locally using local optimization modules such as Stochastic Gradient Descent (SGD) and exchange information with their neighboring nodes to reach consensus. In Federated Learning, due to privacy and communication constraints, each communication round consists of *multiple local updates* before each node aggregates the neighboring updates.

While FL enables us to efficiently train a model, an important challenge is to ensure the robustness of the learned model to possible noisy/adversarial perturbations [4]. The problem becomes more critical in FL since due to its distributed nature, it is more vulnerable to the presence of adversarial nodes and adversarial attacks [5]. Adversarial training based on minimax optimization is the key tool to robustify the learned model in machine learning applications [6]. Thus, it is critical to develop decentralized minimax optimization algorithms that are also communication-efficient, i.e. optimization methods that employ local updates suitable for a federated setting. Other applications of federated minimax optimization include using optimal transport to develop personalized FL [7] and robustness against distributed shifts [8]. Another major challenge in decentralized learning methods is data heterogeneity. Data heterogeneity refers to the fact that the data distributions across distributed nodes are statistically heterogeneous (or non-iid). In this work, we employ the *gradient tracking* (GT) technique that guarantees convergence of the algorithm in the presence of data heterogeneity.

Contributions. We propose the Dec-FedTrack algorithm which is a decentralized minimax optimization method over a network of n communicating nodes with two modules of local updates and gradient tracking, and analyze its communication complexity and convergence rate for the case of nonconvex-strongly-concave (NC-SC) minimax optimization. We show that Dec-FedTrack achieves the $O\left(\kappa^5 n^{-1}\epsilon^{-4}\right)$ stochastic first-order oracle (SFO) complexity and the $O\left(\kappa^3\epsilon^{-2}\right)$ communication complexity, where the condition number is defined by $\kappa \triangleq \ell/\mu$. This is the first federated minimax optimization

algorithm that incorporates GT in a decentralized setting. Moreover, we conduct several numerical experiments that demonstrate the communication efficiency and adversarial robustness of Dec-FedTrack over baselines.

II. Related Work

A. Federated Learning With Heterogeneous Data

One of the most challenging aspects of federated learning is data heterogeneity, where training data is not identically and independently distributed across clients (non-i.i.d.). Under such conditions, local models of clients may drift away from the global model optimum, slowing down convergence [9], [10]. Several studies have attempted to tackle this issue in federated learning [11], [12], [13], [14]. However, these studies are typically not decentralized, their results are often limited to (strongly) convex objective functions, or they make restrictive assumptions about the gradients of objective functions. In this context, gradient tracking (GT) algorithms have been proposed to address these challenges [15], [16], [17], [18]. Particularly, in this paper, we also leverage the GT technique to mitigate the data heterogeneity problem.

B. Decentralized Minimization

Many works have examined minimization problems within a decentralized setting [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29]. Works such as K-GT [30], LU-GT [15], [16], [31], [32], and [33] have introduced decentralized algorithms incorporating local updates and GT, although they are tailored for minimization rather than minimax optimization.

C. Centralized Minimax Optimization

Centralized minimax optimization has become increasingly significant, particularly with the rise of machine learning applications like GANs [34] and adversarial training of neural networks. This optimization paradigm tackles the challenges posed by nonconvex-concave and nonconvex-nonconcave problems, drawing attention due to its relevance in various domains. For NC-SC problems, several works have utilized momentum or variance reduction techniques to achieve the SFO complexity of $O\left(\kappa^3\epsilon^{-3}\right)$ [35], [36], [37], [38].

D. Decentralized Minimax Optimization

Numerous studies have explored decentralized minimax optimization for (strongly) convex-concave [39], [40], [41], [42], [43], nonconvex-strongly-concave [44], [45], [46], [47], [48], [49], [50], [51], [52], and nonconvex-nonconcave [53], objective functions. DPOSG [53] has the assumption of identical distributions, and most of the mentioned works on nonconvex-strongly-concave minimax optimization have a very high gradient complexity. The closest ones to our setting and results are DM-HSGD [50], DREAM [49], and black [51]. These studies explore decentralized minimax optimization using gradient tracking and variance reduction techniques. DM-HSGD employs the variance reduction technique of STORM [54], whereas DREAM and black utilize the variance reduction technique of SPIDER [55]. However, clients in these algorithms do not perform multiple local updates between communication rounds, making them unsuitable for federated learning scenarios.

E. Distributed/Federated Minimax Learning

Several works have studied minimax optimization in the federated learning setting across various function types: (strongly) convex-concave [8], [56], [57], [58] and nonconvex-strongly-concave/nonconvex-PL/nonconvex-one-point-concave [59], [60], [61], [62], [63]. FedGDA-GT [58] has delved into federated minimax learning with both local updates and GT, but it is not decentralized and assumes strongly-convex-strongly-concave objective functions. Momentum Local SGDA [62], SAGDA [64], and De-Norm-SGDA [63] explore federated minimax optimization with local updates but lacks decentralization and does not incorporate GT.

We summarize the comparison of related algorithms with Dec-FedTrack in Table I.

III. Problem Setup

We consider a connected network of n clients with $\mathcal{V} = [n] := \{1, \dots n\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ as the set of nodes and edges, respectively. This network collaboratively seeks to solve the following minimax optimization problem:

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

where $f_i(\mathbf{x}, \mathbf{y}) = \mathbb{E}[F_i(\mathbf{x}, \mathbf{y}; \xi_i)]$ denotes the local function associated with node $i \in \mathcal{V}$. Here, the expectation is with respect to $\xi_i \sim \mathcal{D}_i$ and $\mathcal{D}_i$ denotes the local distribution for node i. In our decentralized setting, clients communicate with each other along the edges $e \in \mathcal{E}$, that is, each node is allowed to communicate with its neighboring nodes.

A. Motivating Example: Federated Adversarial Training

Consider a network of clients that wish to train a common model $\mathbf{x}$ that is robust to adversarial perturbation $\mathbf{y}$. In this model, the adversary can attack the network by adding a common perturbation to *all* the samples of every node, i.e. *universal perturbation* [65], [66]. This model corresponds to an adversarial cost function $f_i(\mathbf{x}, \mathbf{y})$ for each node i and results in a minimax problem shown in (1) that should be solved over the connected network. One should add that in adversarial machine learning, the adversary is restricted to a bounded noise power; therefore, in this case, the minimax problem (1) will have a constraint $\|\mathbf{y}\| \leq \delta$.

B. Convergence Measure

In this paper, we focus on a particular setting where each local function $f_i(\mathbf{x}, \mathbf{y})$ is nonconvex in $\mathbf{x}$ and strongly concave in $\mathbf{y}$ which is well-studied in the minimax optimization literature [67]. This assumption allows us to define the *primal* function of (1) for every $\mathbf{x}$ as $\Phi(\mathbf{x}) := \max_{\mathbf{y}\in\mathbb{R}^q} f(\mathbf{x}, \mathbf{y})$. Solving the minimax problem (1) is equivalent to minimizing the primal function, i.e., $\min_{\mathbf{x}\in\mathbb{R}^d} \Phi(\mathbf{x})$ which is nonconvex. A

TABLE I

COMPARISON OF DEC-FEDTRACK WITH RELATED ALGORITHMS FOR MINIMAX AND MINIMIZATION OPTIMIZATION. CRITERIA IN THIS COMPARISON ARE: SFO COMPLEXITY; NUMBER OF COMMUNICATIONS; TYPE OF CENTRALIZATION; TYPE OF FUNCTION CLASS; IF THE ALGORITHM IS STOCHASTIC; AND IF THE ALGORITHM HAS LOCAL UPDATE (LU), HETEROGENEITY ROBUSTNESS (HR), AND ADVERSARIAL ROBUSTNESS (AR)

Name	SFO	Comm. Round	Decentralized	Objective	LU	HR	AR
MLSGDA [62]	$O\left(\frac{\kappa^4}{n\epsilon^4}\right)$	$O\left(\frac{\kappa^3}{\epsilon^3}\right)$	×	NC-SC	✓	×	✓
SAGDA [64]	$O\left(\frac{\kappa^4}{n\epsilon^4}\right)$	$O\left(\frac{\kappa^2}{\epsilon^2}\right)$	×	NC-SC	✓	×	✓
Fed-Norm-SGDA [63]	$O\left(\frac{\kappa^4}{n\epsilon^4}\right)$	$O\left(\frac{\kappa^2}{\epsilon^2}\right)$	×	NC-SC	✓	×	✓
DM-HSGD [50]	$O\left(\frac{\kappa^3}{n\epsilon^3}\right)$	$O\left(\frac{\kappa^3}{\epsilon^3}\right)$	✓	NC-SC	×	✓	✓
DREAM [49]	$O\left(\frac{\kappa^3}{n\epsilon^3}\right)$	$O\left(\frac{\kappa^2}{\epsilon^2}\right)$	✓	NC-SC	×	✓	✓
black [51]	$O\left(\frac{\kappa^4}{n\epsilon^3}\right)$	$O\left(\frac{\kappa^3}{\epsilon^2}\right)$	✓	NC-SC	×	✓	✓
K-GT [30]	$O\left(\frac{1}{n\epsilon^4}\right)$	$O\left(\frac{1}{\epsilon^2}\right)$	✓	NC	✓	✓	×
Dec-FedTrack (Ours)	$O\left(\frac{\kappa^5}{n\epsilon^4}\right)$	$O\left(\frac{\kappa^3}{\epsilon^2}\right)$	✓	NC-SC	✓	✓	✓

well-established convergence measure for such minimization problems is to find a *stationary point* $\hat{\mathbf{x}}$ of Φ, that is a point for which $\|\nabla\Phi(\hat{\mathbf{x}})\| \leq \epsilon$.

C. Notation

We represent vectors using bold small letters and matrices using bold capital letters. The vector $\mathbf{x}_i^{(t)+k}$ denotes a variable on node i at local step k and communication round t, as will be explained in Section IV. The average of vectors $\mathbf{x}_i$ is defined as $\bar{\mathbf{x}} = \frac{1}{n}\sum_i \mathbf{x}_i$. We denote a matrix whose columns are the collection of n vectors, each belonging to a client, as $\mathbf{X} \in \mathbb{R}^{d\times n}$, i.e., $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$. Additionally, we use $\bar{\mathbf{X}}$ to represent a matrix whose columns are equal to $\bar{\mathbf{x}}$, and it can be written in a more useful way as

$$\bar{\mathbf{X}} = [\bar{\mathbf{x}}, \ldots, \bar{\mathbf{x}}] = \frac{1}{n}\mathbf{X}\mathbf{1}_n\mathbf{1}_n^T = \mathbf{X}\mathbf{J} \in \mathbb{R}^{d\times n},$$

where $\mathbf{J} = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$. We also use the below notation for convenience throughout the paper:

$$\begin{aligned}\nabla F(\mathbf{X},\mathbf{Y};\xi) &= \left[\nabla F_1\left(\mathbf{x}_1,\mathbf{y}_1;\xi_1\right),\ldots,\nabla F_n\left(\mathbf{x}_n,\mathbf{y}_n;\xi_n\right)\right],\\ \nabla f(\mathbf{X},\mathbf{Y}) &= \mathbb{E}_{(\xi_1,\ldots,\xi_n)}\nabla F(\mathbf{X},\mathbf{Y};\xi)\\ &= \left[\nabla f_1\left(\mathbf{x}_1,\mathbf{y}_1\right),\ldots,\nabla f_n\left(\mathbf{x}_n,\mathbf{y}_n\right)\right] \in \mathbb{R}^{d\times n}.\end{aligned}$$

We denote the batch sizes for variables $\mathbf{x}$ and $\mathbf{y}$ as b_x and b_y, respectively.

IV. DEC-FEDTRACK ALGORITHM

In this section, we describe our proposed method to solve the minimax problem (1) over a connected network of n nodes. Our method, namely Dec-FedTrack, comprises of two main modules: *local updates* and *gradient tracking* which we elaborate on in the following.

Dec-FedTrack (shown in Algorithm 1) consists of a number of communication rounds, T, where in each round, every node performs K local updates on its variables. In particular, in the kth iteration of round t, each node computes unbiased stochastic gradients and updates its local min and max variables $\mathbf{x}_i$ and $\mathbf{y}_i$ using the so-called *correction terms* (Lines 4 and 5). Next, each node obtains tracking variables

$$\begin{aligned}\mathbf{z}_i^{(t)} &= \frac{1}{K\eta_c}(\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t)+K}),\\ \mathbf{r}_i^{(t)} &= \frac{1}{K\eta_d}(\mathbf{y}_i^{(t)+K} - \mathbf{y}_i^{(t)}),\end{aligned}$$

and sends variable $\{\mathbf{z}_i^{(t)}, \mathbf{r}_i^{(t)}, \mathbf{x}_i^{(t)}, \mathbf{y}_i^{(t)}\}$ to its neighboring nodes. After aggregating these variables from the neighbors, node i updates its correction terms and model variables using

Algorithm 1 Dec-FedTrack

Initialize: $\forall i,j \in [n], \mathbf{x}_i^{(0)} = \mathbf{x}_j^{(0)}, \mathbf{y}_i^{(0)} = \mathbf{y}_j^{(0)}$; $\mathbf{c}_i^{(0)}$ and $\mathbf{d}_i^{(0)}$ according to Lemma 3.

