FedLGA: Toward System-Heterogeneity of Federated Learning via Local Gradient Approximation

Xingyu Li, Zhe Qu, *Graduate Student Member, IEEE*, Bo Tang, *Senior Member, IEEE*, and Zhuo Lu, *Senior Member, IEEE*

***Abstract*—Federated learning (FL) is a decentralized machine learning architecture, which leverages a large number of remote devices to learn a joint model with distributed training data. However, the system-heterogeneity is one major challenge in an FL network to achieve robust distributed learning performance, which comes from two aspects: 1) device-heterogeneity due to the diverse computational capacity among devices and 2) data-heterogeneity due to the nonidentically distributed data across the network. Prior studies addressing the heterogeneous FL issue, for example, FedProx, lack formalization and it remains an open problem. This work first formalizes the system-heterogeneous FL problem and proposes a new algorithm, called federated local gradient approximation (FedLGA), to address this problem by bridging the divergence of local model updates via gradient approximation. To achieve this, FedLGA provides an alternated Hessian estimation method, which only requires extra linear complexity on the aggregator. Theoretically, we show that with a device-heterogeneous ratio ρ, FedLGA achieves convergence rates on non-i.i.d. distributed FL training data for the nonconvex optimization problems with $O([(1+\rho)/\sqrt{ENT}]+1/T)$ and $O([(1+\rho)\sqrt{E}/\sqrt{TK}]+1/T)$ for full and partial device participation, respectively, where E is the number of local learning epoch, T is the number of total communication round, N is the total device number, and K is the number of the selected device in one communication round under partially participation scheme. The results of comprehensive experiments on multiple datasets indicate that FedLGA can effectively address the system-heterogeneous problem and outperform current FL methods. Specifically, the performance against the CIFAR-10 dataset shows that, compared with FedAvg, FedLGA improves the model's best testing accuracy from 60.91% to 64.44%.**

***Index Terms*—Federated learning (FL), local gradient approximation, mobile-edge computing, nonconvex optimization.**

Manuscript received 20 July 2022; revised 7 December 2022 and 29 January 2023; accepted 17 February 2023. This work was supported by the U.S. National Science Foundation (NSF) under Award IIS-2047570 and Award CNS-2044516. This article was recommended by Associate Editor J. C.-W. Lin. *(Corresponding author: Bo Tang.)*

Xingyu Li is with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS 39762 USA (e-mail: xl292@msstate.edu).

Zhe Qu and Zhuo Lu are with the Department of Electrical Engineering, University of South Florida, Tampa, FL 33620 USA (e-mail: zhequ@usf.edu, zhuolu@usf.edu).

Bo Tang is with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609 USA (e-mail: btang1@wpi.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2023.3247365.

Digital Object Identifier 10.1109/TCYB.2023.3247365

I. Introduction

FEDERATED learning (FL) [1], [2], [3] has emerged as an attractive distributed machine learning paradigm that leverages remote devices to collaboratively learn a joint model with decentralized training data via the coordination of a centralized aggregator. Typically, the joint model is trained on all remote devices in the FL network to solve an optimization problem without exchanging their private training data, which distinguishes the FL paradigm from traditional centralized machine learning, and thus the data privacy can be greatly protected [2], [4], [5], [6], [7]. Specifically, due to the flexibility for remote device participation (e.g., mobile-edge computing), devices can randomly join or leave the federated network during the training process. This makes the full participation scheme infeasible as the network needs extra communication costs to wait for the slowest device, which dominates the bottleneck of FL [8], [9], [10]. As such, in recent FL algorithms, only a fixed subset of remote devices are chosen by the aggregator in each communication round, also known as the partial participation scheme [2], [11], [12].

However, in the current FL study, there is a fundamental gap that has not been seen in traditional centralized ML paradigms, known as the system heterogeneity issue. Specifically, we consider that the system-heterogeneous FL issue consists of two types of heterogeneity: 1) *data* and 2) *device*. The data heterogeneity is also known as the non-i.i.d. training dataset. As the training samples on the remote devices are collected by the devices themselves based on their unique environment, the data distribution can vary heavily between different remote devices. Although the optimization of non-i.i.d. FL has recently drawn significant attention, prior works have shown that compared to the i.i.d. setting, the performance of the joint model degrades significantly and remains an open problem [11], [13], [14].

The device-heterogeneity stems from the heterogeneous FL network, where remote devices are in large numbers and have a variety of computational capacities [12], [15]. Specifically, for the partially participated FL scheme where each remote learning process is usually limited to a responding time, the diverged computational capacity can lead to heterogeneous local training updates, for example, the remote device with limited computational capacity is only able to return a nonfinished update. To tackle this problem, several FL frameworks

have been studied in literature [8], [11], [16], [17], [18]. For example, FedProx [12] develops a broader framework over FedAvg [2], which provides a proximal term to the local objective of heterogeneous remote devices. However, most current works are developed on the side of remote devices, which requires extra computational costs that could worsen the divergence, and there is no widely accepted formulation provided.

In this article, we investigate the system-heterogeneous issue in FL. A more realistic FL scenario under the device heterogeneity is formulated, which synchronously learns the joint model on the aggregator with diverged local updates. Unlike most current FL approaches, our formulated scenario does not require remote devices to complete all local training epochs before the aggregation, but it leverages whatever their current training updates are at present. Particularly, different from the previous works that usually establish a communication response threshold in the partial participation scheme, the formulated system-heterogeneous FL provides a guarantee that each remote device shares the same probability of being chosen for the training process.

Then, the biggest challenge to achieving the distributed optimization objective under the system-heterogeneous FL comes from the diverse local updates. To address this, we propose a new algorithm, called federated local gradient approximation (FedLGA) which approximates the optimal gradients with a complete local training process from the received heterogeneous remote local learning updates. Specifically, considering the computation complexity, the proposed FedLGA algorithm provides an alternated Hessian estimation method to achieve the approximation, whose extra complexity compared to existing FL approaches is only linear. Additionally, the FedLGA is deployed on the aggregator of FL, so no extra computational cost is required for remote devices. In this article, we also provide the nonconvex optimization analysis of the proposed FedLGA under the formulated system heterogeneity. The results of comprehensive experiments on multiple real-world datasets indicate that FedLGA can effectively address the system-heterogeneous problem and empirically outperform existing methods. For example, we implement system-heterogeneous FL on the CIFAR-10 dataset, compared to FedAvg, FedLGA increases the best testing accuracy from 60.91% to 64.44%. In summary, we highlight the contribution of this article as follows.

1) We formulate the system-heterogeneous FL problem and propose the FedLGA as a promising solution, which tackles the heterogeneity of remote local updates due to the diverse remote computational capacity.
2) For the nonconvex optimization problems, the FedLGA algorithm under the system-heterogeneous FL achieves a convergence rate $O([(1+\rho)/\sqrt{ENT}] + 1/T)$ and $O([(1+\rho)\sqrt{E}/\sqrt{TK}] + 1/T)$ for full and partial participation schemes, respectively.
3) We conduct comprehensive experiments on multiple real-world datasets and the results show that FedLGA outperforms existing FL approaches.

The remainder of this article is organized as follows: The summaries of related works for this article are introduced in Section II. Section III describes the background of FL and the formulation of the system-heterogeneous FL problem. Section IV details the development of our proposed FedLGA algorithm, followed by the theoretical analysis and the convergence rate discussion in Section V. Section VI provides our comprehensive experimental results and analysis for the proposed FedLGA, followed by a conclusion in Section VII.

II. Related Works

FL [1], [2], [3] has been considered as a recently fast-evolving ML topic, where a joint model is learned on a centralized aggregator with the private training data being distributed on remote devices. Typically, the joint model is learned to address distributed optimization problems, for example, word prediction, image classification, and predictive models [19], [20], [21]. The aggregator and the remote devices are considered two key components in an FL framework. Note that both of these two can be denoted as an optimization objective, which focuses on minimizing the corresponding loss functions. The main applications of FL can be summarized into multiple classical ML problems, such as privacy [22], [23], [24], [25], [26], [27], [28], [29], large-scale machine learning, and distributed optimization [30], [31], [32], [33], [34], [35].

Since the first term of FL has been provided in [2], there have been a number of methods in the field. Work in [8] proposes local Stochastic gradient descent (SGD), where each participating remote device in the network performs a single local SGD epoch, and the aggregator averages the received local updates for the joint model. Then, FedAvg in [2] makes modifications to the previous local SGD, which designs the local training process with a large number of epochs. Additionally, [2], [11] have proven that by carefully tuning the number of epochs and learning rate, a good accuracy-communication tradeoff in the FL network can be achieved. However, existing methods still face problems due to the scale of distributed networks, which causes the heterogeneity of statistical training data distribution.

The statistical challenges arise when training the joint model in FL from the non-i.i.d. distributed training dataset, which first causes the problem of modeling the heterogeneity. In literature, there exists a large body of methods against this problem, (e.g., meta-learning [36], asynchronous learning [37], and multitask learning [38]) which has been extended into the FL field, such as [13], [39], [40], [41], [42], [43], and [44]. Additionally, the statistical heterogeneity of FL also causes problems in both the empirical performance and the convergence guarantee, even when learning a single joint model. Indeed, as shown in [2] and [12], the learned joint model from the first proposed FL method is extremely sensitive to the nonidentically distributed training data across remote devices in the network. While parallel SGD and its related variants that are close to FedAvg are also analyzed in the i.i.d. setting [8].

TABLE I
NOTATIONS SUMMARY

N, i	total number, index of the remote device
$f(\cdot)$	joint objective of FL
$F_i(\cdot)$	local objective for remote device i
$\mathcal{X}_i$	private training dataset on remote device i
t	index of global communication round
e	index of local epoch step
$\boldsymbol{w}^t$	joint model after the aggregation of t-th global round
$\boldsymbol{w}_{i,e}^t$	i-th remote model after e local epochs at round t
$\Delta_{i,E}^t$	local update for device i at t-th round as $\boldsymbol{w}_i^t - \boldsymbol{w}_{i,E}^t$
$\hat{\Delta}_{i,E}^t$	approximated local update from the proposed FedLGA

There have been several modifications of FedAvg to address the non-i.i.d. distributed training data in FL. For example, work in [11] uses a decreasing learning rate and provides a convergence guarantee against non-i.i.d. FL. Reddi et al. [45] modified the aggregation rule on the server side. FedProx [12] adds a proximal term on the local loss function to limit the impact from non-i.i.d. data. Additionally, Scaffold [17] and FedDyn [46] augment local updates with extra transmitted variables. Though they suffer from extra communication costs and local computation, the tighter convergence bound can be guaranteed by adding those device-dependent regularizes.

