A New Look and Convergence Rate of Federated Multitask Learning With Laplacian Regularization

Canh T. Dinh, Tung T. Vu, *Member, IEEE*, Nguyen H. Tran, *Senior Member, IEEE*, Minh N. Dao, and Hongyu Zhang, *Senior Member, IEEE*

***Abstract*— Non-independent and identically distributed (non-IID) data distribution among clients is considered as the key factor that degrades the performance of federated learning (FL). Several approaches to handle non-IID data, such as personalized FL and federated multitask learning (FMTL), are of great interest to research communities. In this work, first, we formulate the FMTL problem using Laplacian regularization to explicitly leverage the relationships among the models of clients for multitask learning. Then, we introduce a new view of the FMTL problem, which, for the first time, shows that the formulated FMTL problem can be used for conventional FL and personalized FL. We also propose two algorithms FedU and decentralized FedU (dFedU) to solve the formulated FMTL problem in communication-centralized and decentralized schemes, respectively. Theoretically, we prove that the convergence rates of both algorithms achieve linear speedup for strongly convex and sublinear speedup of order 1/2 for nonconvex objectives. Experimentally, we show that our algorithms outperform the conventional algorithm FedAvg, FedProx, SCAFFOLD, and AFL in FL settings, MOCHA in FMTL settings, as well as pFedMe and Per-FedAvg in personalized FL settings.**

***Index Terms*— Federated learning (FL), federated multitask learning (FMTL), Laplacian regularization, personalized learning.**

I. Introduction

RECENTLY, federated learning (FL) has been considered as a promising distributed and privacy-preserving method for building a global model from a massive number of handheld devices [1], [2], [3], [4]. FL has a wide range of futuristic applications, such as detecting the symptoms of possible diseases (e.g., stroke, heart attack, and diabetes) from wearable devices in health-care systems [5], [6], [7], or predicting disaster risks from the Internet-of-Things devices in smart cities [8], [9]. In FL, one of the key challenges is the naturally non-independent and identically distributed (non-IID) data distributions among clients [10], [11]. When the differences among clients' data distributions increase, the generalization error of the FL global model on each client's local data significantly increases [12], [13].

Personalized FL [14], [15] and federated multitask learning (FMTL) [16] have been proposed as solutions to handle non-IID data distributions among clients. Personalized FL aims to build a global model that is leveraged to find a "personalized model" for each client's local data. Here, the global model is considered as an "agreed point" for each client to start personalizing its model based on its heterogeneous local data distribution. Different from personalized FL, FMTL aims to simultaneously learn separate models, which is motivated by multitask learning frameworks [17], [18]. Each of these models fits the data distribution of each client. Therefore, FMTL directly addresses the issue stemming from non-IID data distributions without building any global model as personalized FL.

On the other hand, from the aspect of the local data at clients, it is observed that the clients with similar features (e.g., location, time, age, and gender) are likely to share similar behaviors. Therefore, although the clients' models are separated, they are normally related to each other. In FMTL, the relationships among the clients' models are captured by a regularization term, which is minimized to encourage the clients' models to be mutually impacted. Unfortunately, these relationships have not been clearly taken into consideration in the FMTL problem. Moreover, communication-decentralized and nonconvex FMTL algorithms with guaranteed convergence are generally less explored.

The main contributions of this work are as follows.

1) We formulate an FMTL problem using Laplacian regularization to explicitly leverage the relationships among the models of clients. We then introduce a new view of the FMTL problem that the formulated FMTL problem can be used not only for the conventional FL but also for the personalized FL.
2) We propose a communication-centralized FMTL algorithm FedU and decentralized FedU (dFedU) to solve the formulated FMTL problem. We also analyze the convergence rate of FMTL algorithms with both

Manuscript received 14 December 2021; revised 6 June 2022 and 11 October 2022; accepted 12 November 2022. Date of publication 7 December 2022; date of current version 4 June 2024. The work of Tung T. Vu and Hongyu Zhang was supported by the Australian Research Council's Discovery Projects under Grant DP200102940 and Grant DP220103044. The work of Minh N. Dao was supported by a public grant as part of the Investissement d'avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH. *(Corresponding author: Canh T. Dinh.)*

Canh T. Dinh and Nguyen H. Tran are with the School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: canh.dinh@sydney.edu.au; nguyen.tran@sydney.edu.au).

Tung T. Vu is with the Department of Electrical Engineering (ISY), Linköping University, SE-581 83 Linköping, Sweden (e-mail: thanh.tung.vu@liu.se).