1: **for communication:** $t \leftarrow 0$ to $T-1$ **do**
2: **for** node $i \in [n]$ parallel **do**
3: **for local step:** $k \leftarrow 0$ to $K-1$ **do**
4: Update min variables

$$\mathbf{X}^{(t)+k+1} = \mathbf{X}^{(t)+k} - \eta_c(\nabla_x F(\mathbf{X}^{(t)+k}, \mathbf{Y}^{(t)+k}; \xi^{(t)+k}) + \mathbf{C}^{(t)})$$

5: Update max variables

$$\mathbf{Y}^{(t)+k+1} = \mathbf{Y}^{(t)+k} + \eta_d(\nabla_y F(\mathbf{X}^{(t)+k}, \mathbf{Y}^{(t)+k}; \xi^{(t)+k}) + \mathbf{D}^{(t)})$$

6: **end for**
7: $\mathbf{Z}^{(t)} = \frac{1}{K\eta_c}\left(\mathbf{X}^{(t)} - \mathbf{X}^{(t)+K}\right)$
8: $\mathbf{R}^{(t)} = \frac{1}{K\eta_d}\left(\mathbf{Y}^{(t)+K} - \mathbf{Y}^{(t)}\right)$
9: $\mathbf{C}^{(t+1)} = \mathbf{C}^{(t)} - \mathbf{Z}^{(t)} + \mathbf{Z}^{(t)}\mathbf{W}$
10: $\mathbf{D}^{(t+1)} = \mathbf{D}^{(t)} - \mathbf{R}^{(t)} + \mathbf{R}^{(t)}\mathbf{W}$
11: $\mathbf{X}^{(t+1)} = \left(\mathbf{X}^{(t)} - K\eta_x\mathbf{Z}^{(t)}\right)\mathbf{W}$
12: $\mathbf{Y}^{(t+1)} = \left(\mathbf{Y}^{(t)} + K\eta_y\mathbf{R}^{(t)}\right)\mathbf{W}$
13: **end for**
14: **end for**

gradient tracking [30] as follows:

$$\begin{aligned}
\mathbf{c}_i^{(t+1)} &= \mathbf{c}_i^{(t)} - \mathbf{z}_i^{(t)} + \sum_j w_{ij}\mathbf{z}_j^{(t)},\\
\mathbf{d}_i^{(t+1)} &= \mathbf{d}_i^{(t)} - \mathbf{r}_i^{(t)} + \sum_j w_{ij}\mathbf{r}_j^{(t)},\\
\mathbf{x}_i^{(t+1)} &= \sum_j w_{ij}\left(\mathbf{x}_j^{(t)} - K\eta_x\mathbf{z}_j^{(t)}\right),\\
\mathbf{y}_i^{(t+1)} &= \sum_j w_{ij}\left(\mathbf{y}_j^{(t)} + K\eta_y\mathbf{r}_j^{(t)}\right),
\end{aligned}$$

where $\eta_x := \eta_s\eta_c$ and $\eta_y := \eta_r\eta_d$ denote the global step sizes. The proposed Dec-FedTrack algorithm is described in Algorithm 1 using matrix notations.

Next, we comment on the necessity of using GT in our proposed algorithm. Given that clients' distributions are non-iid, to prove convergence one needs to establish an upper bound on the local gradients. While bounding assumptions can be directly imposed on local gradients, such as Assumption 3b in [68], in many distributed learning settings that are unconstrained, assuming the existence of such bounds can be restrictive. The gradient tracking algorithm [16] addresses this challenge by incorporating a correction term into gradients at each node. In fact, the correction term aims to bring the tracking variable for each client close to the tracking variable of its neighbors, preventing client-drift. The matrix format of the correction term in GT is as follows:

$$\begin{aligned}
\mathbf{X}^{(t+1)} &= \left(\mathbf{X}^{(t)} - \eta\mathbf{Z}^{(t)}\right)\mathbf{W}\\
\mathbf{Z}^{(t+1)} &= \nabla F\left(\mathbf{X}^{(t+1)};\xi^{(t+1)}\right) + \underbrace{\mathbf{Z}^{(t)}\mathbf{W} - \nabla F\left(\mathbf{X}^{(t)};\xi^{(t)}\right)}_{\text{correction term}}.
\end{aligned}$$

V. Convergence Analysis

In this section, we provide rigorous convergence analysis for the proposed Dec-FedTrack algorithm solving (1). We first present the following preliminary definitions for functions with one variable:

Definition 1: A function f is called L-Lipschitz if for any $\mathbf{x}$ and $\mathbf{x}'$, we have $\|f(\mathbf{x}) - f(\mathbf{x}')\| \le L\|\mathbf{x} - \mathbf{x}'\|$.

Definition 2: A function f is called ℓ-smooth if it is differentiable and for any $\mathbf{x}$ and $\mathbf{x}'$, we have $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}')\| \le \ell\|\mathbf{x} - \mathbf{x}'\|$.

Let us proceed with a few assumptions.

As explained before, in our decentralized setting, agents communicate with each other along the edges of a fixed communication graph connecting n nodes. Moreover, each edge of the graph is associated with a positive mixing weight and we denote the mixing matrix by $\mathbf{W} \in \mathbb{R}^{n\times n}$.

Assumption 1: The mixing matrix $\mathbf{W}$ has the following properties: (i) Every element of $\mathbf{W}$ is non-negative, and $W_{i,j} = 0$ if and only if i and j are not connected, (ii) $\mathbf{W}\mathbf{1} = \mathbf{W}^\top\mathbf{1} = 1$, (iii) there exists a constant $0 \le p \le 1$ such that

$$\|\mathbf{X}\mathbf{W} - \bar{\mathbf{X}}\|_F^2 \le (1-p)\|\mathbf{X} - \bar{\mathbf{X}}\|_F^2, \forall \mathbf{X} \in \mathbb{R}^{d\times n}.$$

The mixing rate illustrates the degree of connectivity within the network. A higher p signifies a more interconnected communication graph. When $p = 1$, $\mathbf{W} = \frac{1}{n}\mathbf{1}\mathbf{1}^T$, suggesting full connectivity in the graph, while $p = 0$ yields $\mathbf{W} = \mathbf{I}_n$, indicating a disconnected graph [30].

Assumption 2: We assume that each local objective function f_i is ℓ-smooth, that is, for all $\mathbf{x}, \mathbf{x}', \mathbf{y}, \mathbf{y}'$

$$\|\nabla f_i(\mathbf{x},\mathbf{y}) - \nabla f_i(\mathbf{x}',\mathbf{y}')\|^2 \le \ell^2(\|\mathbf{x} - \mathbf{x}'\|^2 + \|\mathbf{y} - \mathbf{y}'\|^2).$$

We also assume that each $f_i(\mathbf{x},\cdot)$ is μ-strongly concave with respect to its second argument. We denote the condition number by $\kappa := \ell/\mu$.

The above assumption implies that the objective function f in (1) is ℓ-smooth and strongly concave with respect to its second argument.

Assumption 3: We assume that the stochastic gradients are unbiased and variance-bounded, that is,

$$\begin{aligned}
\mathbb{E}\left[\nabla F_i(\mathbf{x},\mathbf{y};\xi_i)\right] &= \nabla f_i(\mathbf{x},\mathbf{y}),\\
\mathbb{E}\|\nabla F_i(\mathbf{x},\mathbf{y};\xi_i) - \nabla f_i(\mathbf{x},\mathbf{y})\|^2 &\le \sigma^2.
\end{aligned}$$

Assumption 4: The function $\Phi(\cdot)$ is lower bounded, that is $\inf_{\mathbf{x}}\Phi(\mathbf{x}) = \Phi^* > -\infty$.

Next, we provide the main result of the paper.

Theorem 1: Suppose Assumptions 1– 4 hold and consider the iterates of Dec-FedTrack in Algorithm 1 with step-sizes $\eta_d = \Theta\left(\frac{p}{\kappa K\ell}\right), \eta_c = \Theta\left(\frac{\eta_d}{\kappa^2}\right)$, and $\eta_s = \eta_r = \Theta(p)$. Then, after T communication rounds each with K local updates, there exists an iterate $0 \le t \le T$ such that $\mathbb{E}\|\nabla\Phi(\bar{\mathbf{x}}^{(t)})\|^2 \le \epsilon^2$ for

$$T = O\left(\frac{\kappa^3}{p^2\epsilon^2}\right)\mathcal{H}_0\ell, \quad K = O\left(\frac{p^2\sigma^2}{\kappa^2 n\epsilon^2} + \frac{\sigma^2}{\epsilon^2} + \frac{\kappa^2\sigma^2}{np\epsilon^2}\right),$$

where $\mathcal{H}_0 = O\left(1 + \frac{\delta_0}{K\kappa p}\right)$ and $\delta_0 = O\left(\frac{q}{\mu^2}\right)$.

Remark 1: Focusing on the dependency of the convergence rate on accuracy ϵ, the above theorem shows that in the regime of interest where ϵ gets small, the algorithm reaches an ϵ-stationary point within $T = O(1/\epsilon^2)$ communication rounds, each consisting of $K = O(1/\epsilon^2)$ local updates. Therefore, the resulting SFO complexity is $T\cdot K = O(1/\epsilon^4)$. As we elaborated in Section II and Table I, the proposed Dec-FedTrack algorithm simultaneously assembles all three components of local updates, heterogeneity and adversarial robustness.

Remark 2: It is also possible to derive the communication complexity for any given K. If we choose step-sizes $\eta_c = \Theta\left(\frac{p}{\kappa^3 K\ell\sqrt{T}}\right)$, $\eta_d = \Theta\left(\frac{p}{\kappa K\ell T}\right)$, and $\eta_s = \eta_r = \Theta(p)$, after T communication rounds each with K local updates, there exists an iterate $0 \le t \le T$ such that $\mathbb{E}\|\nabla\Phi(\bar{\mathbf{x}}^{(t)})\|^2 \le \epsilon^2$ for

$$T = O\left(\frac{\kappa^6}{\epsilon^4 p^4} + \frac{p^4\sigma^4}{n^2\kappa^4 K^2\epsilon^4} + \frac{\kappa^4\sigma^4}{n^2K^2p^2\epsilon^4}\right),$$

which holds for any given K.

A. Proof Steps

We first state the following standard results from optimization theory.

Proposition 1: Under Assumption 2, $\Phi(\cdot)$ is $(\ell+\kappa\ell)$-smooth and $\mathbf{y}^*(\cdot) = \arg\max_{\mathbf{y}\in\mathbb{R}^q} f(\cdot,\mathbf{y})$ is κ-Lipschitz [35].

Proposition 2: Under Assumption 2, for every $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y}, \mathbf{y}' \in \mathbb{R}^q$, we have

$$\nabla_y f(\mathbf{x},\mathbf{y})^\top (\mathbf{y}-\mathbf{y}') + \frac{1}{2\ell}\left\|\nabla_y f(\mathbf{x},\mathbf{y})\right\|^2 + \frac{\mu}{2}\left\|\mathbf{y}-\mathbf{y}'\right\|^2 \leq f\left(\mathbf{x},\mathbf{y}^+\right) - f\left(\mathbf{x},\mathbf{y}'\right),$$

where $\mathbf{y}^+ = \mathbf{y} - \frac{1}{\ell}\nabla_y f(\mathbf{x},\mathbf{y})$ [69].