III. BACKGROUND AND PROBLEM FORMULATION

A. Federated Learning Objective

The FL methods [2], [41] are designed to solve optimization problems with a centralized aggregator and a large group of remote devices, which collect and process training samples without sharing raw data. For better presentation, we provide a summary of the most important notations throughout the proposed FedLGA algorithm in Table I. Considering an FL system that consists of N remote devices indexed as $\mathcal{N} = \{1, \ldots, N\}$, the objective $f(\cdot)$ that a learning model aims to minimize could be formalized as

$$\min_{\boldsymbol{w}} f(\boldsymbol{w}) = \frac{1}{N} \sum_{i=1}^{N} F_i(\boldsymbol{w}) \tag{1}$$

where $\boldsymbol{w}$ is the learned joint model parameters, note that in this article, we simplify the dimension of both the input data and the deep neural network model $\boldsymbol{w}$ into vectors for better presentation. And $F_i(\cdot)$ denotes the local objective for the ith device, which typically represents the empirical minimization risk (ERM) $l_i(\cdot;\cdot)$, where the true distribution $\mathcal{D}_i$ is measured by the set of corresponding training data $\mathcal{X}_i$, for example, $F_i(\boldsymbol{w}) = \mathbb{E}_{\mathcal{X}_i \sim \mathcal{D}_i}[l_i(\boldsymbol{w}; \mathcal{X}_i)]$. In this article, we consider the local objective F_i to be nonconvex, which is solved by the corresponding local solver, for example, SGD. During each communication round, remote devices download the current joint model from the aggregator as their local models and perform local solvers toward minimizing the nonconvex objective for E epochs as

$$\boldsymbol{w}_{i,E}^t = \boldsymbol{w}_{i,0}^t - \eta_l \sum_{e=0}^{E-1} \nabla F_i\big(\boldsymbol{w}_{i,e}^t, \mathcal{B}_{i,e}\big) \tag{2}$$

where η_l is the local learning rate, $\nabla F_i(\cdot,\cdot)$ denotes the gradient descent of objective F_i, $\boldsymbol{w}_{i,E}^t$ represents the updated local model, and $\mathcal{B}_{i,e}$ is the eth training batch in SGD, which is typically randomly sampled from $\mathcal{X}_i$ at each epoch. The updated local models are sent back to the aggregator for a new joint model with an aggregation rule. In this article, we consider one round of communication in the network between the aggregator and remote devices as one global round, which is performed T times for the joint model training.

B. Problem Formulation

Due to the consideration of device-heterogeneity in the FL network, recent studies mainly focus on the partial participation scheme, which can avoid waiting for the slowest devices in fully participated scenario [11], [17], [18]. Typically, partially participated FL algorithms establish a threshold $K << N$ at each round, that is, it only selects the first K responded remote devices, all of which complete E local training epochs prior to sending their updated local models to the aggregator.

However, such a partial participation scheme suffers from the known performance-speed dilemma in a system-heterogeneous FL network: a small K can speed up the distributed training but it would also significantly degrade the learning performance as it discards many important training data only stored in those slow devices (i.e., data-heterogeneity [47]), while a large K can utilize more training data but its distributed training process would be greatly slowed down (i.e., data-heterogeneity [47]). Though there have been studies in the literature, the optimization of system-heterogeneous FL lacks formalization. For example, [12] targets this problem by adding a proximal term, which empirically improves the learning performance, and [48] proves that existing FL algorithms will converge to a stationary status with mismatched objective functions under heterogeneous local epochs.

Instead of only waiting for all devices to complete E local epochs, a better solution to address this dilemma is to gather all current local learning models and aggregate them in a manner such that all local training data are utilized to learn the joint model. Specifically, we formalize the training process of FL under system heterogeneity with the following three steps at the tth global round.

1) *Step I:* K remote devices are selected by the aggregator as a subset $\mathcal{K}$, where $|\mathcal{K}| = K$, which receive the current joint model $\boldsymbol{w}^t$ as their local model $\boldsymbol{w}_{i,0}^t = \boldsymbol{w}^t$. The aggregator also delivers an expected epoch number E.
2) *Step II:* Due to the diverse computational capacity, the ith device performs local training for E_i steps, where $1 \leq E_i < E$. Then, the learning results are sent back synchronously within response time constrain.
3) *Step III:* The aggregator updates the joint model $\boldsymbol{w}^{t+1}$ with the received local learning results under a well-designed aggregation rule.

The formulated system-heterogeneous FL shares the same device selection strategy with prior works, such as FedAvg and its variants [11]: the K remote devices are randomly selected in

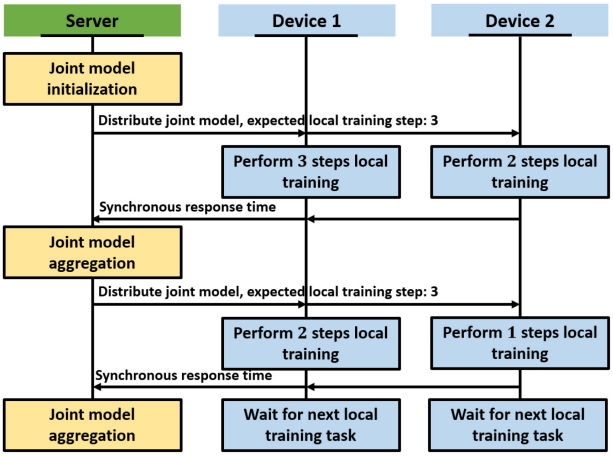

Fig. 1. Heterogeneous local gradients due to system-heterogeneity of FL in FedAvg, illustrated for two remote devices with two rounds of communications.

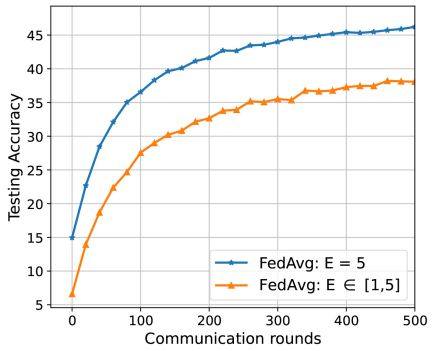

Fig. 2. Toy example in system-heterogeneous FL on CIFAR-100 with 50 remote devices and each device has 20 classes.

each round, which guarantees that each device shares the same probability (K/N) of being selected. Moreover, it allows the synchronous remote updates can be trained with different local epochs, which mitigates the computational capacity difference due to the device-heterogeneity problem. Particularly, we use a virtual subset $\mathcal{K}_1 \in \mathcal{K}$ to represent the remote device i that only performs $E_i < E$ local training epochs, where $|\mathcal{K}_1| = K_1$, and introduce a hyperparameter $\rho = K_1/K$ as the device-heterogeneous ratio. To better present the diverse local updates due to the system heterogeneity, we denote the local update of the ith device at round t after E_i epochs as $\Delta^t_{i,E_i} = \boldsymbol{w}^t_i - \boldsymbol{w}^t_{i,E_i}$, where $\boldsymbol{w}^t_i$ is the initial model before local training (i.e., $\boldsymbol{w}^t_i = \boldsymbol{w}^t_{i,0}$) and the expected update with full E epochs is $\Delta^t_{i,E}$. Hence, under the system-heterogeneity of FL, we aim to minimize the following objective between $\Delta^t_{i,E}$ and Δ^t_{i,E_i} at each communication round t

$$\min \sum_{i \in \mathcal{K}_1} \left\| \Delta^t_{i,E} - \Delta^t_{i,E_i} \right\|. \tag{3}$$

In other words, we want to approximate the expected model update $\Delta^t_{i,E}$ from the received Δ^t_{i,E_i}. This approximation can be performed in the aggregator, which does not introduce any extra computations in remote devices. To achieve this, inspired by prior studies on gradient approximation for improving centralized SGD optimization problems [49], [50], [51], we propose the FedLGA algorithm, which is introduced in detail in the next section.

IV. Proposed Algorithm: FedLGA

A. Design Motivation

For better presentation, we first introduce an example to illustrate the problem of the diverged local gradients in system-heterogeneous FL, as shown in Fig. 1. The introduced FL network consists of two remote devices, where at the start of each communication, the aggregator initializes/aggregates the joint model and sends it to each device. Different from the original FL framework in [2], there is a synchronous response time constraint in Fig. 1 that prevents the device spend more time on local training. We can notice that for the first round, device 2 is only able to perform 2 steps local training. Also, the computation capacity for each communication round can vary, for example, the device 1 does not meet the expected training steps in the round 2. Formally, we denote $\boldsymbol{w}^\star \in \mathbb{R}^n$ be the optimal joint model that leads to the minimum values of the learning objective $f(\boldsymbol{w}^t)$, which can be only ideally obtained when these two devices perform the expected 3 epochs. Due to the uncompleted local learning of device 2 in round 1 and both devices in round 2, the direction of joint model $\boldsymbol{w}^t$ would incrementally deviate from $\boldsymbol{w}^\star$. Moreover, to better illustrate the impact of insufficient local training epochs on the performance of the joint model in system-heterogeneous FL, we provide a toy example as shown in Fig. 2, where a ResNet network [52] is trained with 50 remote devices against the CIFAR-100 dataset under the FedAvg algorithm [11]. It can be easily noticed FedAvg with a sufficient number of five local training epochs significantly outperforms FedAvg where each device runs uniformly random steps between 1 and 5.

For the ith device in the tth round, when the aggregator receives local update Δ^t_{i,E_i}, our proposed FedLGA algorithm applies the following Taylor expansion [53], [54] to approximate the ideal update $\Delta^t_{i,E}$

$$\begin{aligned}\Delta^t_{i,E} = \Delta^t_{i,E_i} + \nabla_{\boldsymbol{g}}\left(\boldsymbol{w}^t_{i,E_i}\right)\left(\boldsymbol{w}^t_{i,E} - \boldsymbol{w}^t_{i,E_i}\right) \\ + O\Big(\left(\boldsymbol{w}^t_{i,E} - \boldsymbol{w}^t_{i,E_i}\right)^2\Big) I_n\end{aligned} \tag{4}$$

where I_n is a n-dimension vector with all elements equal to 1, $\nabla_{\boldsymbol{g}}(\cdot) = \nabla^2 F_i(\cdot)$ is the matrix whose element $g_{j,k} = (\partial F_i^2/[\partial w^t_{i,j}\partial w^t_{i,k}])$ for $j,k \in [n]$, and $(\boldsymbol{w}^t_{i,E} - \boldsymbol{w}^t_{i,E_i})^2$ denotes $(w^t_{i,E,1} - w^t_{i,E_i,1})^{a_1} \cdots (w^t_{i,E,n} - w^t_{i,E_i,n})^{a_n}$ where $w^t_{i,E,j}$ is the jth component of $\boldsymbol{w}^t_{i,E}$ corresponding to I_j and $\sum_{j=1}^n a_j = 2$ as illustrated in [55]. We use $\boldsymbol{g}$ to represent $(\Delta^t_{i,E} - \Delta^t_{i,E_i}/E - E_i)$ which is the averaged gradient $\nabla F_i(\cdot)$ between epoch E_i and E. This also indicates that the joint model performance degradation as shown in the demo example in Fig. 2 is caused by ignoring the higher-order terms $\nabla_{\boldsymbol{g}}(\boldsymbol{w}^t_{i,E_i})(\boldsymbol{w}^t_{i,E} - \boldsymbol{w}^t_{i,E_i}) + O((\boldsymbol{w}^t_{i,E} - \boldsymbol{w}^t_{i,E_i})^2) I_n$. Hence, we can tackle the difference in (3) by approximating the higher-order terms in (4) for each device $i \in \mathcal{K}_1$. To achieve this, a straightforward way is to use the full Taylor expansion for gradient compensation.