Minh N. Dao is with the School of Science, RMIT University, Melbourne, VIC 3000, Australia (e-mail: minh.dao@rmit.edu.au).

Hongyu Zhang is with The University of Newcastle, Callaghan, NSW 2308, Australia (e-mail: hongyu.zhang@newcastle.edu.au).

Digital Object Identifier 10.1109/TNNLS.2022.3224252

convex and nonconvex objective functions. In particular, FedU and dFedU are proved to achieve a linear speedup (resp. sublinear speedup of order $1/2$) for strongly convex (resp. nonconvex) objective cases.

3) We empirically evaluate the performance of FedU and dFedU using real datasets that capture the non-IID data distribution among clients. We show that in terms of local accuracy, FedU and dFedU outperform the traditional algorithm FedAvg in FL settings, the conventional algorithm MOCHA in FMTL settings, as well as pFedMe and Per-FedAvg in personalized FL settings.

II. Related Work

A. Federated Learning

One of the earliest work of FL is FedAvg [1], which builds the global model based on averaging the local stochastic gradient descent (SGD) updates. Various methods [11], [19], [20], [21], [22] are introduced to improve the robustness of the global model under non-IID settings. For example, FedProx [19] adds a proximal term to the local objective, therefore addressing the statistical heterogeneity of clients.

B. Personalized FL

Several personalized FL approaches have been proposed to tackle the issues stemming from non-IID data in the conventional FL. Mixture methods [13], [23] attempted to combine a local model with the global model, while [24] applied this mixing to jointly learn compact local representations on each client and a global model across all clients. Motivating by creating a well-generalized global model to quickly adapt to client's data after few gradient descent steps, pFedMe [14] used Moreau envelopes, while Per-FedAvg [15] took advances of meta-learning approaches: model-agnostic meta-learning [25]. Jiang et al. [26] proposed the combination of FedAvg and Reptile [27] to improve FL personalization. A different personalized FL approach to train deep neural networks (DNNs) is FedPer [28]. Clients share a set of base layers with a server and keep personalization layers that adapt quickly to the local data.

C. Federated Multitask Learning

Another approach to deal with the non-IID data distributions at clients is learning separate models each of which fits each local data distribution. In this sense, FMTL was first introduced in [16] where a systems-aware optimization framework MOCHA for handling stragglers and fault tolerance in FL settings is proposed. Besides that, there are also several other works studying FMTL. Sarcheshmehpour et al. [29] proposed a framework for generalized total variation minimization, which is useful in FMTL networks. Li et al. [30] introduced an FMTL algorithm to deal with the issues of accuracy, fairness, and robustness in FL. By treating the FL network as a star-shaped Bayesian network, Shen et al. [31] developed an FMTL algorithm using approximated variational inference. Li et al. [32] focused on an FMTL algorithm for online applications. However, in all these works, the convergence rate of FMTL with nonconvex objectives has not been studied. Moreover, the relations among the problems of FMTL, the standard FL, and personalized FL are not yet investigated in the literature.

III. FMTL: New View

A. Formulation of the FMTL Problem With Laplacian Regularization

In this work, the goal of FMTL is to fit separate models (i.e., $w_k \in \mathbb{R}^d, \forall k \in \mathcal{N}$) to the local data of clients, taking into account the relationships among these models. For instance, smart-device clients in a mobile network are trying to learn their activities using their personal and private data (e.g., image, text, voice, and sensor data). In FL settings, their data may come from different environments, contexts, and applications and, thus, have non-IID distributions. Despite of this, these clients are likely to behave similarly under similar features or scenarios (e.g., location, time, and age). Therefore, there normally exist relationships among the models of clients [33], [34], [35].

To present the relationships among the models of clients, we consider a connected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}, A\}$, where $\mathcal{N} := \{1, \ldots, N\}$ is the set of vertices representing FL clients, $\mathcal{E}$ is the set of edges representing relationships among the models of clients, and $A \in \mathbb{R}^N$ is a symmetric, weighted adjacency matrix with $a_{k\ell} := [A]_{k\ell}$. The relationship between clients k and ℓ is presented by $a_{k\ell}$ and reversible, i.e., $a_{k\ell} = a_{\ell k}, \forall k, \ell$. Here, $a_{k\ell} = 0$ means no relationship between the models of clients k and ℓ. The value of $a_{k\ell} > 0$ shows that client k is a neighbor of client ℓ and also determines the strength of the relationship between these two clients' models. Let $D \in \mathbb{R}^N$ be a diagonal matrix in which $[D]_{kk} = \sum_{\ell=1}^{N} a_{k\ell}$. The Laplacian matrix of the graph is, thus, $L = D - A$.