Next, we introduce some terminology that will be useful throughout the entire proof:

1) The client (node) variance for variable $\mathbf{x}$ that measures the deviation of variable $\mathbf{x}$ at global steps from its averaged model:

$$\Xi_t^x := \frac{1}{n}\sum_i^n \mathbb{E}\|\mathbf{x}_i^{(t)} - \bar{\mathbf{x}}^{(t)}\|^2.$$

2) Client-drift for variable $\mathbf{x}$ that measures the deviation of the variable $\mathbf{x}$ at local steps from its averaged model:

$$e_{k,t}^x := \frac{1}{n}\sum_i^n \mathbb{E}\|\mathbf{x}_i^{(t)+k} - \bar{\mathbf{x}}^{(t)}\|^2.$$

The accumulation of local steps for variable $\mathbf{x}$ is shown by

$$\mathcal{E}_t^x := \sum_{k=0}^{K-1} e_{k,t}^x = \sum_{k=0}^{K-1}\frac{1}{n}\sum_i^n \mathbb{E}\left\|\mathbf{x}_i^{(t)+k} - \bar{\mathbf{x}}^{(t)}\right\|^2.$$

3) The quality of the correction for the variable $\mathbf{x}$ that measures the accuracy of the gradient correction in the local updates, which aims to bring local updates closer to global updates:

$$\gamma_t^x = \frac{1}{n\ell^2}\mathbb{E}\left\|\mathbf{C}^{(t)} + \nabla_x f\left(\bar{\mathbf{X}}^{(t)},\bar{\mathbf{Y}}^{(t)}\right) - \nabla_x f\left(\bar{\mathbf{X}}^{(t)},\bar{\mathbf{Y}}^{(t)}\right)\mathbf{J}\right\|_F^2.$$

Similarly, we can define $\Xi_t^y, e_{k,t}^y, \mathcal{E}_t^y$, and γ_t^y for variable $\mathbf{y}$. 4) Consensus distance for variable $\mathbf{y}$ that measures the deviation of the optimum $\mathbf{y}$ when $\mathbf{x} = \bar{\mathbf{x}}$ and the averaged $\mathbf{y}$, that is, $\delta_t = \|\hat{\mathbf{y}}^{(t)} - \bar{\mathbf{y}}^{(t)}\|^2$ where $\hat{\mathbf{y}}^{(t)} = \arg\max_{\mathbf{y}\in\mathbb{R}^q} f\left(\bar{\mathbf{x}}^{(t)},\mathbf{y}\right)$. Now, we introduce several useful lemmas before the proof of Theorem 1.

Lemma 1: For a set of arbitrary vectors $a_1,\ldots,a_n$ such that $a_i \in \mathbb{R}^d$, we have

$$\left\|\frac{1}{n}\sum_{i=1}^n a_i\right\|^2 \leq \frac{1}{n}\sum_{i=1}^n \|a_i\|^2.$$

Lemma 2: (Young's Inequality) For any vectors $a, b \in \mathbb{R}^d$ and $\alpha > 0$ we have

$$2\langle a,b\rangle \leq \alpha\|a\|^2 + \frac{1}{\alpha}\|b\|^2,$$
$$\|a+b\|^2 \leq (1+\alpha)\|a\|^2 + \left(1+\frac{1}{\alpha}\right)\|b\|^2.$$

Lemma 3: If we initialize $\mathbf{C}^{(0)}$ and $\mathbf{D}^{(0)}$ as below

$$\mathbf{c}_i^{(0)} = -\nabla_x F_i\left(\mathbf{x}^{(0)},\mathbf{y}^{(0)};\xi_i\right) + \frac{1}{n}\sum_j \nabla_x F_j\left(\mathbf{x}^{(0)},\mathbf{y}^{(0)};\xi_j\right),$$
$$\mathbf{d}_i^{(0)} = -\nabla_y F_i\left(\mathbf{x}^{(0)},\mathbf{y}^{(0)};\xi_i\right) + \frac{1}{n}\sum_j \nabla_y F_j\left(\mathbf{x}^{(0)},\mathbf{y}^{(0)};\xi_j\right), \quad (2)$$

then the averaged correction for variables $\mathbf{x}$ and $\mathbf{y}$ in any communication round equals to zero.

Proof. Appendix.

Lemma 4: Using Assumption 2 and Young's Inequality we have

$$\mathbb{E}\left\|\nabla_x f\left(\bar{\mathbf{x}}^{(t)},\bar{\mathbf{y}}^{(t)}\right)\right\|^2 \leq 2\ell^2\delta_t + 2\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2,$$
$$\mathbb{E}\left\|\nabla_y f\left(\bar{\mathbf{x}}^{(t)},\bar{\mathbf{y}}^{(t)}\right)\right\|^2 \leq \ell^2\delta_t.$$

Proof. Appendix.

Next, we provide recursion bounds for client variance, client drift, and quality of correction—for both variables $\mathbf{x}$ and $\mathbf{y}$—as well as consensus distance for variable $\mathbf{y}$.

Lemma 5: Under the assumption that $\eta_c, \eta_d \leq \frac{1}{8KL}$, we can bound the local drift for variables $\mathbf{x}$ and $\mathbf{y}$ as follows

$$\mathcal{E}_t^x \leq 3K\Xi_t^x + 12K^2\eta_c^2\ell^2\mathcal{E}_t^y + 12K^3\eta_c^2\ell^2\gamma_t^x + 12K^3\eta_c^2\ell^2\delta_t + 12K^3\eta_c^2\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2 + 3K^2\eta_c^2\sigma^2,$$
$$\mathcal{E}_t^y \leq 3K\Xi_t^y + 12K^2\eta_d^2\ell^2\mathcal{E}_t^x + 12K^3\eta_d^2\ell^2\gamma_t^y + 6K^3\eta_d^2\ell^2\delta_t + 3K^2\eta_d^2\sigma^2.$$

Proof. Lemma A.5 in the appendix of [70].

Lemma 6: We have the following bounds on client variance for variable $\mathbf{x}$ and $\mathbf{y}$

$$\Xi_{t+1}^x \leq \left(1-\frac{p}{2}\right)\Xi_t^x + \frac{6K\eta_x^2\ell^2}{p}\left(\mathcal{E}_t^x + \mathcal{E}_t^y\right) + \frac{6K^2\eta_x^2\ell^2}{p}\gamma_t^x + K\eta_x^2\sigma^2,$$
$$\Xi_{t+1}^y \leq \left(1-\frac{p}{2}\right)\Xi_t^y + \frac{6K\eta_y^2\ell^2}{p}\left(\mathcal{E}_t^x + \mathcal{E}_t^y\right) + \frac{6K^2\eta_y^2\ell^2}{p}\gamma_t^y + K\eta_y^2\sigma^2.$$

Proof. Lemma A.6 in the appendix of [70].

Lemma 7: The sum of averaged progress between communications for variables $\mathbf{x}$ and $\mathbf{y}$ can be bounded by

$$\mathbb{E}\left\|\bar{\mathbf{x}}^{(t+1)} - \bar{\mathbf{x}}^{(t)}\right\|^2 + \mathbb{E}\left\|\bar{\mathbf{y}}^{(t+1)} - \bar{\mathbf{y}}^{(t)}\right\|^2 \leq 2K\ell^2\left(\eta_x^2+\eta_y^2\right)\left(\mathcal{E}_t^x + \mathcal{E}_t^y\right) + 2K^2\ell^2\left(2\eta_x^2+\eta_y^2\right)\delta_t + 4K^2\eta_x^2\mathbb{E}\left\|\nabla\Phi\left(\bar{x}^{(t)}\right)\right\|^2 + \frac{K\sigma^2}{n}\left(\eta_x^2+\eta_y^2\right).$$

Proof. Appendix.

Lemma 8: Assuming that $\eta_x, \eta_y \leq \frac{\sqrt{p}}{\sqrt{24K\ell}}$, we have the following bounds on the quality of correction for variables $\mathbf{x}$ and $\mathbf{y}$

$$\gamma_{t+1}^x \leq \left(1-\frac{p}{2}\right)\gamma_t^x + \frac{25}{pK}\left(\mathcal{E}_t^x + \mathcal{E}_t^y\right) + \frac{12K^2\ell^2}{p}\left(2\eta_x^2+\eta_y^2\right)\delta_t + \frac{24K^2\eta_x^2}{p}\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2 + \frac{2\sigma^2}{K\ell^2}, \quad (3)$$
$$\gamma_{t+1}^y \leq \left(1-\frac{p}{2}\right)\gamma_t^y + \frac{25}{pK}\left(\mathcal{E}_t^x + \mathcal{E}_t^y\right) + \frac{12K^2\ell^2}{p}\left(2\eta_x^2+\eta_y^2\right)\delta_t + \frac{24K^2\eta_x^2}{p}\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2 + \frac{2\sigma^2}{K\ell^2}. \quad (4)$$

Lemma 9: Using Proposition 2 and assuming that $\eta_y \leq \frac{1}{K\ell}$, we have the following bound on $\mathbb{E}\left\|\hat{\mathbf{y}}^{(t)} - \bar{\mathbf{y}}^{(t+1)}\right\|^2$ for any $\alpha > 0$:

$$\mathbb{E}\left\|\hat{\mathbf{y}}^{(t)} - \bar{\mathbf{y}}^{(t+1)}\right\|^2 \leq (1+\alpha)(1-K\eta_y\mu)\,\delta_t + \left(1+\frac{1}{\alpha}\right)\eta_y^2\ell^2 K\left(\mathcal{E}^x + \mathcal{E}^y\right) + \frac{K\eta_y^2\sigma^2}{n}.$$

Proof. Appendix.

Lemma 10: Assuming that $\eta_x \leq \frac{\eta_y}{4\sqrt{6}\kappa^2}$ and $\eta_y \leq \frac{1}{K\ell}$, we have the following bound on δ_t

$$\delta_{t+1} \leq \left(1 - \frac{K\eta_y\ell}{6\kappa}\right)\delta_t + 12\eta_y\ell\kappa\left(\mathcal{E}_t^x + \mathcal{E}_t^y\right) + \frac{16\kappa^3 K\eta_x^2}{\eta_y\ell}\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2 + \frac{8\eta_y\sigma^2\kappa}{n\ell}.$$

Proof. Appendix.

Now, we state the following descent lemma for $\Phi(\mathbf{x})$:

Lemma 11: Assuming that $\eta_x \leq \frac{1}{16K\ell\kappa}$, we have the following bound on $\mathbb{E}\Phi\left(\bar{\mathbf{x}}^{(t+1)}\right)$ as follows

$$\mathbb{E}\Phi\left(\bar{\mathbf{x}}^{(t+1)}\right) \leq \mathbb{E}\Phi\left(\bar{\mathbf{x}}^{(t)}\right) + 2\eta_x\ell^2\left(\mathcal{E}_t^x + \mathcal{E}_t^y\right) + 2\ell^2\eta_x K\delta_t - \frac{\eta_x K}{4}\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2 + \frac{K\eta_x^2\ell\sigma^2\kappa}{n}.$$

Proof. Appendix.

Using Lemmas 5–9, we have the following recursive bound on the Lyapunov function $\mathcal{H}_t$.

Lemma 12: Under the assumption that $\eta_d = \Theta\left(\frac{p}{\kappa K\ell}\right)$, $\eta_c = \Theta\left(\frac{\eta_d}{\kappa^2}\right)$, and $\eta_s = \eta_r = \Theta(p)$, we can find constants A_x, A_y, B_x, B_y, and C, such that $D > 0$ and $D_9 \geq 0$, and we have

$$\mathcal{H}_{t+1} - \mathcal{H}_t \leq -DK\eta_x\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2 + D_9K\ell\eta_d^3\sigma^2 + \frac{K\eta_x^2\ell\kappa}{n}\sigma^2 + \frac{8\eta_y}{np}\sigma^2,$$

where

$$\mathcal{H}_t = \mathbb{E}\Phi\left(\bar{\mathbf{x}}^{(t)}\right) - \mathbb{E}\Phi\left(\mathbf{x}^*\right) + A_x\eta_d K\ell^2\Xi_t^x + A_y\eta_d K\ell^2\Xi_t^y + B_xK^3\ell^4\eta_d^3\gamma_t^x + B_yK^3\ell^4\eta_d^3\gamma_t^y + C\frac{\ell}{\kappa p}\delta_t. \tag{5}$$

Proof. Appendix.