B. Hessian Approximation

However, computing the full Taylor expansion can be practically unrealistic because of two fundamental challenges: 1) for the devices $i \in \mathcal{K}_1$, the $\boldsymbol{w}_{i,E}^t$ is still not known to the aggregator and 2) the approximation of higher-order terms in the Taylor expansion requires a sum of an infinite number of items, where even solving the first-order approximation $\nabla_{\boldsymbol{g}}(\boldsymbol{w}_{i,E_i}^t)(\boldsymbol{w}_{i,E}^t - \boldsymbol{w}_{i,E_i}^t)$ is also highly nontrivial. To address the first challenge, we make a first-order approximation of $\boldsymbol{w}_{i,E}^t$ from those devices with full local epochs, which is denoted by $\hat{\boldsymbol{w}}_{i,E}^t$. Inspired by prior works on asynchronous FL (AFL) weight approximation [56], we obtain the first-order approximation of $\hat{\boldsymbol{w}}_{i,E}^t = \boldsymbol{w}^t + (1/K_2)\sum_{i\in\mathcal{K}_2}\Delta_{i,E}^t$, where $\mathcal{K}_2 = \mathcal{K} - \mathcal{K}_1$ is the set of devices with full local epochs. As such, we show the first-order item approximation as

$$\hat{\Delta}_{i,E}^t \approx \Delta_{i,E_i}^t + \nabla_{\boldsymbol{g}}\left(\boldsymbol{w}_{i,E_i}^t\right)\left(\hat{\boldsymbol{w}}_{i,E}^t - \boldsymbol{w}_{i,E_i}^t\right) \tag{5}$$

where $\hat{\Delta}_{i,E}^t$ denotes the approximated heterogeneous local updates that $i \in \mathcal{K}_1$, for distinguishing the approximation from the ideal updates $\Delta_{i,E}^t, i \in \mathcal{K}_2$. Note that the second challenge comes from the derivative term $\nabla_{\boldsymbol{g}}(\boldsymbol{w}_{i,E_i}^t)$, which corresponds to the Hessian matrix of the local objective function $F_i(\cdot)$ that is, defined as $\mathbf{H} = [h_i^{j,k}], j,k = 1,\ldots,n$, where $[h_i^{j,k}] = ([\partial F_i(\cdot)^2]/[\partial w_{i,j}^t \partial w_{i,k}^t])$. Since the computation cost of obtaining the Hessian matrix of a deep learning model is still expensive, our FedLGA algorithm applies the outer product matrix of $\nabla_{\boldsymbol{g}}(\boldsymbol{w}_{i,E_i}^t)$, which is denoted as $G(\boldsymbol{w}_{i,E_i}^t)$ that follows:

$$G\left(\boldsymbol{w}_{i,E_i}^t\right) = \left(\frac{\partial F_i\left(\boldsymbol{w}_{i,E_i}^t\right)}{\partial \boldsymbol{w}_{i,E_i}^t}\right)\left(\frac{\partial F_i\left(\boldsymbol{w}_{i,E_i}^t\right)}{\partial \boldsymbol{w}_{i,E_i}^t}\right)^{\top}. \tag{6}$$

This outer product of the remote gradient has been proved as an asymptotic estimation of the Hessian matrix using the Fisher information matrix [57], which has a linear extra complexity compared to the computation of Δ_{i,E_i}^t [58]. Note that this equivalent approach for solving the approximation of the Hessian matrix has been also applied in [59] and [60].

C. Algorithm of FedLGA

In order to quantitate the difference between E and E_i for device i, we introduce a new parameter $\tau_i = E - E_i + 1$, where the devices with full local learning epochs satisfy $\tau_i = 1$. Additionally, to decouple the local learning and the aggregation, we introduce a joint model learning rate η_g and the aggregation rule for $\boldsymbol{w}^{t+1}$ in our FedLGA is given by

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t + \eta_g \frac{1}{K}\left(\sum_{i\in\mathcal{K}_1}\hat{\Delta}_{i,E}^t + \sum_{i\in\mathcal{K}_2}\Delta_{i,E}^t\right). \tag{7}$$

It can be noticed that the most representative FL algorithm FedAVG [2], [11] could be considered as a special case of the proposed FedLGA, where the network has no system-heterogeneity with $\eta_g = 1$ and $\tau_i = 1$ for all devices. We briefly summarize the learning process of the proposed FedLGA algorithm. Specifically, Algorithm 1 introduces the local training process on remote device i at the tth round. With a given joint model $\boldsymbol{w}^t$ and an expected local epoch number E from the server, device i performs E_i epochs of training within the constraint of synchronous responding time. Then, the gradient update Δ_{i,E_i}^t and the delay parameter τ_i are sent back to the server, as illustrated in lines 4–7.

Algorithm 1 FedLGA: Local Learning on Device i

1: **Input:** Joint model $\boldsymbol{w}^t$ for t-th round, expected epoch E, local learning rate η_l, constrained response time.
2: **Return:** Remote update $(\Delta_{i,E_i}^t, \tau_i)$
3: Initialize training model $\boldsymbol{w}_{i,0}^t = \boldsymbol{w}^t$
4: $\boldsymbol{w}_{i,E_i}^t = \boldsymbol{w}_{i,0}^t - \eta_l \sum_{e=0}^{E_i-1} \nabla F_i(\boldsymbol{w}_{i,e}^t, \mathcal{B}_{i,e})$
5: $\Delta_{i,E_i}^t = \boldsymbol{w}_{i,E_i}^t - \boldsymbol{w}_{i,0}^t$
6: Calculate $\tau_i = E - E_i + 1$
7: Communicate $(\Delta_{i,E_i}^t, \tau_i)$ to the server

And Algorithm 2 presents the joint model training at the t-th communication round on the aggregator, where the developed local gradient approximation method is applied. Lines 4–7 illustrate that at the t-th round, the partial participate subset $\mathcal{K}$ is randomly chosen from $\mathcal{N}$, where each device performs local training with the given $(\boldsymbol{w}^t, E)$ as demonstrated in Algorithm 1. Then, as shown in line 8, $\hat{\boldsymbol{w}}_{i,E}^t$ is computed with the updates from $\mathcal{K}_2$ where devices can perform E expected epochs. And for remote devices which lack computation capacity in $\mathcal{K}_1$, the gradient approximation for $\hat{\Delta}_{i,E}^t$ with the received Δ_{i,E_i}^t and the computed $\hat{\boldsymbol{w}}_{i,E}^t$ is introduced in lines 9–14. Finally, the joint model is updated through line 15.

Note that unlike FedProx [12] which uses an added proximal term on its local learning objective, which increases the local training computation cost, our proposed FedLGA does not require any extra computation and communication cost on remote devices. Instead, the local gradient approximation method against device heterogeneity is developed on the aggregator side, which is usually considered to have powerful computational resources in FL network settings. And the computation cost of our FedLGA mainly comes from the calculation of (5), and its complexity has been proved to be linear to the dimension of $\boldsymbol{w}^t$ [37], [58].

There have been other approaches against the FL device heterogeneity problem, where one important category is AFL. Typically, AFL approaches maximized the computation resource utilization on heterogeneous devices in the FL network to improve the model learning efficiency [61]. Some AFL works [62], [63] mitigate the device heterogeneity by adaptively selecting remote devices with more robustness and powerful computation. However, this leads to the absence of training information on the other devices, which bounds the joint model performance against data heterogeneity. Other AFL works [37], [64] group remotes devices into different clusters based on the computation capacity before asynchronous training, however, utilizing the extra strategy usually results in a decline in computation efficiency instead. Compared to these AFL approaches, the proposed FedLGA considers both the device- and data-heterogeneity problems

Algorithm 2 FedLGA: Server Side at Round t

Input: Initialized model $\boldsymbol{w}^0$, communication round upper bound T, expected local epoch E, joint learning rate η_g.
Output: Trained joint model $\boldsymbol{w}^T$.
for Each communication round $t = 0$ **to** T **do**
  Select partial participate devices $\mathcal{K}$ from $\mathcal{N}$
  Communicate $(\boldsymbol{w}^t, E)$ to each device $i \in \mathcal{K}$
  For each device, perform local training as Algorithm 1
  Receive $(\Delta^t_{i,E_i}, \tau_i)$ from device i as Algorithm 1
  Compute $\hat{\boldsymbol{w}}^t_{i,E} = \boldsymbol{w}^t + \frac{1}{K_2}\sum_{i\in\mathcal{K}_2}\Delta^t_{i,E}$
  for each device $i \in \mathcal{K}_1$ **do**
    if $\tau_i > 1$ **then**
      Approx $G(\boldsymbol{w}^t_{i,E_i})$ from Eq. (6)
      $\hat{\Delta}^t_{i,E} = \Delta^t_{i,E_i} + G(\boldsymbol{w}^t_{i,E_i})(\hat{\boldsymbol{w}}^t_{i,E} - \boldsymbol{w}^t_{i,E_i})$
    end if
  end for
  $\boldsymbol{w}^{t+1} = \boldsymbol{w}^t + \frac{\eta_g}{K}(\sum_{i\in\mathcal{K}_1}\hat{\Delta}^t_{i,E} + \sum_{i\in\mathcal{K}_2}\Delta^t_{i,E})$
end for

in the formulated system-heterogeneous framework with only linear extra computation cost, which shows its superiority.

V. Convergence Analysis

In this section, we provide the convergence analysis of the proposed FedLGA algorithm under smooth, nonconvex settings against the introduced system-heterogeneous FL network. Note that to illustrate the analysis process, we analyze both the full and partial device participation schemes with the following assumptions, theorems, corollaries, and remarks.