Let $W = [w_1^T, \ldots, w_N^T]^T \in \mathbb{R}^{dN}$ be a collective model vector, and $\mathcal{L} := L \otimes I_d$ be a Laplacian regularization matrix. Now, we formulate the following FMTL problem:

$$\min_W J(W) = \underbrace{F(W)}_{\text{Global loss}} + \underbrace{\eta R(W)}_{\text{Laplacian regularization}} \tag{1}$$

where

$$F(W) = \sum_{k=1}^{N} F_k(w_k) \tag{2}$$

$$\mathcal{R}(W) = W^T \mathcal{L} W = \frac{1}{2} \sum_{k=1}^{N} \sum_{\ell \in \mathcal{N}_k} a_{k\ell} ||w_k - w_\ell||^2 \tag{3}$$

$\mathcal{N}_k = \mathcal{N} \setminus \{k\}$, and $\|\cdot\|$ is the Euclidean norm. $F_k(\cdot)$ represents the expected loss function at client k

$$F_k(w_k) = \mathbb{E}_{\zeta_k}[f_k(w_k; \zeta_k)]$$

where ζ_k is a random data sample drawn from the distribution of client k, and $f_k(w_k; \zeta_k)$ is the regularized loss function corresponding to this sample and w_k. The distribution of ζ_k and ζ_ℓ can be distinct when $k \neq \ell$.

Note that in our work, we do not extract the similarity of the existing relationships between the clients by any visualization

methods in order to develop our proposed method. Instead, we present the existing relationships among the models of the clients by a Laplacian regularization matrix $\mathcal{L}$ and put it into the Laplacian regularization term in the objective function of the FMTL problem (1). Theoretically, in (1), $\eta \geq 0$ is a regularization hyperparameter that controls the impact of the models of neighboring clients on each local model. If $\eta = 0$, (1) turns to an individual learning problem where each client learns its local model w_k based on its own local data without collaboration with server or other clients. If $\eta > 0$, minimizing the Laplacian regularization term encourages the models of the neighboring clients to be close to each other. The impacts of the existing relationship between the models of the clients on the performance of our proposed algorithms will be shown in Section VI of experiment.

Remark 1: There are other methods of regularization to encourage the models of the neighboring clients to be close to each other, e.g., using $||w_k - w_\ell||$ instead of $||w_k - w_\ell||^2$ in (3) as Network Lasso does [36], [37], [38], or using $\text{tr}(\widehat{W}\Omega\widehat{W}^T)$ instead of (3) as MOCHA does [16], where $\widehat{W} := [w_1, \ldots, w_N] \in \mathbb{R}^{d\times N}$. On the other hand, problem (1) is a generalization of the problem in [39] where several algorithms are developed for strongly convex objectives. Problem (1) is also similar to the generalized total variation minimization problem [29], which is solved by a primal–dual method for convex objectives. Vanhaesebrouck et al. [40] have a convex version of problem (1), which is solved by a decentralized algorithm using an alternating direction method of multipliers (ADMM). In (1), we present the FTML problem using the Laplacian regularization matrix $\mathcal{L}$. Utilizing the special properties of $\mathcal{L}$, we successfully design FMTL algorithms using SGD. Importantly, our algorithms can work in the following: 1) in both centralized and decentralized communication schemes and 2) with both strongly convex and nonconvex objective functions.

Assumption 1 (Smoothness): For each $k \in \mathcal{N}$, F_k is β-smooth, i.e., for any $w, w' \in \mathbb{R}^d$

$$\|\nabla F_k(w) - \nabla F_k(w')\| \leq \beta\|w - w'\|.$$

Assumption 2 (Strong Convexity): For each $k \in \mathcal{N}$, F_k is α-strongly convex, i.e., for any $w, w' \in \mathbb{R}^d$

$$F_k(w) \geq F_k(w') + \langle \nabla F_k(w'), w - w' \rangle + \frac{\alpha}{2}\|w - w'\|^2.$$

Assumption 3 (Bounded Variance): The set of $\nabla \widetilde{F}_k(w, \zeta_k)$, $k \in \mathcal{N}$, is unbiased stochastic gradients of $\nabla F_k(w)$, $k \in \mathcal{N}$, with total variance bounded by σ_1^2, i.e., for any $W \in \mathbb{R}^{dN}$

$$\sum_{k=1}^{N} \mathbb{E}_{\zeta_k} \|\nabla \widetilde{F}_k(w_k, \zeta_k) - \nabla F_k(w_k)\|^2 \leq \sigma_1^2.$$

We note that Assumption 3 is weaker than the assumption of individual bounded variance that is used at each client in FL and personalized FL problems [10], [14], [15]. It should also be noted that (1) shares some similarities to the multitask learning problem of [41] and [42]. However, the latter requires that each $F_k(w_k)$ is twice differential with the Hessian $\nabla^2_{w_k} F_k(w_k)$ uniformly bounded from below and above, which is more restrictive than our assumptions. Moreover, this problem does not take into account the issue of non-IID data distributions among clients, and thus, it is not formulated for FL settings.