Proof of Theorem 1: Using the telescopic sum for $\mathcal{H}_t$, we have

$$\frac{1}{T+1}\sum_{t=0}^{T}\left(\mathcal{H}_{t+1} - \mathcal{H}_t\right) = \frac{1}{T+1}\left(\mathcal{H}_{T+1} - \mathcal{H}_0\right)$$

$$\leq -DK\eta_x\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2 + D_9K^2\ell^2\eta_d^3\sigma^2 + \frac{K\eta_x^2\ell\kappa}{n}\sigma^2 + \frac{8\eta_y}{np}\sigma^2,$$

which results in

$$\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2 \leq \frac{\mathcal{H}_0 - \mathcal{H}_{T+1}}{(T+1)D}\frac{1}{K\eta_x} + \frac{D_9K\ell^2\eta_d^3}{D\eta_x}\sigma^2 + \frac{\eta_x\ell\kappa}{nD}\sigma^2 + \frac{8\eta_y}{nDKp\eta_x}\sigma^2. \tag{6}$$

Now, we want to ensure $\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2 \leq \epsilon^2$ for any arbitrary $\epsilon > 0$, which is equivalent to bounding each term on the RHS of (6) to the order of ϵ^2. Given that $D = \Theta(1)$, $D_9 = O\left(\frac{1}{p}\right)$, $\eta_x = \Theta\left(\frac{p^2}{\kappa^3 K\ell}\right)$, and $\eta_y = \Theta\left(\frac{p^2}{\kappa K\ell}\right)$, we have

$$T = O\left(\frac{\kappa^3}{p^2\epsilon^2}\right)\mathcal{H}_0\ell,$$

$$K = O\left(\frac{p^2\sigma^2}{\kappa^2 n\epsilon^2} + \frac{\sigma^2}{\epsilon^2} + \frac{\kappa^2\sigma^2}{np\epsilon^2}\right),$$

where $\mathcal{H}_0 = O\left(1 + \frac{\delta_0}{K\kappa p}\right)$ and $\delta_0 = O\left(\frac{q}{\mu^2}\right)$. □

VI. Numerical Results

A. Robust Logistic Regression

We consider the problem of training a robust logistic regression classifier with a non-convex regularizer similar to [37], [49], [50]. In this problem, we aim to train a binary classifier $\mathbf{x} \in \mathbb{R}^d$ on the dataset $\{(a_{ij}, b_{ij})\}$, where $a_{ij} \in \mathbb{R}^d$ denotes the feature vector and $b_{ij} \in \{-1, +1\}$ represents the label for the jth sample in the dataset associated with client i. Each client is allocated m samples, resulting in a total of $N = mn$ samples. The loss function at client i is given by

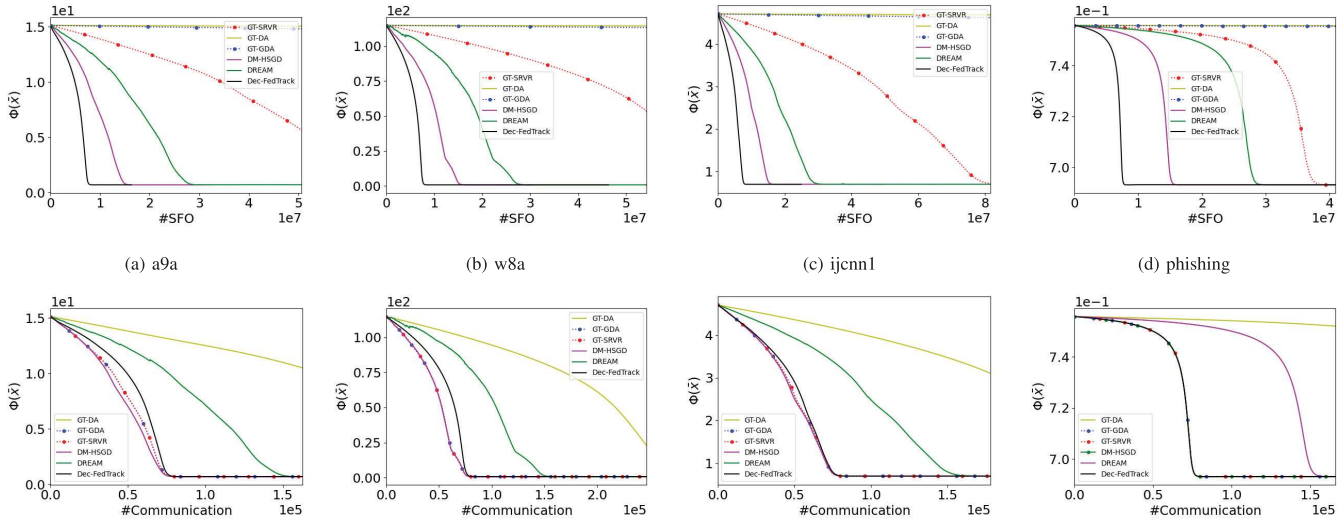
$$f_i(\mathbf{x}, \mathbf{y}) \triangleq \frac{1}{m}\sum_{j=1}^{m}\left(\mathbf{y}_{ij}l_{ij}(\mathbf{x}) - V(\mathbf{y}) + g(\mathbf{x})\right),$$

where $l_{ij}(\mathbf{x}) = \log\left(1 + \exp\left(b_{ij}a_{ij}^\top\mathbf{x}\right)\right)$, $V(\mathbf{y}) = \frac{1}{2N^2}\|N\mathbf{y} - \mathbf{1}\|^2$, $g(\mathbf{x}) = \theta\sum_{k=1}^{d}\frac{\nu x_k^2}{1+\nu x_k^2}$, $\theta = 10^{-5}$, and $\nu = 10$. The parameter $\mathbf{y}$ is restricted to the simplex $\Delta_N = \left\{\mathbf{y} \in \mathbb{R}^N : y_k \in [0,1], \sum_{k=1}^{N} y_k = 1\right\}$. Here, we set the mixing matrix $\mathbf{W}$ as the π-lazy random walk matrix [50] on a ring graph with $n = 10$.

As previously highlighted, the main distinction of Dec-FedTrack compared to other decentralized minimax methods lies in its use of multiple local updates, which aligns well with FL applications. Notably, multiple local steps are essential in FL to ensure privacy.

However, in this section, we set the number of local updates for the Dec-FedTrack algorithm to 1 ($K = 1$) and compare our proposed algorithm against DREAM [49], DM-HSGD [50], GT-DA [52], GT-GDA, and GT-SRVR [71]. These comparisons are conducted on the datasets "a9a", "ijcnn1", "phishing", and "w8a" [72], evaluating performance in terms of the number of SFO calls and communication rounds against $\Phi(\bar{\mathbf{x}}) = \max_{\mathbf{y}\in\Delta_N} f(\bar{\mathbf{x}}, \mathbf{y})$, as well as test accuracy.

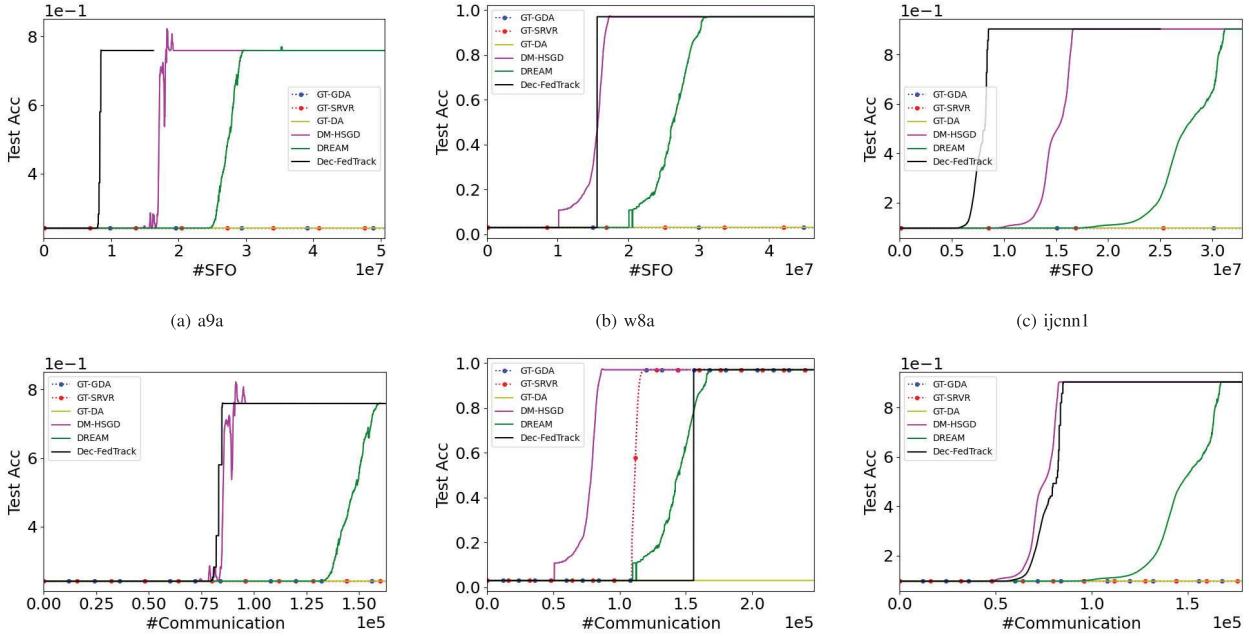
We fix the batch size to 64 across all algorithms and tune the learning rates with $\eta_x \in \{0.1, 0.01, 0.001, 0.0001\}$ and $\eta_y \in \{1, 0.1, 0.01, 0.001\}$. Fig. 1 presents the comparison of the number of SFO calls and number of communication rounds against $\Phi(\bar{\mathbf{x}})$ on datasets "a9a", "ijcnn1", "phishing", and "w8a". As shown, Dec-FedTrack demonstrates a faster decay rate on $\Phi(\bar{\mathbf{x}})$ against the number of SFO calls and faster or very close decay rate on $\Phi(\bar{\mathbf{x}})$ against the number of communications. Furthermore, Fig. 2 compares the comparison



(a) a9a (b) w8a (c) ijcnn1 (d) phishing

(e) a9a (f) w8a (g) ijcnn1 (h) phishing

Fig. 1. Convergence of $\Phi(\bar{\mathbf{x}})$ against the number of SFO calls (above) and the number of communication rounds (bottom).



(a) a9a (b) w8a (c) ijcnn1

(d) a9a (e) w8a (f) ijcnn1

Fig. 2. Test accuracy against the number of SFO calls (above) and the number of communication rounds (bottom).

of the number of SFO calls and number of communication rounds against the test accuracy on datasets "a9a", "ijcnn1", and "w8a". Note that the "phishing" dataset does not include a test dataset.

B. Robust Neural Network Training

In this section, we consider the problem of robust neural network (NN) training, in the presence of adversarial perturbations. We consider a similar problem as considered in [59],

$$\min_{\mathbf{x}} \max_{\|\mathbf{y}\|_\infty \leq \delta} \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}, \mathbf{y})$$

where $f_i(\mathbf{x}, \mathbf{y}) := 1/m \sum_{j=1}^{m} \ell\left(h_{\mathbf{x}}\left(a_{ij} + \mathbf{y}\right), b_{ij}\right)$. Here, $\mathbf{x}$ denotes the parameters of the NN, $\mathbf{y}$ denotes the perturbation, and (a_{ij}, b_{ij}) denotes the j-th data sample of client i.

We consider the accuracy of our formulation against three popular attacks: The Fast Gradient Sign Method (FGSM) [73], Projected Gradient descent (PGD) [74], and Universal Adversarial Perturbation (UAP) [76]. We have provided a description of each attack in Section VI-C.

We evaluate the robustness of Dec-FedTrack against adversarial attacks by comparing it with K-GT, a benchmark minimization algorithm. The evaluation was conducted on the MNIST and CIFAR-10 datasets, utilizing 2-layer and

TABLE II
TEST ACCURACY FOR K-GT AND DEC-FEDTRACK ALGORITHMS UNDER DIFFERENT ATTACK METHODS AND ADVERSARY BUDGETS

Dataset & Model	Method	Clean Acc.	FGSM L_∞ [73]			PGD L_∞ [74]			UAP [75]		
			$\delta = 0.05$	$\delta = 0.1$	$\delta = 0.15$	$\delta = 0.05$	$\delta = 0.1$	$\delta = 0.15$	$\delta = 0.20$	$\delta = 0.25$	$\delta = 0.30$
MNIST	K-GT	**99.20**	93.73	73.10	39.65	94.90	74.67	30.86	93.64	75.15	36.26
	Dec-FedTrack	99.14	**94.83**	**78.02**	**49.06**	**96.20**	**81.72**	**46.49**	**96.14**	**85.73**	**43.87**
			$\delta = 0.003$	$\delta = 0.005$	$\delta = 0.01$	$\delta = 0.003$	$\delta = 0.005$	$\delta = 0.01$	$\delta = 0.03$	$\delta = 0.05$	$\delta = 0.07$
CIFAR-10	K-GT	**77.3**	67.7	44.8	23.6	67.6	44.6	26.4	58.9	53.3	51.5
	Dec-FedTrack	77.1	**69.7**	**51.5**	**32.5**	**69.5**	**51.7**	**35.9**	**74.9**	**66.1**	**56.3**

3-layer convolutional neural networks for training MNIST and CIFAR-10, respectively. For CIFAR-10 experiments, we only use two classes to demonstrate the efficacy of our method.

During training, we set $n = 5$, $K = 5$, and experiment with various constant learning rates chosen from $\{1, 0.5, 0.1, 0.05, 0.01\}$, using a batch size of 128. The results for K-GT and our proposed algorithm under different attack methods and varying values of δ are summarized in Table II. As shown in the table, the proposed algorithm demonstrates superior performance compared to its non-robust counterpart.