Assumption 1 (L-Lipschitz Gradient): For all remote devices $i \in \mathcal{N}$, there exists a constant $L > 0$, such that

$$||\nabla F_i(\boldsymbol{v}) - \nabla f(\boldsymbol{u})|| \le L||\boldsymbol{v} - \boldsymbol{u}|| \ \forall \boldsymbol{u}, \boldsymbol{v}. \tag{8}$$

Assumption 2 (Unbiased Local Stochastic Gradient Estimator): Let $\mathcal{B}^t_{i,e}$ be the random sampled local training batch in the t-th round on device i at local step e, the local training stochastic gradient estimator is unbiased that

$$\mathbb{E}\left[\nabla F_i(\boldsymbol{w}^t_i, \mathcal{B}^t_{i,e})\right] = \nabla F_i(\boldsymbol{w}^t_i) \forall i \in \mathcal{N}. \tag{9}$$

Assumption 3 (Bounded Local and Global Variance): For each remote device i, there existing a constant value σ_l that the variance of each local gradient satisfies

$$\mathbb{E}\left[\left\|\nabla F_i(\boldsymbol{w}^t_i, \mathcal{B}^t_{i,e}) - \nabla F_i(\boldsymbol{w}^t_i)\right\|^2\right] \le \sigma_l^2 \tag{10}$$

and the global variability of the ith gradient to the gradient of the joint objective is also bounded by another constant σ_g, which satisfies

$$\left\|\nabla F_i(\boldsymbol{w}^t_i) - \nabla f(\boldsymbol{w}^t)\right\|^2 \le \sigma_g^2 \ \forall i \in \mathcal{N}. \tag{11}$$

Note that the first two assumptions are standard in studies on nonconvex optimization [65], [66]. And for Assumption 3, besides the widely applied local gradient bounded variance in FL, we use the global bound σ_g to quantify the data-heterogeneity due to the non-i.i.d. distributed training dataset, as illustrated in recent FL studies [18], [45]. Note that in this article, we do not assume a bounded gradient which is often introduced in FL optimization analysis that leads to a loose convergence bound. Additionally, to illustrate the device-heterogeneity under the formulated system-heterogeneous FL in this article, we make an extra assumption on the boundary of the approximated gradients from the proposed FedLGA algorithm as the following.

Assumption 4 (Bounded Taylor Approximation Remainder): For the quadratic term remainder of Taylor expansion $\nabla^2_{\boldsymbol{g}}(\boldsymbol{w}^t_i)$, there exists a constant M for an arbitrary device i that satisfies

$$\left\|\nabla^2_{\boldsymbol{g}}(\boldsymbol{w}^t_i)\right\| \le M. \tag{12}$$

Note that Assumption 4 states an upper bound for the second term in the Taylor expansion approximation, which can be considered as the worst-case scenario for the difference between the approximated local gradient in FedLGA to its optimal gradient value. Additionally, for better presentation, we consider an upper bound $\tau_{\max}$ for the heterogeneous local gradients in the rest of our analysis that $\tau_i \le \tau_{\max} \ \forall i \in \mathcal{N}$. It is worth noting that this assumption does not affect the convergence rate of FedLGA in the following theorems and corollaries, instead of simplifying the mathematical deviations.

A. Convergence Analysis for Full Participation

We first provide the convergence analysis of the proposed FedLGA algorithm under the full device participation scheme, where we have the following results.

Theorem 1: Let Assumptions 1–4 hold. The local and global learning rates η_l and η_g are chosen such that $\eta_l < (1/[\sqrt{30(1+\rho)}LE])$ and $\eta_g\eta_l \le (1/[(1+\rho)LE])$. Under a full device participation scheme, the iterates of FedLGA satisfy

$$\min_{t\in T}\mathbb{E}\left\|\nabla f(\boldsymbol{w}^t)\right\|^2 \le \frac{f^0 - f^\star}{c_1\eta_g\eta_l ET} + \Phi_1 \tag{13}$$

where $f^0 = f(\boldsymbol{w}^0), f^\star = f(\boldsymbol{w}^\star)$, c_1 is constant, the expectation is over the remote training dataset among all devices, and $\Phi_1 = (1/c_1)[((1+\rho)\eta_g\eta_l\sigma_l^2/2N) + (5/2)\eta_l^2EL^2(\sigma_l^2 + 6E\sigma_g^2) + c_2\mathbb{E}||\nabla F_i(\boldsymbol{w}^T)||^4]$, $((1/2) - 15(1+\rho)E^2\eta_l^2L^2) > c_1 > 0$, and $c_2 = ([\eta_g\eta_l^2\rho M^2\tau^2_{\max}]/N\eta_g\eta_l)(\eta_gL + \eta_l^3\tau^2_{\max})$.

Proof: See in online Appendix A, available in [67]. ■

Corollary 1: Suppose the learning rates η_l and η_g are such that the condition in Theorem 1 are satisfied. Let $\eta_l = (1/\sqrt{TEL})$ and $\eta_g = \sqrt{EN}$. The convergence rate of the proposed FedLGA under the full device participation scheme satisfies

$$\min_{t\in T}\mathbb{E}\left\|\nabla f(\boldsymbol{w}^t)\right\|^2 = O\left(\frac{(1+\rho)}{\sqrt{ENT}} + \frac{1}{T}\right). \tag{14}$$

Remark 1: From the results in Theorem 1, the convergence bound of full device participation FedLGA contains two parts: 1) a vanishing term $(f^0 - f^\star/c_1\eta_g\eta_l ET)$ corresponding to the increase of T and 2) a constant term Φ_1, which is independent of T. We can notice that as the value of c_1 is related to ρ, the vanishing term which dominates the convergence of the FedLGA algorithm is impacted by the device-heterogeneity.

TABLE II
CONVERGENCE RATES FOR FL OPTIMIZATION APPROACHES

Method	Dataset	Convexity[1]	Partial Worker[2]	Device-Heterogeneous[3]	Other Assumptions[4]	Convergence Rate
Stich et al. [8]	i.i.d.	SC	✗	✗	BCGV; BOGV	$O(\frac{NE}{T}) + O(\frac{1}{\sqrt{NET}})$
Khaled et al. [68]	non-i.i.d.	C	✗	✗	BOGV; LBG	$O(\frac{1}{T}) + O(\frac{1}{\sqrt{NT}})$
Li et al. [11]	non-i.i.d.	SC	✓	✗	BOBD; BLGV; BLGN	$O(\frac{E}{T})$
FedProx [12]	non-i.i.d.	NC	✓	✓	BGV; Prox	$O(\frac{1}{\sqrt{T}})$
Scaffold [17]	non-i.i.d.	NC	✓	✗	BLGV; VR	$O(\frac{1}{T}) + O(\frac{1}{\sqrt{NET}})$
Yang et al. [18]	non-i.i.d.	NC	✓	✗	BLGV	$O(\frac{1}{T}) + O(\frac{1}{\sqrt{NET}})$
FedSAM [69]	non-i.i.d.	NC	✓	✗	BLGV	$O(\frac{1}{T}) + O(\frac{1}{\sqrt{NET}})$
FedBuff [70]	non-i.i.d.	NC	✗	✓	BCGV; BOGV	$O(\frac{1}{TE}) + O(\frac{1}{\sqrt{NET}})$
FedLGA	**non-i.i.d**	**NC**	✓	✓	**BLGV**	$O(\frac{1}{\mathbf{TE^2}}) + O(\frac{(1+\rho)\sqrt{\mathbf{E}}}{\sqrt{\mathbf{TK}}})$

[1] Shorthand notations for the convexity of the introduced methods: SC: Strongly Convex, C: Convex and NC: Non-Convex.
[2] Shorthand summaries for whether the compared method satisfies the partial participation scheme: ✓: satisfy and ✗: not satisfy.
[3] Shorthand summaries for whether the device-heterogeneity of FL is considered: ✓: yes and ✗: no.
[4] Shorthand notation for other assumptions and variants. BCGV: the remote gradients are bounded as $\mathbb{E}[||\nabla F_i(\boldsymbol{w}_i^t, \mathcal{B}_{i,e}^t) - \nabla f(\boldsymbol{w}_i^t)||^2] \leq \sigma^2$. BOGV: the variance of optimal gradient is bounded as $\mathbb{E}[||\nabla f(\boldsymbol{w}^\star||^2] \leq \sigma^2$. BOBD: the difference of optimal objective is bounded as $f(\boldsymbol{w}^\star) - \mathbb{E}[F_i(\boldsymbol{w}^\star)] \leq \sigma^2$. BGV: the dissimilarity of remote gradients are bounded $\mathbb{E}[||\nabla F_i(\boldsymbol{w}_i^t)||^2]/||\nabla f(\boldsymbol{w}^t)||^2 \leq \sigma^2$. BLGV: the variance of stochastic gradients on each remote device is bounded (same as our Assumption 3). BLGN: the norm of an arbitrary remote update is bounded. LBG: each remote devices use the full batch of local training data for update computing. Prox: the remote objective considers proximal gradient steps. VR: followed by trackable states, there is variance reduction.
Note that for better presentation, we use a unified σ symbol, which can vary depending on the detailed method.

Additionally, we find an interesting boundary phenomenon on the vanishing term in Theorem 1 that, when the FL network satisfies $\rho = 0$, the decay rate of the vanishing term matches the prior studies of FedAVG with two-sided learning rates [18].

Remark 2: For the constant term Φ_1 in Theorem 1, we consider the first part $([(1+\rho)\eta_g\eta_l\sigma_l^2]/2N)$ is from the local gradient variance of remote devices, which is linear to ρ. And the second part $(5/2)\eta_l^2EL^2(\sigma_l^2 + 6E\sigma_g^2)$ denotes the cumulative variance of E local training epochs, which is also influenced by the data-heterogeneity σ_g. Inspired by [18], we consider an inverse relationship between η_l and E, for example, $\eta_l \propto O(1/K)$. For the third term, we can notice that it is quadratically amplified by the variance of optimal gradient as $\mathbb{E}||\nabla F_i(\boldsymbol{w}^\star)||^2$. Note that different from other FL optimization analysis that assumes a bounded optimal gradient [11], [12], the proposed FedLGA does not require such an assumption. Hence, in order to address the high power third term of $\mathbb{E}||\nabla F_i(\boldsymbol{w}^\star)||^2$, we apply a weighted decay γ factor to local learning rate as $\eta_l^{t+1} = (1-\gamma)\eta_l^t$. Additionally, as suggested in [56], the third term indicates the staleness, which could be controlled via an inverse function such as $\tau_i(t) \propto O(1/t+1)$.

B. Convergence Analysis for Partial Participation

We then analyze the convergence of FedLGA under the partial device participation scheme, which follows the sampling strategy I in [11], where the subset $\mathcal{K} \in \mathcal{N}$ is randomly and independently sampled by the aggregator with replacement.