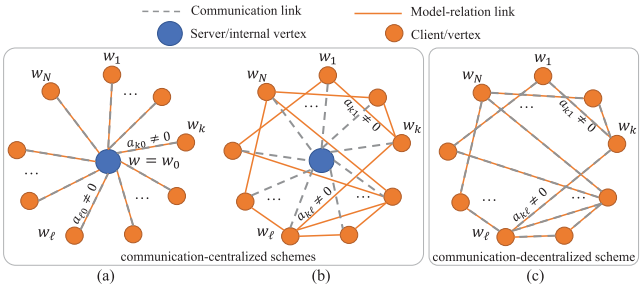

Fig. 1. Illustrations of undirected weighted graphs in FL. (a) Star graph with a server for traditional FL and personalized FL. (b) and (c) Entity graph with and without server for FMTL.

B. New View of the FMTL Problem

We first observe that in conventional FL and personalized FL, all clients connect to a server under a communication-centralized scheme shown in Fig. 1(a). The relationships among the models of the clients and the server are presented by a star graph. In this graph, a server is considered as a virtually internal vertex 0 with its loss function $F_0 = 0$ and a model w_0. Here, all the models of clients are only related to the server model w_0, i.e., $a_{k0} > 0, \forall k$, but not with each other, i.e., $a_{k\ell} = 0, \forall k, \ell \neq 0$. In this work, we assume that the weights $a_{k\ell}$ are known and focus on the development of FMTL algorithms to solve problem (1). The finding of $a_{k\ell}$ in specific learning applications is referred to [43] and [44]. In what follows, we show that the formulated FMTL problem (1) can be used for the conventional FL and some types of personalized FL. For a more general optimization problem of personalized FL, we refer to LSGD-PFL [45].

1) Relation of FMTL to Conventional FL: The objective function of (1) can be seen as a Lagrangian function of the following problem:

$$\min_{W} \sum_{k=1}^{N} F_k(w_k), \quad \text{s.t. } w_1 = w_2 = \cdots = w_N \tag{4}$$

which is equivalent to the conventional FL problem (FedAvg) [1]. Therefore, the solution of the conventional FL problem can be obtained by solving (1).

2) Relation of FMTL to Personalized FL With Moreau Envelopes (pFedMe): The problem of pFedMe [14] is formulated as follows:

$$\min_{w} J(w) = \sum_{k=1}^{N} \tilde{J}_k(w) \tag{5}$$

where $\tilde{J}_k(w) = \min_{z_k} F_k(z_k) + (\eta/2)||z_k - w||^2$. We observe that

$$J(w) = \sum_{k=1}^{N} \min_{z_k} \left(F_k(z_k) + \frac{\eta}{2}\|z_k - w\|^2 \right)$$

$$= \min_{z_1,\ldots,z_N} \sum_{k=1}^{N} \left(F_k(z_k) + \frac{\eta}{2}\|z_k - w\|^2 \right).$$

Therefore, (5) is equivalent to the following problem with $z_0 = w$ and $F_0 \equiv 0$:

$$\min_{z_0,z_1,\ldots,z_N} \sum_{k=0}^{N} F_k(z_k) + \frac{\eta}{2} \sum_{k=0}^{N} \|z_k - z_0\|^2$$

which is a special case of (1) with the star graph topology and $a_{k0} = 1, \forall k \in \mathcal{N}$.