C. Adversarial Attacks

In this section, we provide descriptions of the attacks we used in the robust neural network training section.

FGSM [73] is a single-step adversarial attack designed to create adversarial examples by slightly perturbing the input to maximize the loss of a neural network. The FGSM attack perturbs the input a in the direction of the gradient of the loss with respect to the input. This is achieved by computing the gradient of the loss function $f(\mathbf{x}, a, b)$, where $\mathbf{x}$ represents the model parameters and b is the true label. The adversarial example is then generated as:

$$a' = a + \epsilon \cdot \text{sign}(\nabla_a f(\mathbf{x}, a, b)),$$

where ϵ controls the magnitude of the perturbation.

PGD [74] is an iterative extension of FGSM, providing a stronger adversarial attack by applying FGSM multiple times with smaller step sizes. The PGD attack iteratively refines the adversarial example by applying small perturbations to the input. Starting from an initial adversarial example a_0, the method updates the adversarial input a_t at each iteration using the formula:

$$a_{t+1} = \text{Proj}_{\mathcal{B}_\epsilon(a)} \left(a_t + \eta \cdot \text{sign}(\nabla_a f(\mathbf{x}, a_t, b)) \right),$$

where η is the step size, and $\text{Proj}_{\mathcal{B}_\epsilon(x)}$ ensures the perturbed input remains within the L_∞-norm ball of radius ϵ around the original input.

UAP [76] is a technique designed to craft a single perturbation vector $\mathbf{y}$ that, when added to any input, significantly degrades the performance of a model. Unlike input-specific adversarial perturbations (e.g., FGSM or PGD), UAPs are input-agnostic and aim to generalize across a wide range of inputs. We use the universal perturbation introduced in [76], where the authors employ Stochastic Projected Gradient Descent (SPGD) to generate UAP. Their algorithm computes the gradient of the loss function $f(\mathbf{x}, a + \mathbf{y}, b)$ with respect to $\mathbf{y}$ as $g = \nabla_{\mathbf{y}} f(\mathbf{x}, a + \mathbf{y}, b)$. Using SPGD, $\mathbf{y}$ is updated as $\mathbf{y} \leftarrow \mathbf{y} + \eta \cdot g$, where η is the learning rate. After each update, $\mathbf{y}$ is projected back onto the constraint set $\|\mathbf{y}\|_p \leq \delta$ using $\mathbf{y} \leftarrow \text{Proj}_{\|\mathbf{y}\|_p \leq \delta}(\mathbf{y})$. This process is iterated until $\mathbf{y}$ achieves the desired attack success rate across the dataset.

VII. CONCLUSION

This paper presents Dec-FedTrack, a decentralized minimax optimization algorithm specifically tailored for addressing the challenges prevalent in distributed learning systems, particularly within federated learning setups. Dec-FedTrack, by integrating local updates and gradient tracking mechanisms, aims to enhance robustness against universal adversarial perturbations while efficiently mitigating data heterogeneity. The theoretical analysis establishes convergence guarantees under certain assumptions, affirming Dec-FedTrack's reliability and efficacy. Our empirical evaluations demonstrate that for an equal adversary budget, Dec-FedTrack is more robust to adversarial perturbations compared to non-robust baselines.

APPENDIX

Proof of Lemma 3: According to Algorithm 1 we have

$$\begin{aligned}\mathbf{C}^{(t+1)}\mathbf{J} &= \mathbf{C}^{(t)}\mathbf{J} + \frac{1}{K\eta_c}\left(\mathbf{X}^{(t)} - \mathbf{X}^{(t)+K}\right)(\mathbf{W} - \mathbf{I})\mathbf{J} \\ &= \frac{1}{K\eta_c}\left(\mathbf{X}^{(t)} - \mathbf{X}^{(t)+K}\right)(\mathbf{I} - \mathbf{I}) = \mathbf{C}^{(t)}\mathbf{J}.\end{aligned}$$

Using the initialization assumption in (2), we have $\mathbf{C}^{(t)}\mathbf{J} = \mathbf{C}^{(0)}\mathbf{J} = \mathbf{0}$. Similarly, we have $\mathbf{D}^{(t)}\mathbf{J} = \mathbf{D}^{(0)}\mathbf{J} = \mathbf{0}$. □

Proof of Lemma 4: We can write

$$\begin{aligned}&\mathbb{E}\left\|\nabla_x f\left(\bar{\mathbf{x}}^{(t)}, \bar{\mathbf{y}}^{(t)}\right)\right\|^2 \\ &= \mathbb{E}\left\|\nabla_x f\left(\bar{\mathbf{x}}^{(t)}, \bar{\mathbf{y}}^{(t)}\right) - \nabla_x f\left(\bar{\mathbf{x}}^{(t)}, \hat{\mathbf{y}}^{(t)}\right)\right. \\ &\quad \left. + \nabla_x f\left(\bar{\mathbf{x}}^{(t)}, \hat{\mathbf{y}}^{(t)}\right)\right\|^2 \\ &\leq 2\ell^2 \mathbb{E}\left\|\bar{\mathbf{y}}^{(t)} - \hat{\mathbf{y}}^{(t)}\right\|^2 + 2\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2 \\ &= 2\ell^2\delta_t + 2\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2.\end{aligned}$$

Moreover,

$$\begin{aligned}&\mathbb{E}\left\|\nabla_y f\left(\bar{\mathbf{x}}^{(t)}, \bar{\mathbf{y}}^{(t)}\right)\right\|^2 \\ &= \mathbb{E}\left\|\nabla_y f\left(\bar{\mathbf{x}}^{(t)}, \bar{\mathbf{y}}^{(t)}\right) - \nabla_y f\left(\bar{\mathbf{x}}^{(t)}, \hat{\mathbf{y}}^{(t)}\right)\right\|^2 \leq \ell^2\delta_t. \quad (7)\end{aligned}$$

The equality in (7) holds due to the fact that $\nabla_y f\left(\bar{\mathbf{x}}^{(t)}, \hat{\mathbf{y}}^{(t)}\right) = 0$. □

Proof of Lemma 7: First, we derive an upper bound on the averaged progress for variable $\mathbf{x}$ as follows

$$\begin{aligned}
&\mathbb{E}\left\|\bar{\mathbf{x}}^{(t+1)} - \bar{\mathbf{x}}^{(t)}\right\|^2 \\
&= \eta_x^2 \mathbb{E}\left\|\frac{1}{n}\sum_{i,k} \nabla_x F_i\left(\mathbf{x}_i^{(t)+k}, \mathbf{y}_i^{(t)+k}; \xi_i^{(t)+k}\right) + \frac{K}{n}\sum_i \mathbf{c}_i^{(t)}\right\|^2 \\
&\overset{(a)}{\leq} \frac{2K\eta_x^2}{n}\sum_{i,k}\mathbb{E}\left\|\nabla_x f_i\left(\mathbf{x}_i^{(t)+k}, \mathbf{y}_i^{(t)+k}\right) - \nabla_x f_i\left(\bar{\mathbf{x}}_i^{(t)}, \bar{\mathbf{y}}_i^{(t)}\right)\right\|^2 \\
&\quad + 2K^2\eta_x^2\mathbb{E}\left\|\nabla_x f\left(\bar{\mathbf{x}}_i^{(t)}, \bar{\mathbf{y}}_i^{(t)}\right)\right\|^2 + \frac{K\eta_x^2\sigma^2}{n} \\
&\leq \frac{2K\eta_x^2\ell^2}{n}\sum_{i,k}\left(\mathbb{E}\left\|\mathbf{x}_i^{(t)+k} - \bar{\mathbf{x}}^{(t)}\right\|^2 + \mathbb{E}\left\|\mathbf{y}_i{}^{(t)+k} - \bar{\mathbf{y}}^{(t)}\right\|^2\right) \\
&\quad + 2K^2\eta_x^2\mathbb{E}\left\|\nabla_x f\left(\bar{\mathbf{x}}_i{}^{(t)}, \bar{\mathbf{y}}_i^{(t)}\right)\right\|^2 + \frac{K\eta_x^2\sigma^2}{n} \\
&\overset{(b)}{\leq} 2K\eta_x^2\ell^2\left(\mathcal{E}_t^x + \mathcal{E}_t^y\right) \\
&\quad + 2K^2\eta_x^2\left(2\ell^2\delta_t + 2\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2\right) + \frac{K\eta_x^2\sigma^2}{n}. \qquad (8)
\end{aligned}$$

Similar to the above derivations, we have

$$\begin{aligned}
\mathbb{E}\left\|\bar{\mathbf{y}}^{(t+1)} - \bar{\mathbf{y}}^{(t)}\right\|^2 &\leq 2K^2\eta_y^2\ell^2\left(\mathcal{E}_t^x + \mathcal{E}_t^y\right) \\
&\quad + 2K^2\eta_y^2\mathbb{E}\left\|\nabla_y f\left(\bar{\mathbf{x}}_i^{(t)}, \bar{\mathbf{y}}_i^{(t)}\right)\right\|^2 + \frac{K\eta_y^2\sigma^2}{n} \\
&\overset{(c)}{\leq} 2K\eta_y^2\ell^2\left(\mathcal{E}_t^x + \mathcal{E}_t^y\right) + 2K^2\eta_y^2\ell^2\delta_t + \frac{K\eta_y^2\sigma^2}{n}. \qquad (9)
\end{aligned}$$

We used Lemma 3, 4, and 4 in (a), (b), and (c), respectively. Combining (8) and (9) completes the proof. □

Proof of Lemma 8: We can write that

$$\begin{aligned}
&n\ell^2\gamma_{t+1}^x := \mathbb{E}\left\|\mathbf{C}^{(t+1)} + \nabla_x f\left(\bar{\mathbf{X}}^{(t+1)}, \bar{\mathbf{Y}}^{(t+1)}\right)(\mathbf{I}-\mathbf{J})\right\|_F^2 \\
&= \mathbb{E}\left\|\mathbf{C}^{(t)}\mathbf{W} \right. \\
&\quad + \frac{1}{K}\sum_{k=0}^{K-1}\nabla_x F\left(\mathbf{X}^{(t)+k}, \mathbf{Y}^{(t)+k}; \xi^{(t)+k}\right)(\mathbf{W}-\mathbf{I}) \\
&\quad \left. + \nabla_x f\left(\bar{\mathbf{X}}^{(t+1)}, \bar{\mathbf{Y}}^{(t+1)}\right)(\mathbf{I}-\mathbf{J})\right\|_F^2 \\
&\leq \mathbb{E}\left\|\left(\mathbf{C}^{(t)} + \nabla_x f\left(\bar{\mathbf{X}}^{(t)}, \bar{\mathbf{Y}}^{(t)}\right)(\mathbf{I}-\mathbf{J})\right)\mathbf{W}\right. \\
&\quad + \left(\frac{1}{K}\sum_{k=0}^{K-1}\nabla_x f\left(\mathbf{X}^{(t)+k}, \mathbf{Y}^{(t)+k}\right)\right. \\
&\quad \left.\left. - \nabla_x f\left(\bar{\mathbf{X}}^{(t)}, \bar{\mathbf{Y}}^{(t)}\right)\right)(\mathbf{W}-\mathbf{I}) \right. \\
&\quad \left. + \left(\nabla_x f\left(\bar{\mathbf{X}}^{(t+1)}, \bar{\mathbf{Y}}^{(t+1)}\right) - \nabla_x f\left(\bar{\mathbf{X}}^{(t)}, \bar{\mathbf{Y}}^{(t)}\right)\right)(\mathbf{I}-\mathbf{J})\right\|_F^2 \\
&\quad + \frac{n\sigma^2}{K} \overset{(a)}{\leq} (1+\alpha)(1-p)n\ell^2\gamma_t^x \\
&\quad + 2\left(1+\frac{1}{\alpha}\right)\left(\|\mathbf{W}-\mathbf{I}\|^2\frac{\ell^2}{K}\sum_{k=0}^{K-1}\left(\mathbb{E}\left\|\mathbf{X}^{(t)+k} - \bar{\mathbf{X}}^{(t)}\right\|^2\right.\right. \\
&\quad \left. + \mathbb{E}\left\|\mathbf{Y}^{(t)+k} - \bar{\mathbf{Y}}^{(t)}\right\|^2\right) \\
&\quad + \|\mathbf{I}-\mathbf{J}\|^2 n\ell^2\left(\mathbb{E}\left\|\bar{\mathbf{x}}^{(t+1)} - \bar{\mathbf{x}}^{(t)}\right\|^2\right. \\
&\quad \left.\left. + \mathbb{E}\left\|\bar{\mathbf{y}}^{(t+1)} - \bar{\mathbf{y}}^{(t)}\right\|^2\right)\right) \\
&\quad + \frac{n\sigma^2}{K} \overset{\alpha=\frac{p}{2}, \frac{1}{p}\geq 1}{\leq} \left(1-\frac{p}{2}\right)n\ell^2\gamma_t^x + \frac{6}{p}\left(\frac{4\ell^2 n}{K}\left(\mathcal{E}_t^x + \mathcal{E}_t^y\right)\right. \\
&\quad \left. + n\ell^2\left(\Delta_{t+1}^x + \Delta_{t+1}^y\right)\right) + \frac{n\sigma^2}{K}.
\end{aligned}$$