Theorem 2: Let Assumptions 1–4 hold. Under the partial device participation scheme, the iterates of FedLGA with local and global learning rates η_l and η_g satisfy

$$\min_{t\in T} \mathbb{E}\left\|\nabla f\left(\boldsymbol{w}^t\right)\right\|^2 \leq \frac{f^0 - f^\star}{d_1\eta_g\eta_l ET} + \Phi_2 \tag{15}$$

where $f^0 = f(\boldsymbol{w}^0), f^\star = f(\boldsymbol{w}^\star)$, d_1 is constant, and the expectation is over the remote training dataset among all devices. Let η_l and η_g be defined such that $\eta_l \leq (1/[\sqrt{30(1+\rho)}LE])$, $\eta_g\eta_l E \leq (K/[(K-1)(1+\rho)L])$ and $([30(1+\rho)K^2E^2\eta_l^2L^2]/N^2) + ([L\eta_g\eta_l(1+\rho)]/K)(90E^3L^2\eta_l^2 + 3E) < 1$. Then, we have $\Phi_2 = (1/d_1)[d_2(\sigma_l^2 + 3E\sigma_g^2) + d_3(\sigma_l^2 + 6E\sigma_g^2) + d_4\mathbb{E}||\nabla F_i(\boldsymbol{w}_i^t)||^4]$, where $d_2 = ([(1+\rho)\eta_g\eta_l L]/2K)$, $d_3 = ((5K^2/2N^2) + (15EL\eta_l\eta_g/2K))((1+\rho)\eta_l^2EL^2)$ and $d_4 = \eta_l\rho\tau_{\max}^2M^2((L\eta_g/K^2) + (\eta_l^3K\tau_{\max}^2/N^2))$.

Proof: See in online Appendix B, available in [67]. ■

We restate the results in Theorem 2 for a specific choice of η_l and η_g to clarify the convergence rate as follows.

Corollary 2: Suppose the learning rates η_l and η_g are such that the condition in Theorem 2 are satisfied. Let $\eta_l = (1/\sqrt{TEL})$ and $\eta_g = \sqrt{EK}$. The convergence rate of the proposed FedLGA under the partial device participation scheme satisfies

$$\min_{t\in T} \mathbb{E}\left\|\nabla f\left(\boldsymbol{w}^t\right)\right\|^2 = O\left(\frac{(1+\rho)\sqrt{E}}{\sqrt{TK}} + \frac{1}{T}\right). \tag{16}$$

Remark 3: Compared to the convergence rate of full participated FedLGA, the partial scheme has a larger variance term, which indicates that the uniformly random sampling strategy does not incur a significant convergence difference.

Remark 4: We compare the convergence rate of the proposed FedLGA and related FL optimization approaches in Table II. We can notice that compared to the previous works in [8] and [68] which focus on only convex or strongly convex optimization problems, the proposed FedLGA can address the nonconvex problem. And comparing to [11], the FedLGA algorithm achieves a better convergence rate with fewer assumptions, especially the bounded gradient assumption.

TABLE III
DATASET INFORMATION OVERVIEW

Dataset	Dataset Size	Classes	P[1]	Image Feature
FMNIST [73]	60,000	10	2	28×28
CIFAR-10 [74]	60,000	10	2	$32 \times 32 \times 3$
CIFAR-100 [74]	60,000	100	20	$32 \times 32 \times 3$

[1] Shorthand notation for the number of classes in one remote device.

Remark 5: As shown in Table II, we also find that the dominating term of the obtained convergence rate for both the full and partial schemes is linear to the system-heterogeneity, that is, $(1 + \rho)$. When $\rho = 0$, the convergence rate matches the results in [17] and [18], and when ρ reaches 1, the proposed FedLGA still gets the same order. Specifically, we can notice that compared to Scaffold [17], works in [18] and our proposed FedLGA does not require the assumption of variance reduction.

Remark 6: We can also notice that the only method which addresses both nonconvex optimization and device-heterogeneity under the partial participation FL scheme is FedProx [12], with convergence rate $O(1/\sqrt{T})$ [47]. From Corollary 2, the proposed FedLGA algorithm can achieve $O(\sqrt{E}/\sqrt{TK})$. Compared to FedProx, if the number of sampled devices and the number of local epoch steps satisfy that $K > E$, it is obvious that our FedLGA achieves a speedup of convergence rate against FedProx. Moreover, the analysis of FedLGA does not require the assumptions of either the proximal local training step or the bounded gradient dissimilarity.

VI. EXPERIMENTS

We conducted comprehensive experiments under the formulated system-heterogeneous FL framework on multiple real-world datasets to evaluate the proposed FedLGA algorithm. The experiments are performed with one GeForce GTX 1080Ti GPU card on Pytorch [71] and we follow the settings in [72] to implement the FL baselines (e.g., FedAVG). In the rest of this section, Section VI-A details the experiment setup, followed by the performance analysis of FedLGA on multiple evaluation metrics in Section VI-B. Then, Section VI-C provides the ablation study for the hyperparameters of FedLGA.

A. Experimental Setup

1) Datasets and Models: Three popular real-world datasets are considered in this article: 1) FMNIST [73] (Fashion MMNIST); 2) CIFAR-10; and 3) CIFAR-100 [74]. Considering an FL network with $N = 50$ remote devices, we introduce the general information of each dataset as shown in Table III. Note that for the $32 \times 32 \times 3$ color images in CIFAR-10 and CIFAR-100 datasets, we make the following data preprocessing to improve the FL training performance: each image sample is normalized, cropped to size 32, horizontally flipped with the probability of 50% and resized to 224×224.

Then, we follow the previous settings in [72] and [2] to present the data heterogeneity of FL. In this article, we consider the following nonoverlapped non-i.i.d. training data partition scenario, where the ith remote private dataset $\mathcal{X}_i$ and the total training dataset $\mathcal{X}$ satisfy: $|\mathcal{X}| = \sum_i |\mathcal{X}_i|$. Then, for each remote training dataset $\mathcal{X}_i$, we consider it contains P classes of samples. Note that for FMNIST and CIFAR-10, we set $P = 2$ and for CIFAR-100, we set $P = 20$ by default. To solve the classification problems from the introduced datasets, we run two different neuron network models. For FMNIST, we run a two-layer fully connect MLP network with 400 hidden nodes. For CIFAR-10 and CIFAR-100, we run a ResNet network, which follows the settings in [52].

2) Compared Methods: We compared FedLGA with the following eight representative FL methods.

1) *FedAvg [2]* is considered one of the groundbreaking works in the FL research field. We set up the FedAvg approach based on the settings in [11], which first provides a convergence guarantee against data-heterogeneous FL. Note that in our simulation, we follow scheme I in [11] for partial participation.
2) *FedProx [12]* is one popular variant of FedAvg which adds a quadratic proximal term to limit the impact from local updates in the device-heterogeneous FL. In this article, we follow the instructions provided in [12] to set the proximal hyperparameter, which controls the local objective dissimilarity.
3) *FedNova [48]* improves FedAvg from the aggregator side. It assumes a diverse local update scenario where each remote device may perform a different number of local epochs. To achieve this, FedNova normalizes and scales the local updates, which is also considered a modification to FedAvg.
4) *Scaffold [17]* models the data-heterogeneous FL problem as the global variance among each remote device in the network. Scaffold addresses this problem by controlling the variates between the aggregator and the devices to estimate the joint model update direction, which is achieved via applying the variance reduction technique [75], [76].
5) *FedDyn [46]* adds a regularization term on FedAvg on the remote device side at each local training epoch, which is developed based on the joint model and the local training model at the previous global round.
6) *FedCM [77]* tackles the data-heterogeneity issue under the partial participation FL. Specifically, FedCM aggregates the joint model information from the previous communication round and modifies the remote gradient update with a momentum-like term, which can effectively correct the bias of local SGD.
7) *MimeLite [78]* provides a general algorithmic framework, called MIME to improve the FL optimization challenge because of the data-heterogeneity, which: 1) mitigates the remote device drift and 2) adapts arbitrary centralized optimization algorithm, for example, momentum.

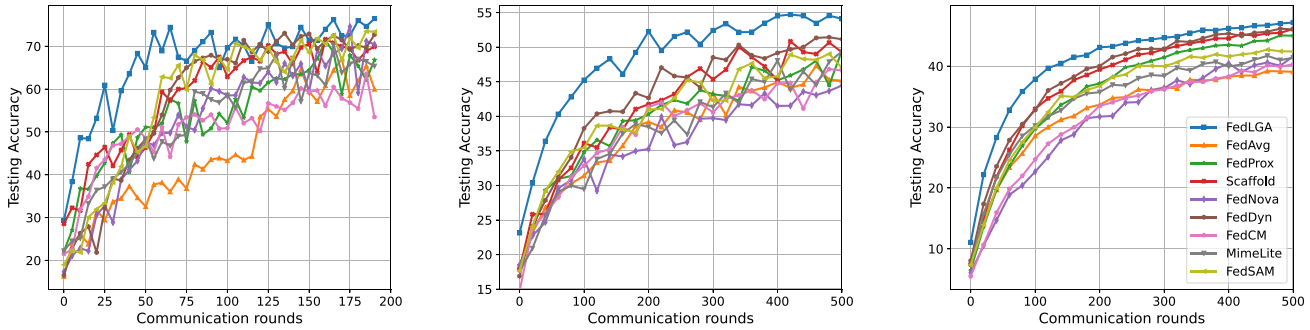

Fig. 3. Learning performance of testing accuracy under the system-heterogeneous FL with $\rho = 0.5$, $\tau_{\max} = E - 1$ and $E = 5$. (a) FMNIST. (b) CIFAR-10. (c) CIFAR-100.

TABLE IV
COMPARED METHODS HYPERPARAMETER GRID SEARCH

Methods	Hyper-parameters
FedProx	proximal parameter $\mu = \{0.1, 0.5, 1\}$
FedNova	proximal SGD: $\mu = \{0.001, 0.005, 0.01\}$
Scaffold	local variance control $c = \{5, 10, 20\}$
FedDyn	dynamic regulator $\alpha = \{0.1, 0.01, 0.001\}$
FedCM	momentum $\beta = \{0.01, 0.1, 0.5\}$
MimeLite	momentum $\beta = \{0.01, 0.1, 0.5\}$
FedSAM	sharpness radius $r = \{0.1, 0.5, 0.9\}$

8) *FedSAM [69]* addresses the data-heterogeneity problem by focusing on the generalization ability of the remote learning process. Specifically, a sharpness aware minimization (SAM) [79] local optimizer is developed which can theoretically improve the generalization bound of FL global aggregation.

3) Implementation: In this work, we simulated an FL network with the formulated system-heterogeneous problem. For the compared methods, we set up the hyperparameters via a grid search, which is introduced in Table IV. Note that for the FedSAM method, as it takes doubled gradient back-propagation operations during each remote training epoch, the performance is evaluated based on the operation number for comparison. The system-heterogeneous FL network is with the following settings by default.

1) The total number of remote devices $N = 50$.
2) For each global communication round, the number of devices being chosen by the aggregator is $K = 10$.
3) For the local training process, we set $E = 5$ and $|\mathcal{B}| = 10$.
4) To illustrate the device-heterogeneity, we set $\rho = 0.5$ and $\tau_{\max} = E - 1$, where τ_i for the ith device is uniformly distributed within $[1, \tau_{\max}]$. For example, when $\rho = 0.5$, $\tau_{\max}$, 50% of chosen remote devices in $\mathcal{K}$ can only perform at most $\tau_{\max}$ local epochs, instead of a full E epoch remote training.