3) Relation of FMTL to Meta-Learning-Based Personalized FL (Per-FedAvg): The problem of Per-FedAvg [15] is given by

$$\min_{w} \; J(w) = \textstyle\sum_{k=1}^{N} F_k(w - \mu \nabla F_k(w)) \tag{6}$$

where $\mu > 0$, and each F_k is assumed to be L_k-Lipschitz continuous. Set $w_k = w - \mu \nabla F_k(w)$, and $\ell_k = (L_k/2), k \in \mathcal{N}$. Using [46, Lemma 1.2.3] twice, we have that, for $\mu < \min_k \ell_k$ and for all $z_k \in \mathbb{R}^d$

$$\begin{aligned} F_k(w_k) &\le F_k(w) + \langle \nabla F_k(w), w_k - w \rangle + \ell_k \|w_k - w\|^2 \\ &= F_k(w) - (\mu - \ell_k \mu^2) \|\nabla F_k(w)\|^2 \\ &\le F_k(z_k) + \langle \nabla F_k(w), z_k - w \rangle + \ell_k \|z_k - w\|^2 \\ &\quad - (\mu - \ell_k \mu^2) \|\nabla F_k(w)\|^2 \\ &= F_k(z_k) + a_{k0} \|z_k - w\|^2 \\ &\quad - (\mu - \ell_k \mu^2) \left\| \nabla F_k(w) - \frac{z_k - w}{2(\mu - \ell_k \mu^2)} \right\|^2 \end{aligned}$$

where $a_{k0} := \ell_k + (1/4(\mu - \ell_k \mu^2))$. Hence

$$F_k(w_k) \le \min_{z_k} \left(F_k(z_k) + a_{k0} \|z_k - w\|^2 \right)$$

which implies that

$$\begin{aligned} J(w) &\le \sum_{k=1}^{N} \min_{z_k} \left(F_k(z_k) + a_{k0} \|z_k - w\|^2 \right) \\ &= \min_{z_1,\ldots,z_N} \sum_{k=1}^{N} \left(F_k(z_k) + a_{k0} \|z_k - w\|^2 \right). \end{aligned}$$

Now, (6) can be solved through its following epigraph problem with $z_0 = w$ and $F_0 = 0$:

$$\min_{z_0,z_1,\ldots,z_N} \sum_{k=0}^{N} F_k(z_k) + \frac{\eta}{2} \sum_{k=0}^{N} a_{k0} \|z_k - z_0\|^2$$

which is also a special case of (1) with the star graph topology and $a_{k0} = 1, \forall k \in \mathcal{N}$.

IV. FMTL: Algorithms

A. FedU: Communication-Centralized Algorithm

In this section, we propose an algorithm FedU, which is presented in Algorithm 1, to solve the formulated FL problem (1) under the communication-centralized scheme. Here, we use an entity graph to capture the relationships among the models of clients, as shown in Fig. 1(b).[1] First, the server uniformly samples a subset of clients $\mathcal{S}^{(t)}$ and sends the latest update of local model w_k to each client $k, \forall k \in \mathcal{S}^{(t)}$. Then, after R local update steps are performed, the server receives the latest local update from the sampled clients to perform model regularization for each local model.

[1]In an entity graph, each vertex is a value of an entity (e.g., a person) and an edge (e.g., friendship) between two entities exists if these entities are perceived to be similar [43].

Algorithm 1 FedU

client k**'s input**: local step-size μ
server's input: graph information $\{a_{k\ell}\}$, initial $w_k^{(0)}, \forall k \in \mathcal{N}$, and global step-size $\widetilde{\mu} = \mu R$
for each round $t = 0, \ldots, T-1$ **do**
  server uniformly samples a subset of clients $\mathcal{S}^{(t)}$ of size S and sends $w_k^{(t)}$ to client $k, \forall k \in \mathcal{S}^{(t)}$
  on client $k \in \mathcal{S}^{(t)}$ **in parallel do**
    initialize local model $w_{k,0}^{(t)} \leftarrow w_k^{(t)}$
    for $r = 0, \ldots, R-1$ **do**
      compute mini-batch gradient $\nabla \widetilde{F}_k(w_{k,r}^{(t)})$
      $w_{k,r+1}^{(t)} \leftarrow w_{k,r}^{(t)} - \mu \nabla \widetilde{F}_k(w_{k,r}^{(t)})$
    end for
    send $w_{k,R}^{(t)}$ to the server
  end on client
  on server do
    $w_{k,R}^{(t)} \leftarrow w_k^{(t)}, \forall k \notin \mathcal{S}^{(t)}$
    $w_k^{(t+1)} \leftarrow w_{k,R}^{(t)} - \widetilde{\mu}\, \eta \sum_{\ell \in \mathcal{N}_k} a_{k\ell} (w_{k,R}^{(t)} - w_{\ell,R}^{(t)}), \forall k \in \mathcal{S}^{(t)}$
    $w_k^{(t+1)} \leftarrow w_k^{(t)}, \forall k \notin \mathcal{S}^{(t)}$
  end on server
end for

Note that in the entity graph, the models of clients are only related to other models but not to any server model, as in the star graph of the conventional FL and personalized FL. Therefore, FedU has a key difference compared with the conventional FL algorithms (e.g., FedAvg [1]) and the personalized FL algorithms (e.g., pFedMe [14] and Per-FedAvg [15]). Instead of updating the personalized models only at the clients using a global model from the server, FedU directly updates each local model at both client and server sides without building a global model.