In (a) we applied Assumption 1 and the fact that

$$\begin{aligned}
&\left(\mathbf{C}^{(t)} + \nabla_x f\left(\bar{\mathbf{X}}^{(t)}, \bar{\mathbf{Y}}^{(t)}\right)(\mathbf{I}-\mathbf{J})\right)\mathbf{J} = \mathbf{C}^{(t)}\mathbf{J} \\
&\qquad + \nabla_x f\left(\bar{\mathbf{X}}^{(t)}, \bar{\mathbf{Y}}^{(t)}\right)(\mathbf{J}-\mathbf{J}) \overset{\text{Lemma 3}}{=} \mathbf{0}.
\end{aligned}$$

Using Lemma 7 to bound $\Delta_{t+1}^x + \Delta_{t+1}^y$ we have

$$\begin{aligned}
&\gamma_{t+1}^x \\
&\leq \left(1-\frac{p}{2}\right)\gamma_t^x + \frac{1}{p}\left(\frac{24}{K} + 12K\eta_x^2\ell^2 + 12K\eta_y^2\ell^2\right)\left(\mathcal{E}_t^x + \mathcal{E}_t^y\right) \\
&\quad + \frac{12K^2\ell^2}{p}\left(2\eta_x^2 + \eta_y^2\right)\delta_t + \frac{24K^2\eta_x^2}{p}\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2 \\
&\quad + \frac{6K\sigma^2\left(\eta_x^2+\eta_y^2\right)}{np} + \frac{\sigma^2}{K\ell^2}.
\end{aligned}$$

Applying the conditions on the step sizes will result in (3). In a similar fashion, we can show (4). □

Proof of Lemma 9: If we replace $\mathbf{x} = \bar{\mathbf{x}}^{(t)}$, $\mathbf{y} = \bar{\mathbf{y}}^{(t)}$, and $\mathbf{y}' = \hat{\mathbf{y}}^{(t)}$ in Proposition 2, we have

$$\begin{aligned}
&\nabla_y f(\bar{\mathbf{x}}^{(t)}, \bar{\mathbf{y}}^{(t)})^\top(\bar{\mathbf{y}}^{(t)} - \hat{\mathbf{y}}^{(t)}) + \frac{1}{2\ell}\left\|\nabla_y f(\bar{\mathbf{x}}^{(t)}, \bar{\mathbf{y}}^{(t)})\right\|^2 \\
&\qquad + \frac{\mu}{2}\|\bar{\mathbf{y}}^{(t)} - \hat{\mathbf{y}}^{(t)}\|^2 \leq 0. \qquad (10)
\end{aligned}$$

We can also write that

$$\begin{aligned}
&\mathbb{E}\left\|\hat{\mathbf{y}}^{(t)} - \bar{\mathbf{y}}^{(t)} - K\eta_y\nabla_y f\left(\bar{\mathbf{x}}^{(t)}, \bar{\mathbf{y}}^{(t)}\right)\right\|^2 = \mathbb{E}\left\|\hat{\mathbf{y}}^{(t)} - \bar{\mathbf{y}}^{(t)}\right\|^2 \\
&\quad - 2K\eta_y\mathbb{E}\left\langle\hat{\mathbf{y}}^{(t)} - \bar{\mathbf{y}}^{(t)}, \nabla y f\left(\bar{\mathbf{x}}^{(t)}, \bar{\mathbf{y}}^{(t)}\right)\right\rangle \\
&\quad + K^2\eta_y^2\mathbb{E}\left\|\nabla_y f\left(\bar{\mathbf{x}}^{(t)}, \bar{\mathbf{y}}^{(t)}\right)\right\|^2 = \mathbb{E}\left\|\hat{\mathbf{y}}^{(t)} - \bar{\mathbf{y}}^{(t)}\right\| \\
&\quad + 2K\eta_y\left(\mathbb{E}\left\langle\bar{\mathbf{y}}^{(t)} - \hat{\mathbf{y}}(t), \nabla_y f\left(\bar{\mathbf{x}}^{(t)}, \bar{\mathbf{y}}^{(t)}\right)\right\rangle\right. \\
&\quad \left. + \frac{K\eta_y}{2}\mathbb{E}\left\|\nabla_y f\left(\bar{\mathbf{x}}^{(t)}, \bar{\mathbf{y}}^{(t)}\right)\right\|^2\right) \overset{(a)}{\leq} \mathbb{E}\left\|\hat{\mathbf{y}}^{(t)} - \bar{\mathbf{y}}^{(t)}\right\|^2 \\
&\quad + 2K\eta_y\left(-\frac{\mu}{2}\mathbb{E}\left\|\hat{\mathbf{y}}^{(t)} - \bar{\mathbf{y}}^{(t)}\right\|^2\right) = (1-K\eta_y\mu)\,\delta_t.
\end{aligned}$$

In (a), we used the assumption that $\eta_y \leq \frac{1}{K\ell}$ and (10). Now, we can write

$$\begin{aligned}
&\mathbb{E}\left\|\hat{\mathbf{y}}^{(t)} - \bar{\mathbf{y}}^{(t+1)}\right\|^2 \overset{(b)}{=} \mathbb{E}\left\|\hat{\mathbf{y}}^{(t)} - \bar{\mathbf{y}}^{(t)}\right. \\
&\quad \left. - \frac{\eta_y}{n}\sum_{i,k}\nabla_y F_i\left(\mathbf{x}_i^{(t)+k}, \mathbf{y}_i^{(t)+k}; \xi_i^{(t)+k}\right)\right\|^2 \\
&\leq \mathbb{E}\left\|\hat{\mathbf{y}}^{(t)} - \bar{\mathbf{y}}^{(t)} - K\eta_y\nabla_y f\left(\bar{\mathbf{x}}^{(t)}, \bar{\mathbf{y}}^{(t)}\right)\right.
\end{aligned}$$

$$
\begin{aligned}
&-\frac{\eta_y}{n}\sum_{i,k}\nabla_y f_i\left(\mathbf{x}_i^{(t)+k},\mathbf{y}_i^{(t)+k}\right)\\
&+\frac{\eta_y}{n}\sum_{i,k}\nabla_y f_i\left(\bar{\mathbf{x}}^{(t)},\bar{\mathbf{y}}^{(t)}\right)\|^2\\
&+\frac{K\eta_y^2\sigma^2}{n}\leq(1+\alpha)\mathbb{E}\left\|\hat{\mathbf{y}}^{(t)}-\bar{\mathbf{y}}^{(t)}\right.\\
&\left.-K\eta_y\nabla_y f\left(\bar{\mathbf{x}}^{(t)},\bar{\mathbf{y}}^{(t)}\right)\right\|^2\\
&+\frac{K\eta_y^2\sigma^2}{n}+\left(1+\frac{1}{\alpha}\right)\frac{\eta_y^2K}{n}\sum_{i,k}\mathbb{E}\left\|\nabla_y f_i\left(\mathbf{x}_i^{(t)+k},y_i^{(t)+k}\right)\right.\\
&\left.-\nabla_y f_i\left(\bar{\mathbf{x}}^{(t)},\bar{\mathbf{y}}^{(t)}\right)\right\|^2\leq(1+\alpha)\left(1-K\eta_y\mu\right)\delta_t\\
&+\left(1+\frac{1}{\alpha}\right)\eta_y^2\ell^2K\left(\mathcal{E}^x+\mathcal{E}^y\right)+\frac{K\eta_y^2\sigma^2}{n},
\end{aligned}
$$

where in (b), we used Lemma 3; i.e., $\frac{1}{n}\sum_i\mathbf{d}_i^{(t)}=\mathbf{0}$. □

Proof of Lemma 10: We begin the proof by writing that

$$
\begin{aligned}
\delta_{t+1}&\overset{(a)}{\leq}(1+\beta)\mathbb{E}\left\|\hat{\mathbf{y}}^{(t)}-\bar{\mathbf{y}}^{(t+1)}\right\|^2\\
&+\left(1+\frac{1}{\beta}\right)\mathbb{E}\left\|\hat{\mathbf{y}}^{(t+1)}-\hat{\mathbf{y}}^{(t)}\right\|^2\\
&\leq(1+\beta)(1+\alpha)\left(1-K\eta_y\mu\right)\delta_t\\
&+(1+\beta)\left(1+\frac{1}{\alpha}\right)\eta_y^2\ell^2K\left(\mathcal{E}_t^x+\mathcal{E}_t^y\right)\\
&+\left(1+\frac{1}{\beta}\right)\kappa^2\mathbb{E}\left\|\bar{\mathbf{x}}^{(t+1)}-\bar{\mathbf{x}}^{(t)}\right\|^2+(1+\beta)\frac{K\eta_y^2\sigma^2}{n}\\
&\overset{(b)}{\leq}\left(1-\frac{K\eta_y\mu}{3}\right)\delta_t+\frac{6\eta_y\ell^2}{\mu}\left(\mathcal{E}_t^x+\mathcal{E}_t^y\right)+\frac{4\eta_y\sigma^2}{n\mu}\\
&+\frac{4\kappa^2}{K\eta_y\mu}\left(2K\eta_x^2\ell^2\left(\mathcal{E}_t^x+\mathcal{E}_t^y\right)\right.\\
&\left.+4K^2\ell^2\eta_x^2\delta_t+4K^2\eta_x^2\mathbb{E}\|\nabla\Phi(\bar{\mathbf{x}}(t))\|^2+\frac{K\eta_x^2\sigma^2}{n}\right)\\
&=\left(1-\frac{K\eta_y\ell}{3\kappa}+\frac{16\ell\kappa^3K\eta_x^2}{\eta_y}\right)\delta_t\\
&+\left(\frac{8\ell\kappa^3\eta_x^2}{\eta_y}+6\eta_y\ell\kappa\right)\left(\mathcal{E}_t^x+\mathcal{E}_t^y\right)\\
&+\frac{16\kappa^3K\eta_x^2}{\eta_y\ell}\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2+\frac{4\kappa^3\eta_x^2\sigma^2}{n\eta_y\ell}+\frac{4\eta_y\sigma^2\kappa}{n\ell}.
\end{aligned}
$$

Using the assumption $\eta_x\leq\frac{\eta_y}{4\sqrt{6}\kappa^2}$ compeletes the proof. In (a), we used the bound in Lemma 9 for the first term and Proposition 1 for the second term. In (b), we replaced $\alpha=\beta=\frac{K\eta_y\mu}{3}$ and used (8) in Lemma 7. □