4) Evaluation Metrics: To evaluate the experimental results accurately, we introduce the following two categories of evaluation metrics, each of which is investigated in multiple ways. Note that in our analysis, we define a target testing accuracy for each dataset as FMNIST 65%, CIFAR-10 55%, and CIFAR-100 40%.

1) *Model Performance:* To evaluate the joint model in FL, we investigate the following features: the learning curves of training loss and testing accuracy, and the best-achieved accuracy among all rounds.
2) *Communication Cost:* In this article, we perform the FL network on one GPU card with the Python threading library. Hence, we present the communication cost of FL by evaluating the following features: the number of rounds, the average single-round running time, and the total running time to achieve the targeted testing accuracy.

B. Performance Analysis

1) Comparison to Existing Approaches: Figs. 3 and 4 show the learning curves of the testing accuracy and the training loss for the compared FL approaches over three datasets, respectively. We can notice that compared to the existing FL methods, the proposed FedLGA algorithm achieves the best overall performance on the lowest training loss, highest testing accuracy, and the fastest convergence speed. For example, as shown in Fig. 3(c), the proposed FedLGA reaches the targeted 40% testing accuracy with only 145 rounds, which is 1.9×, 1.5×, 1.8×, 1.3×, 1.1×, 1.9×, 1.9×, and 1.5× faster than FedAvg, FedProx, FedNova, Scaffold, FedDyn, FedCM, MimeLite, and FedSAM, respectively. Specifically, as shown in Fig. 4(b), though the proposed FedLGA only reaches the second-lowest training loss on the CIFAR-10 dataset, it outperforms other methods with an obviously faster convergence speed. We can also notice that compared to other benchmarks, FedDyn achieves the second-best performance on average.

We then analyze the performance of the best approached testing accuracy for the compared methods, where the results are shown in Fig. 5. It can be noticed that the proposed FedLGA algorithm outperforms other compared methods and achieves the best testing accuracy on each dataset. For example, as shown in Fig. 5(b), FedLGA improves the best-obtained testing accuracy on CIFAR-10 (i.e., 64.44%) by 5.7%, 3.8%, 5.5%, 0.7%, 0.4%, 2.8%, 2.7%, and 1.0%

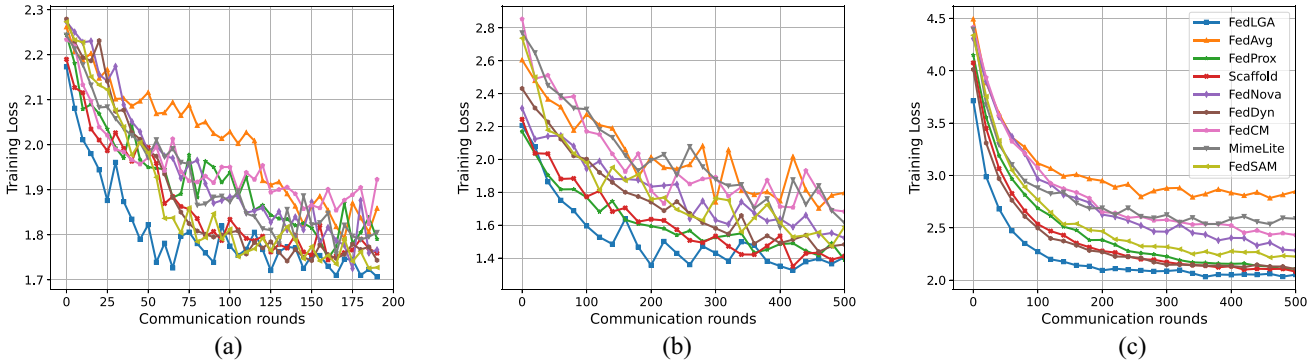

Fig. 4. Learning performance of training loss under the system-heterogeneous FL with $\rho = 0.5$, $\tau_{\max} = E - 1$ and $E = 5$. (a) FMNIST. (b) CIFAR-10. (c) CIFAR-100.

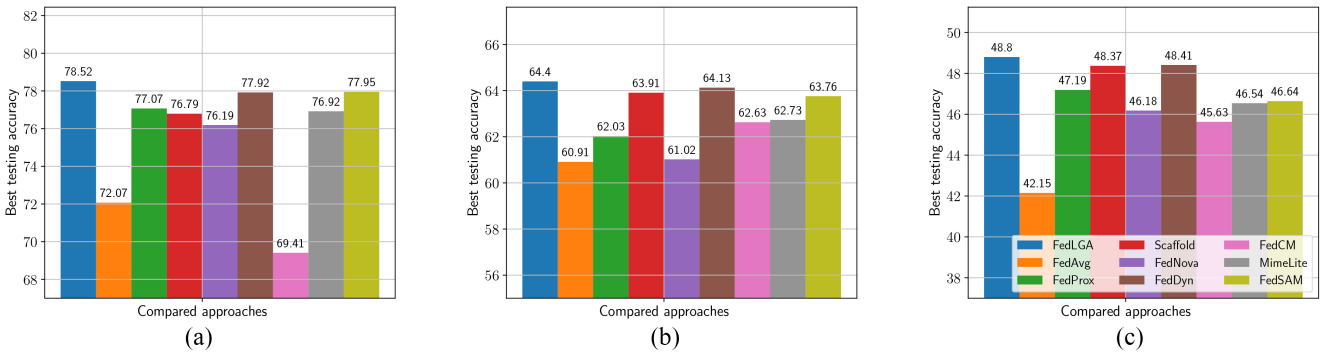

Fig. 5. Learning performance of best accuracy under the system-heterogeneous FL with $\rho = 0.5$, $\tau_{\max} = E - 1$ and $E = 5$. (a) FMNIST. (b) CIFAR-10. (c) CIFAR-100.

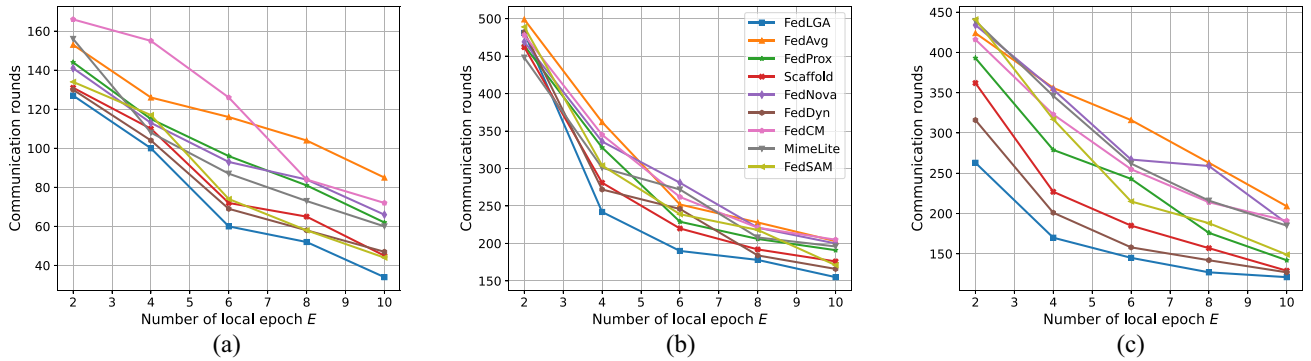

Fig. 6. Communication rounds of the compared FL methods to achieve the targeted testing accuracy under a different number of remote training epoch E with $\rho = 0.5$ and $\tau_{\max} = E - 1$. (a) FMNIST. (b) CIFAR-10. (c) CIFAR-100.

comparing to FedAvg, FedProx, FedNova, Scaffold, FedDyn, FedCM, MimeLite, and FedSAM, respectively.

2) Analysis of System-Heterogeneous FL: To further investigate the learned joint model performance of the compared methods, we construct different system-heterogeneous FL network scenarios. First, we study the impact of different local training epochs E, where the results are shown in Fig. 6. Note that for better comparison, we denote the performance via the number of global communication rounds to the targeted testing accuracy. It can be easily noticed from the results that as the value of E becomes larger, the number of global communication round to the target accuracy is less for each compared method. In this condition, the proposed FedLGA algorithm still outperforms other methods with the lowest number of rounds on each value of E. For example, when $E = 8$ against the FMNIST dataset, the proposed FedLGA reaches the target accuracy with only 52 rounds, whereas FedAvg requires 2 times more rounds for 104.

Then, we study the performance of the compared approaches in an FL network with different device-heterogeneity ratios, which is shown in Fig. 7. We can notice from the results that as ρ becomes larger, the number of communication rounds to achieve the target testing accuracy for all compared methods also increases. Especially, for FMNIST and CIFAR-10 datasets, when $\rho = 0.1$, all the compared FL methods in this article have similar performance. We

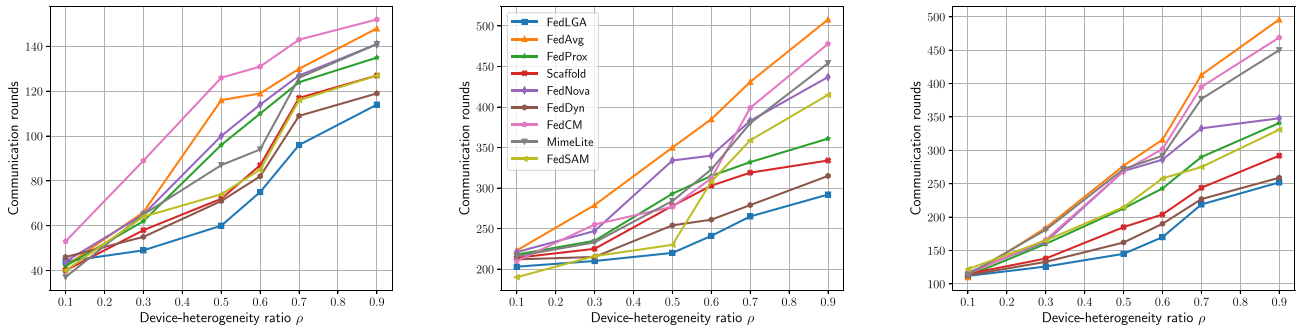

Fig. 7. Communication rounds of the compared FL methods to achieve the targeted testing accuracy under different device-heterogeneity ratio ρ with $E = 5$ and $\tau_{\max} = E - 1$. (a) FMNIST. (b) CIFAR-10. (c) CIFAR-100.