Specifically, in each communication round, each client $k \in \mathcal{S}^{(t)}$ copies its current local model received from the server: $w_{k,0}^{(t)} = w_k^{(t)}$, and performs R local updates of the form

$$w_{k,r+1}^{(t)} \leftarrow w_{k,r}^{(t)} - \mu \nabla \widetilde{F}_k \left(w_{k,r}^{(t)} \right)$$

where μ is the local step size. Then, server receives $\{w_{k,R}^{(t)}\}$ from sampled clients $k \in \mathcal{S}^{(t)}$ and updates

$$w_{k,R}^{(t)} \leftarrow w_k^{(t)}$$

for any nonsampled client $k \notin \mathcal{S}^{(t)}$. Finally, the server performs its regularization update for any sampled client $k \in \mathcal{S}^{(t)}$ as follows:

$$w_k^{(t+1)} \leftarrow w_{k,R}^{(t)} - \widetilde{\mu}\, \eta \sum_{\ell \in \mathcal{N}_k \cap \mathcal{S}^{(t)}} a_{k\ell} \left(w_{k,R}^{(t)} - w_{\ell,R}^{(t)} \right)$$

and for any nonsampled client $k \notin \mathcal{S}^{(t)}$ as follows:

$$w_k^{(t+1)} \leftarrow w_k^{(t)}$$

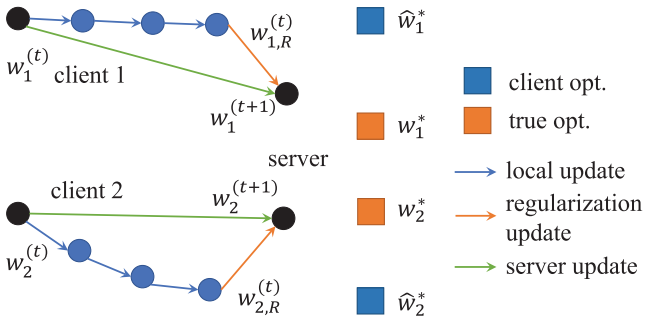

Fig. 2. Update steps of FedU at both client and server sides are illustrated for two related tasks (clients) with three local steps ($N = 2, R = 3$) at round t. The local updates $w_{k,r}^{(t)}$ (blue circles) move toward the client optima $\widehat{w}_k^*$ (blue square). The regularization updates (in orange) ensure the server update (in green) moves toward the true optimum $w_k^*, \forall k \in \mathcal{N}$ (orange square).

where $\widetilde{\mu} = \mu R$ is a global step size. This step finishes one round of communication.

The mechanism of FedU is explained with $N = 2$ example clients, as shown in Fig. 2. The two clients are the neighbors of each other and share a certain similarity model. Let (w_1^*, w_2^*) be the global solution (true optimum or true opt.) to problem (1), which is presented by orange squares. Let $(\widehat{w}_1^*, \widehat{w}_2^*)$ be the local solution (client optimum or client opt.) that obtains the minimum of the local lost function $F_k(w_k)$, which is presented by blue squares. In the case of non-IID data, $\widehat{w}_1^*$ and $\widehat{w}_2^*$ are far away from each other, and $(\widehat{w}_1^*, \widehat{w}_2^*)$ is also far away from (w_1^*, w_2^*). At round t, after making $R = 3$ local updates, the updated models $(w_{1,R}^{(t)}, w_{2,R}^{(t)}$; blue circles) are moved closer to $(\widehat{w}_1^*, \widehat{w}_2^*)$. Then, we make a further step of regularization update in order to move $w_{1,R}^{(t)}$ toward $\widehat{w}_2^*$ and also move $w_{2,R}^{(t)}$ toward $\widehat{w}_1^*$, which finally makes the updated model after round t, i.e., $(w_1^{(t+1)}, w_2^{(t+1)})$, closer to (w_1^*, w_2^*). By doing local and regularization updates in each round, the converged solution of FedU will be (w_1^*, w_2^*).