Proof of Lemma 11: According to the Proposition 1, $\Phi(\cdot)$ is $2\kappa\ell$-smooth, which results in the following, as shown in the equation at the top of the next page. Now, we derive an upper bound for U as follows

$$
\begin{aligned}
U:=&\\
&\mathbb{E}\left\langle\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right),-\frac{\eta_x}{n}\sum_{i,k}\left(\nabla_x F_i\left(\mathbf{x}_i^{(t)+k},\mathbf{y}_i^{(t)+k};\xi_i^{(t)+k}\right)\right.\right.\\
&\left.\left.+\mathbf{c}_i^{(t)}\right)\right\rangle=\mathbb{E}\left\langle\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right),\right.\\
&\left.-\frac{\eta_x}{n}\sum_{i,k}\mathbb{E}_{\xi_i^{(t)+k}}\nabla_x F_i\left(\mathbf{x}_i^{(t)+k},\mathbf{y}_i^{(t)+k};\xi_i^{(t)+k}\right)\right\rangle\\
&=-\eta_x\mathbb{E}\left\langle\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right),\frac{1}{n}\sum_{i,k}\left(\nabla_x f_i\left(\mathbf{x}_i^{(t)+k},\mathbf{y}_i^{(t)+k}\right)\right.\right.\\
&-\nabla_x f_i\left(\bar{\mathbf{x}}^{(t)},\bar{\mathbf{y}}^{(t)}\right)+\nabla_x f_i\left(\bar{\mathbf{x}}^{(t)},\bar{\mathbf{y}}^{(t)}\right)\\
&-\nabla_x f_i\left(\bar{\mathbf{x}}^{(t)},\hat{\mathbf{y}}^{(t)}\right)\\
&\left.\left.+\nabla_x f_i\left(\bar{\mathbf{x}}^{(t)},\hat{\mathbf{y}}^{(t)}\right)\right)\right\rangle=-K\eta_x\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2\\
&-\frac{\eta_x}{n}\sum_{i,k}\left\langle\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right),\nabla_x f_i\left(\mathbf{x}_i^{(t)+k},\mathbf{y}_i^{(t)+k}\right)\right.\\
&-\nabla_x f_i\left(\bar{\mathbf{x}}^{(t)},\bar{\mathbf{y}}^{(t)}\right)+\nabla_x f_i\left(\bar{\mathbf{x}}^{(t)},\bar{\mathbf{y}}^{(t)}\right)\\
&\left.-\nabla_x f_i\left(\bar{\mathbf{x}}^{(t)},\hat{\mathbf{y}}^{(t)}\right)\right\rangle\leq-\frac{K\eta_x}{2}\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2\\
&+\frac{\eta_x}{n}\sum_{i,k}\left(\mathbb{E}\left\|\nabla_x f_i\left(\mathbf{x}_i^{(t)+k},\mathbf{y}_i^{(t)+k}\right)-\nabla_x f_i\left(\bar{\mathbf{x}}^{(t)},\bar{\mathbf{y}}^{(t)}\right)\right\|^2\right.\\
&\left.+\mathbb{E}\left\|\nabla_x f_i\left(\bar{\mathbf{x}}^{(t)},\bar{\mathbf{y}}^{(t)}\right)-\nabla_x f_i\left(\bar{\mathbf{x}}^{(t)},\hat{\mathbf{y}}^{(t)}\right)\right\|^2\right)\\
&\leq-\frac{K\eta_x}{2}\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2+\eta_x\ell^2\left(\mathcal{E}_t^x+\mathcal{E}_t^y\right)+K\eta_x\ell^2\delta_t.
\end{aligned}
$$

Now, we apply the above upper bound for U and (8) in Lemma 7 as follows

$$
\begin{aligned}
\mathbb{E}\Phi\left(\bar{\mathbf{x}}^{(t+1)}\right)&\leq\mathbb{E}\Phi\left(\bar{\mathbf{x}}^{(t)}\right)+\eta_x\ell^2\left(\mathcal{E}_t^x+\mathcal{E}_t^y\right)+\ell^2\eta_xK\delta_t\\
&-\frac{\eta_xK}{2}\mathbb{E}\left\|\nabla\phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2+\kappa\ell\mathbb{E}\left\|\bar{\mathbf{x}}^{(t+1)}-\bar{\mathbf{x}}^{(t)}\right\|^2\\
&\leq\mathbb{E}\Phi\left(\bar{\mathbf{x}}^{(t)}\right)+\left(\eta_x\ell^2+2K\eta_x^2\ell^3\kappa\right)\left(\mathcal{E}_t^x+\mathcal{E}_t^y\right)+\frac{K\eta_x^2\ell\kappa\sigma^2}{n}\\
&+\left(\ell^2\eta_xK+4K^2\ell^3\eta_x^2\kappa\right)\delta_t\\
&+\left(4K^2\eta_x^2\ell\kappa-\frac{\eta_xK}{2}\right)\mathbb{E}\left\|\nabla\phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2.
\end{aligned}
$$

Applying the assumption $\eta_x\leq\frac{1}{16K\ell\kappa}$ completes the proof. □

Proof of Lemma 12: According to the Lemma 5, we have

$$
\begin{aligned}
0\leq&-D_x\ell^2\eta_d\mathcal{E}_t^x+3D_xK\ell^2\eta_d\Xi_t^x+12D_xK^2\eta_c^2\eta_d\ell^4\mathcal{E}_t^y\\
&+12D_xK^3\eta_c^2\eta_d\ell^4\gamma_t^x+12D_xK^3\eta_c^2\eta_d\ell^4\delta_t\\
&+12D_xK^3\eta_c^2\eta_d\ell^2\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2+3D_xK^2\eta_c^2\eta_d\ell^2\sigma^2,\\
0\leq&-D_y\ell^2\eta_d\mathcal{E}_t^y+3D_yK\ell^2\eta_d\Xi_t^y+12D_yK^2\eta_d^3\ell^4\mathcal{E}_t^x\\
&+12D_yK^3\eta_d^3\ell^4\gamma_t^y+6D_yK^3\eta_d^3\ell^4\delta_t+3D_yK^2\eta_d^3\ell^2\sigma^2.
\end{aligned}
\tag{11}
$$

By applying the definition of $\mathcal{H}_t$ from (5) and using (11), we have

$$
\begin{aligned}
\mathcal{H}_{t+1}-\mathcal{H}_t\leq&\underbrace{\left(-B_x\frac{p}{2}+A_x\frac{6\eta_s^2}{p}+D_x12\right)}_{\leq D_1}\eta_d^3K^3\ell^4\gamma_t^x\\
&+\underbrace{\left(-B_y\frac{p}{2}+A_y\frac{6\eta_r^2}{p}+D_y12\right)}_{\leq D_2}\eta_d^3K^3\ell^4\gamma_t^y
\end{aligned}
$$

$$\mathbb{E}\Phi\left(\bar{\mathbf{x}}^{(t+1)}\right) = \mathbb{E}\Phi\left(\bar{\mathbf{x}}^{(t)} - \frac{\eta_x}{n}\sum_{i,k}\left(\nabla_x F_i\left(\mathbf{x}_i^{(t)+k}, \mathbf{y}_i^{(t)+k}; \xi_i^{(t)+k}\right) + \mathbf{c}_i^{(t)}\right)\right) \leq \mathbb{E}\Phi\left(\bar{\mathbf{x}}^{(t)}\right)$$
$$+ \underbrace{\mathbb{E}\left\langle \nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right), \frac{-\eta_x}{n}\sum_{i,k}\left(\nabla_x F_i\left(\mathbf{x}_i^{(t)+k}, \mathbf{y}_i^{(t)+k}; \xi_i^{(t)+k}\right) + \mathbf{c}_i^{(t)}\right)\right\rangle}_{:=U} + \kappa\ell\mathbb{E}\left\|\bar{\mathbf{x}}^{(t+1)} - \bar{\mathbf{x}}^{(t)}\right\|^2.$$

$$+ \underbrace{\left(-A_x\frac{p}{2} + 3D_x\right)}_{\leq D_3}\Xi_t^x\eta_d K\ell^2$$
$$+ \underbrace{\left(-A_y\frac{p}{2} + 3D_y\right)}_{\leq D_4}\Xi_t^y\eta_d K\ell^2$$
$$+ \left(-D_x + A_x\frac{6K^2\ell^2\eta_x^2}{p} + A_y\frac{6K^2\ell^2\eta_y^2}{p} + B_x\frac{25\eta_d^2\ell^2K^2}{p}\ q\right.$$
$$\left. + B_y\frac{25\eta_d^2\ell^2K^2}{p} + D_y 12K^2\ell^2\eta_d^2 + \frac{2\eta_x}{\eta_y} + C\frac{12\eta_r}{p}\right)\ell^2\eta_d\mathcal{E}_t^x$$
$$+ \left(-D_y + A_x\frac{6K^2\ell^2\eta_x^2}{p} + A_y\frac{6K^2\ell^2\eta_y^2}{p} + B_x\frac{25\eta_d^2\ell^2K^2}{p}\right.$$
$$\left. + B_y\frac{25\eta_d^2\ell^2K^2}{p} + D_x 12K^2\ell^2\eta_c^2 + \frac{2\eta_x}{\eta_y} + C\frac{12\eta_r}{p}\right)\ell^2\eta_d\mathcal{E}_t^y$$
$$+ \left(-C\frac{\eta_r}{6p} + B_x\frac{12K^4\ell^4}{p}\eta_d^2\left(3\eta_y^2\right)\kappa^2\right.$$
$$+ B_y\frac{12K^4\ell^4}{p}\eta_d^2\left(3\eta_y^2\right)\kappa^2$$
$$\left. + D_x 12K^2\ell^2\eta_c^2\kappa^2 + D_y 6K^2\ell^2\eta_d^2\kappa^2 + 2\kappa^2\frac{\eta_x}{\eta_d}\right)\frac{K\ell^2\eta_d}{\kappa^2}\delta_t$$
$$+ \left(-\frac{1}{4} + B_x\frac{24K^4\ell^4}{p}\eta_d^3\eta_x + B_y\frac{24K^4\ell^4}{p}\eta_d^3\eta_x + C\frac{16\kappa^2\eta_x}{\eta_y p}\right.$$
$$\left. + D_x 12K^2\ell^2\eta_d\frac{\eta_c}{\eta_s}\right)K\eta_x\mathbb{E}\left\|\nabla\Phi\left(\bar{\mathbf{x}}^{(t)}\right)\right\|^2$$
$$+ \underbrace{\left(A_x\eta_s^2 + A_y\eta_r^2 + B_x 2 + B_y 2 + D_x 3 + D_y 3\right)}_{\leq D_9}K^2\ell^2\eta_d^3\sigma^2$$
$$+ \frac{K\eta_x^2\ell\kappa}{n}\sigma^2 + C\frac{8\eta_y}{np}\sigma^2.$$

Let us denote the fifth, sixth, seventh, and eighth parentheses above as D_5, D_6, D_7, and D_8, respectively. Assuming that $D_x = D_y = v$, as long as $\eta_d \leq \frac{p}{200v\kappa K\ell}$, $\eta_c \leq \frac{\eta_d}{\kappa^2}$, $\eta_s = \eta_r = pv$, $A_x = A_y = \frac{6v}{p}$, $B_x = B_y = \frac{1}{p}(72v^3 + 24v)$, and $C = \frac{1}{24}$, there exists $v > 1$ that makes D_1, D_2, D_3, D_4, D_5, D_6, $D_7 \leq 0$, $D_8 \leq -D < 0$, and $D_9 \geq 0$. □

REFERENCES

[1] S. Ghiasvand, A. Reisizadeh, M. Alizadeh, and R. Pedarsani, "Communication-efficient and decentralized federated minimax optimization," in *Proc. 60th Annu. Allerton Conf. Commun., Control, Comput.*, Sep. 2024, pp. 1–7.

[2] J. Konečný, 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] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, nos. 1–2, pp. 1–210, Jun. 2021.

[4] S. Nabavirazavi, R. Taheri, and S. S. Iyengar, "Enhancing federated learning robustness through randomization and mixture," *Future Gener. Comput. Syst.*, vol. 158, pp. 28–43, 2024.

[5] S. Nabavirazavi, R. Taheri, M. Shojafar, and S. S. Iyengar, "Impact of aggregation function randomization against model poisoning in federated learning," in *Proc. IEEE 22nd Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Nov. 2023, pp. 165–172.

[6] M. Saberi, C. Zhang, and M. Akcakaya, "Training-free mitigation of adversarial attacks on deep learning-based MRI reconstruction," 2025, *arXiv:2501.01908*.

[7] F. Farnia, A. Reisizadeh, R. Pedarsani, and A. Jadbabaie, "An optimal transport approach to personalized federated learning," *IEEE J. Sel. Areas Inf. Theory*, vol. 3, no. 2, pp. 162–171, Jun. 2022.

[8] A. Reisizadeh, F. Farnia, R. Pedarsani, and A. Jadbabaie, "Robust federated learning: The case of affine distribution shifts," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2020, pp. 21554–21565.

[9] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons, "The non-iid data quagmire of decentralized machine learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4387–4398.

[10] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, vol. 2, 2020, pp. 429–450.

[11] G. Zhou et al., "FedPAGE: Pruning adaptively toward global efficiency of heterogeneous federated learning," *IEEE/ACM Trans. Netw.*, vol. 32, no. 3, pp. 1873–1887, Jun. 2024.

[12] A. Rodio, F. Faticanti, O. Marfoq, G. Neglia, and E. Leonardi, "Federated learning under heterogeneous and correlated client availability," *IEEE/ACM Trans. Netw.*, vol. 32, no. 2, pp. 1–10, May 2023.

[13] L. Wang et al., "BOSE: Block-wise federated learning in heterogeneous edge computing," *IEEE/ACM Trans. Netw.*, vol. 32, no. 2, pp. 1362–1377, Apr. 2024.

[14] J. Liu et al., "Federated learning with experience-driven model migration in heterogeneous edge networks," *IEEE/ACM Trans. Netw.*, vol. 32, no. 4, pp. 3468–3484, Aug. 2024.

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

[16] S. Pu and A. Nedić, "Distributed stochastic gradient tracking methods," *Math. Program.*, vol. 187, pp. 409–457, May 2021.