TABLE V
RUNNING TIME (SECONDS) TO TARGET TESTING ACCURACY

	FMNIST		CIFAR-10		CIFAR-100	
	Single	Total	Single	Total	Single	Total
FedLGA	9.4	**565.8**	12.1	**2668.6**	11.8	**1711.0**
FedAvg	**8.9**	1032.4	**10.7**	3741.5	**11.3**	3130.1
FedProx	12.2	1171.7	13.4	3932.1	12.9	2747.7
FedNova	9.1	910.0	10.9	3640.6	11.6	3120.4
Scaffold	11.2	806.7	13.1	3636.2	12.4	2287.8
FedDyn	12.2	869.3	12.8	3251.2	12.7	2057.4
FedCM	9.7	1222.8	12.4	3443.1	11.5	3108.6
MimeLite	10.7	930.9	13.8	3919.2	11.7	3192.6
FedSAM	9.3	688.2	12.5	2875.2	11.8	2537.4

consider this might be due to the reason that only 10% of local gradients are heterogeneous with E_i local epochs. And for the CIFAR-100 dataset, we can notice that the proposed FedLGA algorithm has a significant advantage over other methods when $\rho = 0.1$. Additionally, for different values of ρ, the proposed FedLGA algorithm outperforms other compared methods.

3) Evaluation on Communication Cost: Table V shows the experimental result of the running time (seconds) for each compared method to achieve the target testing accuracy. Note that to describe the performance accurately, we take both the "Single" and "Total" cost time into consideration. The Single represents the average time for running one global communication round during the training process, and the Total is the total required running time for a compared method to reach the targeted testing accuracy. We can notice that FedLGA reaches the best total running time for all of the three introduced datasets, while only the third-best on the single running time. We consider this might be because of the following reasons. Compared to FedAvg and FedNova which reach better single running time, the proposed FedLGA algorithm requires a lower number of global communication round to the target accuracy. And comparing to FedProx, Scaffold, and FedDyn, the results support our theoretical claim that as the extra computation complexity of the proposed FedLGA is on the aggregator, it outperforms other FL methods which perform extra computation costs on the remote devices.

TABLE VI
IMPACT OF $\tau_{\max}$: COMMUNICATION ROUNDS TO TARGET TESTING ACCURACY

	τ_{max}	FMNIST	CIFAR-10	CIFAR-100
FedLGA	$0.8E$	60	220	145
	$0.6E$	54	186	140
	$0.4E$	41	157	126
	$0.2E$	29	109	122
FedAvg	-	116	350	277
FedProx	-	96	293	213
FedNova	-	100	334	269
Scaffold	-	72	278	185
FedDyn	-	71	254	162
FedCM	-	126	277	269
MimeLite	-	87	284	272
FedSAM	-	74	230	215

C. Ablation Study

1) Impact of τ_{max}: We then evaluate the performance of the proposed FedLGA algorithm under further settings of the introduced hyperparameters in this article. The required communication rounds of FedLGA to achieve the target testing accuracy on the introduced dataset with different $\tau_{\max}$ values are shown in Table VI. Note that for better presentation, the performance of the compared FL methods is also introduced in the table. We can notice from the results that on each considered value of $\tau_{\max}$, FedLGA outperforms the compared FL methods. In addition, as $\tau_{\max}$ becomes larger, the performance of FedLGA degrades. We consider that this is due to the reason that when $\tau_{\max}$ is smaller, the variance of the obtained local model update approximation in FedLGA becomes larger. This may also indicate that the performance of FedLGA is also related to $E - E_i$. Specifically, when $E - E_i$ becomes larger (i.e., the FL network with higher device heterogeneity), the performance of FedLGA is more limited.

2) Impact of N: We then study the impact of the total remote device number N on the performance of the proposed FedLGA algorithm, which is illustrated in Fig. 8. Note that, we pick different $N \in \{30, 50, 100\}$ against CIFAR-10 and CIFAR-100 datasets, where other hyperparameters are set as $K = 10, \rho = 0.5$ and $E = 5$. From the results, we can

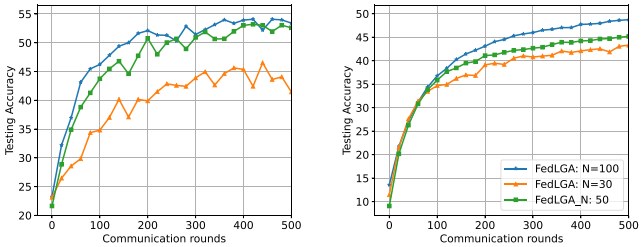

Fig. 8. Learning performance of testing accuracy for FedLGA under the different number of total remote devices N with $\rho = 0.5$, $\tau_{\max} = E - 1$ and $E = 5$. (a) CIFAR-10. (b) CIFAR-100.

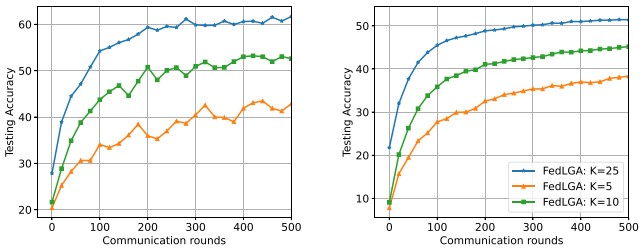

Fig. 9. Learning performance of testing accuracy for FedLGA under the different number of partial participated remote devices K in each round with $\rho = 0.5$, $\tau_{\max} = E - 1$ and $E = 5$. (a) CIFAR-10. (b) CIFAR-100.

notice that as the number of N grows, the proposed FedLGA algorithm presumes a significantly better learning performance on both the testing accuracy and the convergence speed. We can also notice an interesting phenomenon for the CIFAR-10 dataset, when $N = 30$, the performance of FedLGA has a clear gap between the settings of $N = 50$ and $N = 100$. We consider this might be because when N is too small, the variance inner each device can be too big which leads to performance degradation.

3) Impact of K: Also, we investigate the impact of the number K of partially participated remote devices in each communication round on the proposed FedLGA algorithm. Note that, we consider the different values of K as $K \in \{5, 10, 25\}$, where $N = 50, \rho = 0.5$ and $E = 5$. The results shown in Fig. 9 show that the performance of FedLGA has significant improvement as the number of K grows. For example, against the CIFAR-100 dataset, the proposed FedLGA algorithm reaches the target testing accuracy with only 78 rounds when $K = 25$, which is 46.2% faster than the performance with $K = 10$.

4) Impact of Imbalanced Remote Training Data: Finally, we investigate the impact of the imbalanced partitioned remote training data on the proposed FedLGA algorithm. In this condition, for each remote training dataset $\mathcal{X}_i$, the number of training samples can be different. For simplicity, we consider there are only two kinds of partitioned remote dataset $\mathcal{X}_{\text{large}}$ and $\mathcal{X}_{\text{small}}$ in the FL network, and a new hyperparameter ξ is introduced to denote the imbalance ratio that $\xi = (\|\mathcal{X}_{\text{large}}\|/|\mathcal{X}_{\|\text{small}\|})$. We consider three different values of ξ as $\xi \in \{1, 2, 3\}$ (when $\xi = 1$, the FL network is balanced.), where $N = 50, \rho = 0.5, E = 5$, and $P = 3, 30$ for CIFAR-10 and CIFAR100. The results shown in Fig. 10 show that

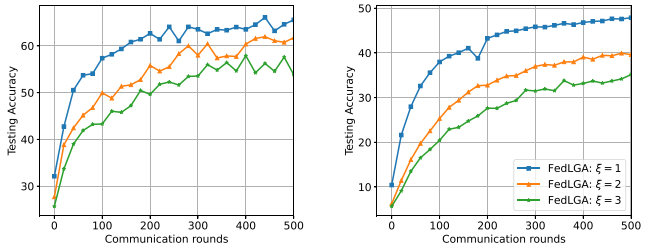

Fig. 10. Learning performance of testing accuracy for FedLGA with different remote training data imbalance ratio ξ with $\rho = 0.5$, $\tau_{\max} = E-1$ and $E = 5$. (a) CIFAR-10. (b) CIFAR-100.

though the performance of FedLGA decreases as the imbalanced ratio increases, it can still get converged and achieve the target testing accuracy under the system-heterogeneous FL network.

VII. Conclusion and Future Work

In this article, we investigate the optimization problems of FL under a system-heterogeneous network, which comes from data- and device-heterogeneity. In addition to the non-i.i.d. training data, which is known as data heterogeneity, we focus on the heterogeneous gradient updates due to the diverse computational capacities across all remote devices. To address the system-heterogeneous issue, we propose a novel algorithm FedLGA, which provides a local gradient approximation for devices with limited computational resources. Particularly, FedLGA achieves the approximation on the aggregator, which requires no extra computation on the remote device. Meanwhile, we demonstrate that the extra computation complexity of the proposed FedLGA is only linear using a Hessian approximation method. Theoretically, we show that FedLGA provides a convergence guarantee on non-convex optimization problems under system-heterogeneous FL networks. The comprehensive experiments on multiple real-world datasets show that FedLGA outperforms existing FL benchmarks in terms of different evaluation metrics, such as testing accuracy, the number of communication rounds between the aggregator and remote devices, and total running time.

The superior empirical performance of the proposed FedLGA algorithm under the system-heterogeneous network might indicate that compared to data-heterogeneity, the device-heterogeneity issue is more dominating during the joint model training. Furthermore, as FedLGA only modifies the aggregation process, an interesting future direction would be to adapt it with algorithms that modify the local optimizer [69], [78]. Meanwhile, as the communication capacity can also impact the device heterogeneity, incorporating model compression strategies [80], [81] into FedLGA can also be an important future direction.

References

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

[2] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2017, pp. 1273–1282.

[3] S. Tan, Z. Fang, Y. Wang, and J. Lü, "An augmented game approach for design and analysis of distributed learning dynamics in multiagent games," *IEEE Trans. Cybern.*, early access, May 23, 2022, doi: 10.1109/TCYB.2022.3174196.

[4] S. Lee and A. Nedic, "Distributed random projection algorithm for convex optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 221–229, Apr. 2013.

[5] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2017, pp. 1175–1191.

[6] J. Qiu et al., "Cyber code intelligence for android malware detection," *IEEE Trans. Cybern.*, vol. 53, no. 1, pp. 617–627, Jan. 2023.

[7] S. Liang, J. Lam, and H. Lin, "Secure estimation with privacy protection," *IEEE Trans. Cybern.*, early access, Mar. 8, 2022, doi: 10.1109/TCYB.2022.3151234.

[8] S. U. Stich, "Local SGD converges fast and communicates little," 2018, *arXiv:1805.09767*.

[9] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5693–5700.

[10] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[11] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–26.

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

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

[14] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[15] K. Bonawitz et al., "Towards federated learning at scale: System design," 2019, *arXiv:1902.01046*.

[16] A. Khaled, K. Mishchenko, and P. Richtárik, "Tighter theory for local SGD on identical and heterogeneous data," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2020, pp. 4519–4529.

[17] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143.

[18] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-IID federated learning," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–23.

[19] A. Hard et al., "Federated learning for mobile keyboard prediction," 2018, *arXiv:1811.03604*.

[20] A. Vaid et al., "Federated learning of electronic health records to improve mortality prediction in hospitalized patients with COVID-19: Machine learning approach," *JMIR Med. Inform.*, vol. 9, no. 1, 2021, Art. no. e24207.