B. dFedU: Decentralized Version of FedU

We note that the server in FedU needs to know all the graph information $\{a_{k\ell}\}$. This requirement can be achieved by letting all the clients send the information of their neighbors to the server at the beginning of the learning process. However, in a network of massive clients (e.g., thousands), it might be impractical to maintain all the information of the graph (e.g., vertices and weighted edge) as well as storage for all model updates at the server. This motivates us to propose dFedU, which is a decentralized version of FedU, and presented in Algorithm 2.

Specifically, in each communication round, each client of an entity graph [as shown in Fig. 1(c)] performs R local updates and sends its updated model to their neighboring clients to perform the model regularization. Here, each client does not need to communicate with the rest of the large number of clients in the whole network. Each client only needs to communicate with its neighbor clients. A client ℓ is a neighbor of client k if and only if it has a communication link (i.e., $a_{k\ell} \neq 0$) and shares a certain model similarity with client k (i.e., $a_{k\ell} > 0$). The set of neighboring clients of client k is defined as $\widetilde{\mathcal{N}}_k = \{\ell \mid a_{k\ell} > 0\}$. Note that because there is no server for coordinating the learning, there is no client sampling in dFedU. Compared with the non-FL decentralized scheme [41], [42], dFedU uses R local updates, which are typical in FL algorithm designs.

V. FMTL: Convergence Rate

In this section, we present the convergence rate of FedU and dFedU. Let $W^* = [w_1^*, \ldots, w_N^*]$ be the optimal solution to (1).

Lemma 1: Suppose that Assumption 1 holds and $\eta\rho > 2\beta$, where $\rho := \|\mathcal{L}\|$. Then, there exists $\sigma_2 \geq 0$, e.g., $\sigma_2 = \|\nabla F(0)\|(\eta\rho/(\eta\rho - 2\beta))^{1/2}$, such that, for any $W \in \mathbb{R}^{dN}$

$$\sum_{k=1}^{N} \|\nabla F_k(w_k)\|^2 \leq \sigma_2^2 + \sum_{k=1}^{N} \|\nabla_{w_k} J(W)\|^2 \tag{7}$$

where $\nabla_{w_k} J(W)$ is the gradient of J with respect to w_k. Consequently, if every F_k is convex, then

$$\sum_{k=1}^{N} \|\nabla F_k(w_k^*)\|^2 \leq \sigma_2^2. \tag{8}$$

Proof: See [47, Appendix B] (full version of this work). □

For any given value of ρ, the condition $\eta\rho > 2\beta$ in Lemma 1 can be always achieved by tuning $\eta \in \mathbb{R}$. Therefore, the impact of the relationships among the models of clients (or the graph Laplacian structure encoded by ρ) on the convergence of FedU and dFedU can be controlled by η. One can choose a large η if ρ is small and vice versa to satisfy this condition.

Note that in the conventional FL setting, i.e., $w_k = w, \forall k \in \mathcal{N}$, (7) is rewritten as follows:

$$\frac{1}{N}\sum_{k=1}^{N} \|\nabla F_k(w)\|^2 \leq \frac{\sigma_2^2}{N} + \gamma^2 \|\nabla_w J(W)\|^2 \quad \text{with } \gamma = 1$$

which is exactly the assumptions of $(\sigma_2/\sqrt{N}, \gamma)$-bounded gradient dissimilarity in [10] and [22] and the γ-local dissimilarity in [19] with $\sigma_2 = 0$. Here, $\sigma_2 = 0$ and $\gamma = 1$ are for the IID cases, while $\sigma_2 \geq 0$ and $\gamma \geq 1$ for non-IID cases.

From now on, let σ_2 and ρ be defined as in Lemma 1, and $W^{(t)} = [w_1^{(t)}, \ldots, w_N^{(t)}]$ be the collective vector generated by FedU (with client sampling) or dFedU (without client sampling, i.e., $S = N$) at round t. Note that the convergence rate of dFedU is obtained directly from the convergence rate of FedU when $S = N$. In the following theorems, we show that FedU admits linear speedup for strongly convex and sublinear speedup of order $1/2$ for nonconvex objective functions.