[17] J. Zhang and K. You, "Decentralized stochastic gradient tracking for non-convex empirical risk minimization," 2019, *arXiv:1909.02712*.

[18] M. Ebrahimi, U. V. Shanbhag, and F. Yousefian, "Distributed gradient tracking methods with guarantees for computing a solution to stochastic MPECs," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2024, pp. 2182–2187.

[19] C. Chen, J. Zhang, L. Shen, P. Zhao, and Z. Luo, "Communication efficient primal-dual algorithm for nonconvex nonsmooth distributed optimization," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 1594–1602.

[20] H. Hendrikx, F. Bach, and L. Massoulié, "An optimal algorithm for decentralized finite-sum optimization," *SIAM J. Optim.*, vol. 31, no. 4, pp. 2753–2783, Jan. 2021.

[21] D. Kovalev, A. Salim, and P. Richtárik, "Optimal and practical algorithms for smooth and strongly convex decentralized optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, Jan. 2020, pp. 18342–18352.

[22] B. Li, S. Cen, Y. Chen, and Y. Chi, "Communication-efficient distributed optimization in networks with gradient tracking and variance reduction," *J. Mach. Learn. Res.*, vol. 21, pp. 1–51, Jan. 2020.

[23] B. Li, Z. Li, and Y. Chi, "DESTRESS: Computation-optimal and communication-efficient decentralized nonconvex finite-sum optimization," *SIAM J. Math. Data Sci.*, vol. 4, no. 3, pp. 1031–1051, Sep. 2022.

[24] H. Li, Z. Lin, and Y. Fang, "Variance reduced EXTRA and DIGing and their optimal acceleration for strongly convex decentralized optimization," *J. Mach. Learn. Res.*, vol. 23, no. 222, pp. 1–41, 2022.

[25] H. Sun, S. Lu, and M. Hong, "Improving the sample and communication complexity for decentralized non-convex optimization: Joint gradient estimation and tracking," in *Proc. Int. Conf. Mach. Learn.*, vol. 1, Jul. 2020, pp. 9217–9228.

[26] C. A. Uribe, S. Lee, A. Gasnikov, and A. Nedic, "A dual approach for optimal algorithms in distributed optimization over networks," in *Proc. Inf. Theory Appl. Workshop (ITA)*, Feb. 2020, pp. 1–37.

[27] Z. Wang et al., "Distributed stochastic consensus optimization with momentum for nonconvex nonsmooth problems," *IEEE Trans. Signal Process.*, vol. 69, pp. 4486–4501, 2021.

[28] R. Xin, U. A. Khan, and S. Kar, "Fast decentralized nonconvex finite-sum optimization with recursive variance reduction," *SIAM J. Optim.*, vol. 32, no. 1, pp. 1–28, Mar. 2022.

[29] J. Liu, J. Liu, H. Xu, Y. Liao, Z. Wang, and Q. Ma, "YOGA: Adaptive layer-wise model aggregation for decentralized federated learning," *IEEE/ACM Trans. Netw.*, vol. 32, no. 2, pp. 1768–1780, Apr. 2024.

[30] Y. Liu, T. Lin, A. Koloskova, and S. U. Stich, "Decentralized gradient tracking with local steps," *Optim. Methods Softw.*, pp. 1–28, Mar. 2024.

[31] E. D. Hien Nguyen, S. A. Alghunaim, K. Yuan, and C. A. Uribe, "On the performance of gradient tracking with local updates," in *Proc. 62nd IEEE Conf. Decis. Control (CDC)*, Dec. 2023, pp. 4309–4313.

[32] F. Haddadpour, M. M. Kamani, A. Mokhtari, and M. Mahdavi, "Federated learning with compression: Unified analysis and sharp guarantees," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2021, pp. 2350–2358.

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

[34] R. Labaca-Castro, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2023, pp. 73–76.

[35] T. Lin, C. Jin, and M. Jordan, "On gradient descent ascent for nonconvex-concave minimax problems," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 6083–6093.

[36] S. Qiu, Z. Yang, X. Wei, J. Ye, and Z. Wang, "Single-timescale stochastic nonconvex-concave optimization for smooth nonlinear TD learning," 2020, *arXiv:2008.10103*.

[37] L. Luo, H. Ye, Z. Huang, and T. Zhang, "Stochastic recursive gradient descent ascent for stochastic Nonconvex-strongly-concave minimax problems," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, Jan. 2020, pp. 20566–20577.

[38] S. Zhang, J. Yang, C. Guzmán, N. Kiyavash, and N. He, "The complexity of nonconvex-strongly-concave minimax optimization," in *Proc. Uncertainty Artif. Intell.*, Jan. 2021, pp. 482–492.

[39] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5149–5164, Oct. 2015.

[40] D. Mateos-Núñez and J. Cortés, "Distributed subgradient methods for saddle-point problems," in *Proc. 54th IEEE Conf. Decis. Control (CDC)*, Dec. 2015, pp. 5462–5467.

[41] A. Rogozin, A. Beznosikov, D. Dvinskikh, D. Kovalev, P. Dvurechensky, and A. Gasnikov, "Decentralized saddle point problems via non-Euclidean mirror prox," *Optim. Methods Softw.*, pp. 1–26, Jan. 2024.

[42] A. Beznosikov, G. Scutari, A. Rogozin, and A. Gasnikov, "Distributed saddle-point problems under data similarity," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, May 2021, pp. 8172–8184.

[43] A. Beznosikov, A. Rogozin, D. Kovalev, and A. Gasnikov, "Near-optimal decentralized algorithms for saddle point problems over time-varying networks," in *Proc. Int. Conf. Optim. Appl.*, Petrovac, Montenegro. Cham, Switzerland: Springer, Jan. 2021, pp. 246–257.

[44] X. Wu, Z. Hu, and H. Huang, "Decentralized Riemannian algorithm for nonconvex minimax problems," 2023, *arXiv:2302.03825*.

[45] Z. Liu, X. Zhang, S. Lu, and J. Liu, "PRECISION: Decentralized constrained min-max learning with low communication and sample complexities," 2023, *arXiv:2303.02532*.

[46] Y. Xu, "Decentralized gradient descent maximization method for composite nonconvex strongly-concave minimax problems," 2023, *arXiv:2304.02441*.

[47] H. Gao, "Decentralized stochastic gradient descent ascent for finite-sum minimax problems," 2022, *arXiv:2212.02724*.

[48] G. Mancino-Ball and Y. Xu, "Variance-reduced accelerated methods for decentralized stochastic double-regularized nonconvex strongly-concave minimax problems," 2023, *arXiv:2307.07113*.

[49] L. Chen, H. Ye, and L. Luo, "An efficient stochastic algorithm for decentralized nonconvex-strongly-concave minimax optimization," 2022, *arXiv:2212.02387*.

[50] W. Xian, F. Huang, Y. Zhang, and H. Huang, "A faster decentralized algorithm for nonconvex minimax problems," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, Dec. 2021, pp. 25865–25877.

[51] X. Zhang, G. Mancino-Ball, N. S. Aybat, and Y. Xu, "Jointly improving the sample and communication complexities in decentralized stochastic minimax optimization," 2023, *arXiv:2307.09421*.

[52] I. Tsaknakis, M. Hong, and S. Liu, "Decentralized min-max optimization: Formulations, algorithms and applications in network poisoning attack," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 5755–5759.

[53] M. Liu et al., "A decentralized parallel algorithm for training generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2019, pp. 11056–11070.

[54] A. Cutkosky and F. Orabona, "Momentum-based variance reduction in non-convex SGD," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2019, pp. 15236–15245.

[55] C. Fang, C. J. Li, Z. Lin, and T. Zhang, "SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, Jan. 2018, pp. 689–699.

[56] C. Hou, K. K. Thekumparampil, G. Fanti, and S. Oh, "Efficient algorithms for federated saddle point optimization," 2021, *arXiv:2102.06333*.

[57] L. Liao, L. Shen, J. Duan, M. Kolar, and D. Tao, "Local AdaGrad-type algorithm for stochastic convex-concave optimization," 2021, *arXiv:2106.10022*.

[58] Z. Sun and E. Wei, "A communication-efficient algorithm with linear convergence for federated minimax learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022, pp. 6060–6073.

[59] Y. Deng, M. M. Kamani, and M. Mahdavi, "Distributionally robust federated averaging," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2021, pp. 15111–15122.

[60] Y. Deng and M. Mahdavi, "Local stochastic gradient descent ascent: Convergence analysis and communication efficiency," in *Proc. Int. Conf. Artif. Intell. Statist.*, Jan. 2021, pp. 1387–1395.

[61] J. Xie, C. Zhang, Z. Shen, W. Liu, and H. Qian, "CDMA: A practical cross-device federated learning algorithm for general minimax problems," 2021, *arXiv:2105.14216*.

[62] P. Sharma, R. Panda, G. Joshi, and P. K. Varshney, "Federated minimax optimization: Improved convergence analyses and algorithms," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2022, pp. 19683–19730.

[63] P. Sharma, R. Panda, and G. Joshi, "Federated minimax optimization with client heterogeneity," 2023, *arXiv:2302.04249*.

[64] H. Yang, X. Zhang, Z. Liu, and J. Liu, "SAGDA: Achieving o (ε^{-2}) communication complexity in federated min-max learning," in *Proc. 36th Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 7142–7154.

[65] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1765–1773.

[66] H. E. Oskouie and F. Farnia, "Interpretation of neural networks is susceptible to universal adversarial perturbations," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[67] T. Lin, C. Jin, and M. I. Jordan, "Near-optimal algorithms for minimax optimization," in *Proc. Conf. Learn. Theory*, Jan. 2020, pp. 2738–2779.

[68] 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. 37th Int. Conf. Mach. Learn.*, vol. 119, Jul. 2020, pp. 5381–5393.

[69] S. Bubeck, "Convex optimization: Algorithms and complexity," *Found. Trends Mach. Learn.*, vol. 8, nos. 3–4, pp. 231–357, 2015.

[70] S. Ghiasvand, A. Reisizadeh, M. Alizadeh, and R. Pedarsani, "Robust decentralized learning with local updates and gradient tracking," 2024, *arXiv:2405.00965*.

[71] X. Zhang, Z. Liu, J. Liu, Z. Zhu, and S. Lu, "Taming communication and sample complexities in decentralized policy evaluation for cooperative multi-agent reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, Dec. 2021, pp. 18825–18838.

[72] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.

[73] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[74] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," 2016, *arXiv:1611.01236*.

[75] C. K. Mummadi, T. Brox, and J. H. Metzen, "Defending against universal perturbations with shared adversarial training," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4927–4936.

[76] A. Shafahi, M. Najibi, Z. Xu, J. P. Dickerson, L. S. Davis, and T. Goldstein, "Universal adversarial training," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 5636–5643.

Sajjad Ghiasvand received the B.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2023. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of California at Santa Barbara (UCSB), Santa Barbara, CA, USA. His research interests include federated and decentralized learning. He was a recipient of the Jack Lin Research Accelerator Fellowship at UCSB.

Amirhossein Reisizadeh received the M.S. degree in electrical engineering from the University of California at Los Angeles in 2016 and the Ph.D. degree in electrical engineering from the University of California at Santa Barbara in 2021. He is currently a Post-Doctoral Associate with the Laboratory for Information and Decision Systems (LIDS), Massachusetts Institute of Technology (MIT). In 2019, he was a Finalist in the Qualcomm Innovation Fellowship Program. His current research focuses on optimization for machine learning and deep learning theory.

Mahnoosh Alizadeh received the Ph.D. degree in electrical and computer engineering from the University of California at Davis in 2014. From 2014 to 2016, she was a Post-Doctoral Scholar with Stanford University. She is currently an Associate Professor of electrical and computer engineering with the University of California at Santa Barbara. Her research is focused on the design of network control, learning, and optimization algorithms for societal-scale cyber-physical systems. She was a recipient of the National Science Foundation CAREER Award in 2019. She serves as an Associate Editor for IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS and IEEE OPEN JOURNAL OF CONTROL SYSTEMS.

Ramtin Pedarsani (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Tehran, Tehran, Iran, in 2009, the M.Sc. degree in communication systems from the Swiss Federal Institute of Technology, Lausanne, Switzerland, in 2011, and the Ph.D. degree from the University of California at Berkeley, Berkeley, in 2015. He is currently an Associate Professor with the ECE Department, University of California at Santa Barbara, Santa Barbara, CA, USA. His research interests include machine learning, information theory, networks, and transportation systems. He was a recipient of the Communications Society and Information Theory Society Joint Paper Award in 2020, the Best Paper Award in the IEEE International Conference on Communications in 2014, and the NSF CRII Award in 2017.