[21] I. Dayan et al., "Federated learning for predicting clinical outcomes in patients with COVID-19," *Nat. Med.*, vol. 27, no. 10, pp. 1735–1743, 2021.

[22] R. Coulter, Q.-L. Han, L. Pan, J. Zhang, and Y. Xiang, "Data-driven cyber security in perspective—Intelligent traffic analysis," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3081–3093, Jul. 2020.

[23] X.-M. Li, Q. Zhou, P. Li, H. Li, and R. Lu, "Event-triggered consensus control for multi-agent systems against false data-injection attacks," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 1856–1866, May 2020.

[24] H. Li, Y. Wu, and M. Chen, "Adaptive fault-tolerant tracking control for discrete-time multiagent systems via reinforcement learning algorithm," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1163–1174, Mar. 2021.

[25] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated-learning-based anomaly detection for IoT security attacks," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2545–2554, Feb. 2022.

[26] L. Zhang, W. Cui, B. Li, Z. Chen, M. Wu, and T. S. Gee, "Privacy-preserving cross-environment human activity recognition," *IEEE Trans. Cybern.*, vol. 53, no. 3, pp. 1765–1775, Mar. 2023, doi: 10.1109/TCYB.2021.3126831.

[27] Y. Liu, X. Dong, P. Shi, Z. Ren, and J. Liu, "Distributed fault-tolerant formation tracking control for multiagent systems with multiple leaders and constrained actuators," *IEEE Trans. Cybern.*, early access, Jan. 26, 2022, doi: 10.1109/TCYB.2022.3141734.

[28] Y. Wang, J. Lam, and H. Lin, "Consensus of linear multivariable discrete-time multiagent systems: Differential privacy perspective," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13915–13926, Dec. 2022.

[29] W. Chen, Z. Wang, J. Hu, and G.-P. Liu, "Differentially private average consensus with logarithmic dynamic encoding–decoding scheme," *IEEE Trans. Cybern.*, early access, Jan. 12, 2023, doi: 10.1109/TCYB.2022.3233296.

[30] K. Wei, C. Deng, X. Yang, and D. Tao, "Incremental zero-shot learning," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13788–13799, Dec. 2022.

[31] J. Hu, Z. Wang, and G.-P. Liu, "Delay compensation-based state estimation for time-varying complex networks with incomplete observations and dynamical bias," *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 12071–12083, Nov. 2022.

[32] J. Le, X. Lei, N. Mu, H. Zhang, K. Zeng, and X. Liao, "Federated continuous learning with broad network architecture," *IEEE Trans. Cybern.*, vol. 51, no. 8, pp. 3874–3888, Aug. 2021.

[33] J.-Y. Li, K.-J. Du, Z.-H. Zhan, H. Wang, and J. Zhang, "Distributed differential evolution with adaptive resource allocation," *IEEE Trans. Cybern.*, early access, Mar. 14, 2022, doi: 10.1109/TCYB.2022.3153964.

[34] G. Chen et al., "Neuromorphic vision-based fall localization in event streams with temporal–spatial attention weighted network," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9251–9262, Sep. 2022.

[35] Y. Zhang, Q. Yang, D. An, D. Li, and Z. Wu, "Multistep multiagent reinforcement learning for optimal energy schedule strategy of charging stations in smart grid," *IEEE Trans. Cybern.*, early access, Apr. 27, 2022, doi: 10.1109/TCYB.2022.3165074.

[36] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[37] X. Li, Z. Qu, B. Tang, and Z. Lu, "Stragglers are not disaster: A hybrid federated learning algorithm with delayed gradients," 2021, *arXiv:2102.06329*.

[38] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[39] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, "Federated Meta-learning with fast convergence and efficient communication," 2018, *arXiv:1802.07876*.

[40] M. Khodak, M.-F. F. Balcan, and A. S. Talwalkar, "Adaptive gradient-based Meta-learning methods," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 5917–5928.

[41] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4424–4434.

[42] Z. Qu, R. Duan, L. Chen, J. Xu, Z. Lu, and Y. Liu, "Context-aware online client selection for hierarchical federated learning," 2021, *arXiv:2112.00925*.

[43] F. Zhang, Y. Mei, S. Nguyen, K. C. Tan, and M. Zhang, "Multitask genetic programming-based generative hyperheuristics: A case study in dynamic scheduling," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10515–10528, Oct. 2022.

[44] Q. Van Tran, Z. Sun, B. D. O. Anderson, and H.-S. Ahn, "Distributed optimization for graph matching," *IEEE Trans. Cybern.*, early access, Jan. 26, 2022, doi: 10.1109/TCYB.2022.3140338.

[45] S. Reddi et al., "Adaptive federated optimization," 2020, *arXiv:2003.00295*.

[46] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," 2021, *arXiv:2111.04263*.

[47] P. Kairouz et al., "Advances and open problems in federated learning," 2019, *arXiv:1912.04977*.

[48] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1–13.

[49] S. U. Stich and S. P. Karimireddy, "The error-feedback framework: Better rates for SGD with delayed gradients and compressed communication," 2019, *arXiv:1909.05350*.

[50] Y. Arjevani, O. Shamir, and N. Srebro, "A tight convergence analysis for stochastic gradient descent with delayed updates," in *Proc. Int. Conf. Algorithmic Learn. Theory*, 2020, pp. 111–132.

[51] M. Glasgow and M. Wootters, "Asynchronous distributed optimization with stochastic delays," 2020, *arXiv:2009.10717*.

[52] K. Hsieh, A. Phanishayee, O. Mutlu, and P. B. Gibbons, "The non-IID data quagmire of decentralized machine learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4387–4398.
[53] G. B. Folland, "Remainder estimates in Taylor's theorem," *Amer. Math. Monthly*, vol. 97, no. 3, pp. 233–235, 1990.
[54] C. Bischof, G. Corliss, and A. Griewank, "Structured second-and higher-order derivatives through univariate Taylor series," *Optim. Methods Softw.*, vol. 2, nos. 3–4, pp. 211–232, 1993.
[55] S. Zheng et al., "Asynchronous stochastic gradient descent with delay compensation," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 4120–4129.
[56] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous federated optimization," 2019, *arXiv:1903.03934*.
[57] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. New York, NY, USA: Springer, 2009, pp. 1–758.
[58] R. Pascanu and Y. Bengio, "Revisiting natural gradient for deep networks," 2013, *arXiv:1301.3584*.
[59] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, "The loss surfaces of multilayer networks," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2015, pp. 192–204.
[60] K. Kawaguchi, "Deep learning without poor local minima," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 586–594.
[61] C. Xu, Y. Qu, Y. Xiang, and L. Gao, "Asynchronous federated learning on heterogeneous devices: A survey," 2021, *arXiv:2109.04269*.
[62] A. Imteaj and M. H. Amini, "FedAR: Activity and resource-aware federated learning model for distributed mobile robots," in *Proc. 19th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, 2020, pp. 1153–1160.
[63] Z. Chen, W. Liao, K. Hua, C. Lu, and W. Yu, "Towards asynchronous federated learning for heterogeneous edge-powered Internet of Things," *Digit. Commun. Netw.*, vol. 7, no. 3, pp. 317–326, 2021.
[64] Z. Chai, Y. Chen, L. Zhao, Y. Cheng, and H. Rangwala, "FedAT: A high-performance and communication-efficient federated learning system with asynchronous tiers," 2020, *arXiv:2010.05958*.
[65] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2341–2368, 2013.
[66] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Rev.*, vol. 60, no. 2, pp. 223–311, 2018.
[67] X. Li, Z. Qu, B. Tang, and Z. Lu, "FedLGA: Towards system-heterogeneity of federated learning via local gradient approximation," 2021, *arXiv:2112.11989*.
[68] A. Khaled, K. Mishchenko, and P. Richtárik, "First analysis of local GD on heterogeneous data," 2019, *arXiv:1909.04715*.
[69] Z. Qu, X. Li, R. Duan, Y. Liu, B. Tang, and Z. Lu, "Generalized federated learning via sharpness aware minimization," 2022, *arXiv:2206.02618*.
[70] J. Nguyen et al., "Federated learning with buffered asynchronous aggregation," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2022, pp. 3581–3607.
[71] A. Paszke et al., "Automatic differentiation in PyTorch," in *Proc. NIPS-W*, 2017, pp. 1–4.
[72] P. P. Liang et al., "Think locally, act globally: Federated learning with local and global representations," 2020, *arXiv:2001.01523*.
[73] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[74] A. Krizhevsky, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
[75] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 315–323.
[76] M. Schmidt, N. Le Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," *Math. Program.*, vol. 162, nos. 1–2, pp. 83–112, 2017.
[77] J. Xu, S. Wang, L. Wang, and A. C.-C. Yao, "FedCM: Federated learning with client-level momentum," 2021, *arXiv:2106.10874*.
[78] S. P. Karimireddy et al., "Mime: Mimicking centralized stochastic algorithms in federated learning," 2020, *arXiv:2008.03606*.
[79] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–19.
[80] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–14.
[81] A. Dutta et al., "On the discrepancy between the theoretical analysis and practical implementations of compressed communication for distributed deep learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 3817–3824.

Xingyu Li received the B.S. degree from the School of Electronic Science and Engineering, National Model Microelectronics College, Xiamen University, Xiamen, China, in 2015, and the M.S. degree from the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA, in 2017. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS, USA. His current research interests include federated learning and continual learning.

Zhe Qu (Graduate Student Member, IEEE) received the B.S. degree from Xiamen University, Xiamen, China, in 2015, and the M.S. degree from the University of Delaware, Newark, DE, USA, in 2017. He is currently pursuing the Ph.D. degree in systems and security with the Department of Electrical Engineering, University of South Florida, Tampa, FL, USA.

His primary research interests include network and mobile system security and machine learning for networks.

Bo Tang (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Rhode Island, Kingstown, RI, USA, in 2016.

He is an Associate Professor with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute (WPI), Worcester, MA, USA. Before joining WPI, he was an Assistant Professor with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MI, USA. His research interests lie in the general areas of statistical machine learning and data mining, as well as their various applications in cyber–physical systems, including robotics, autonomous driving, and remote sensing.

Dr. Tang is the recipient of the NSF CAREER Award in 2021.

Zhuo Lu (Senior Member, IEEE) received the Ph.D. degree from North Carolina State University, Raleigh, NC, USA, in 2013.

He is an Associate Professor with the Department of Electrical Engineering, University of South Florida, Tampa, FL, USA. He is also affiliated with the Florida Center for Cybersecurity and by courtesy with Department of Computer Science and Engineering. His research has been mainly focused on both theoretical and system perspectives on communication, network, and security. His recent research is equally focused on machine learning and AI perspectives on networking and security.

Dr. Lu received the NSF CISE CRII Award in 2016, the Best Paper Award from IEEE GlobalSIP in 2019, and the NSF CAREER Award in 2021.