Theorem 1 (Convergence in Strongly Convex Cases): Suppose that Assumptions 1–3 hold, and $\eta > (2\beta/\rho)$. Then, there

Algorithm 2 dFedU

client k**'s input**: $\{a_{k\ell}\}$, $\widetilde{\mathcal{N}}_k$, initial $w_k^{(0)}, \forall k \in \mathcal{N}$, local step-size μ, and global step-size $\widetilde{\mu} = \mu R$
for each round $t = 0, \ldots, T-1$ **do**
  on client $k \in \mathcal{N}$ **in parallel do**
    initialize local model $w_{k,0}^{(t)} \leftarrow w_k^{(t)}$
    for $r = 0, \ldots, R-1$ **do**
      compute mini-batch gradient $\nabla \widetilde{F}_k(w_{k,r}^{(t)})$
      $w_{k,r+1}^{(t)} \leftarrow w_{k,r}^{(t)} - \mu \nabla \widetilde{F}_k(w_{k,r}^{(t)})$
    end for
    send $w_{k,R}^{(t)}$ to its neighboring clients in $\widetilde{\mathcal{N}}_k$
  end on client
  on client $k \in \mathcal{N}$ **in parallel do**
    $w_k^{(t+1)} \leftarrow w_{k,R}^{(t)} - \widetilde{\mu}\, \eta \sum_{\ell \in \widetilde{\mathcal{N}}_k} a_{k\ell}(w_{k,R}^{(t)} - w_{\ell,R}^{(t)})$
  end on client
end for

exists $\mu \leq (\widetilde{\mu}_1 / R)$, such that, for any $T \geq (4N/\widetilde{\mu}_1 \alpha S)$

$$\mathbb{E}[J(\widetilde{W}^{(T)}) - J(W^*)] \leq \widetilde{\mathcal{O}}\left(\alpha \Delta^{(0)} e^{-\frac{\widetilde{\mu}_1 \alpha S T}{4N}} + \frac{\sigma_1^2}{(\alpha T)^2 RS} + \frac{\sigma_2^2}{(\alpha T)^2 S} + \frac{\sigma_1^2}{\alpha T RS} + \frac{\sigma_2^2}{\alpha T S}\right) \tag{9}$$

where $\widetilde{\mu}_1 := \min\{(1/q), (2/\eta\rho)\}$, $q = (128\beta^2\eta\rho/\alpha^2) + 12(\beta + \eta\rho) + (96\beta^2/\alpha) + (32p\beta^2/\alpha\eta\rho)$, $p = 2(\beta + \eta\rho) + (8\eta^2\rho^2/\alpha) + (64\beta^2/\alpha) + (12(\beta+\eta\rho)^2/\eta\rho) + 6\eta\rho + (48\beta^2/\eta\rho)$, $\Delta^{(0)} := \|W^{(0)} - W^*\|^2$, $\widetilde{W}^{(T)} := \sum_{t=0}^{T-1}(\theta^{(t)} W^{(t)}/\Theta_T)$, $\Theta_T = \sum_{t=0}^{T-1} \theta^{(t)}$, $\theta^{(t)} = (1 - \mu RS\alpha/(4N))^{-(t+1)}$, and $\widetilde{\mathcal{O}}$ hides both constants and polylogarithmic factors. Consequently, the output of FedU has expected error smaller than ε when

$$T = \widetilde{\mathcal{O}}\left(\frac{1}{\alpha S} + \frac{\sigma_1}{\alpha\sqrt{\varepsilon RS}} + \frac{\sigma_2}{\alpha\sqrt{\varepsilon S}} + \frac{\sigma_1^2}{\alpha RS\varepsilon} + \frac{\sigma_2^2}{\alpha S\varepsilon}\right). \tag{10}$$

Proof: See [47, Appendix D] (full version of this work). □

Theorem 2 (Convergence in Nonconvex Cases): Suppose that Assumptions 1 and 3 hold, and $\eta > (2\beta/\rho)$. Then, there exists $\mu \leq (\widetilde{\mu}_2 / R)$, such that, for any $T > 0$

$$\mathbb{E}\,\|\nabla J(W^{(t^*)})\|^2 \leq \mathcal{O}\left(\frac{\Delta_J}{TS} + \frac{\Delta_J^{\frac{2}{3}} M^{\frac{2}{3}}}{T^{\frac{2}{3}}(RS)^{\frac{1}{3}}} + \frac{\Delta_J^{\frac{1}{2}} M^2}{\sqrt{TRS}}\right) \tag{11}$$

where $\widetilde{\mu}_2 := \min\{(1/\upsilon), (2/\eta\rho)\}$, $\upsilon = 8(8\eta\rho + 3(\beta + \eta\rho) + 12(\beta + \eta\rho) + (8u/\eta\rho))$, and $u = ((\beta + \eta\rho)^2/2) + 2\eta^2\rho^2 + 16\eta\rho\beta^2 + ((6(\beta + \eta\rho)^3)/\eta\rho) + 3\eta\rho(\beta + \eta\rho) + ((24(\beta + \eta\rho)\beta^2)/\eta\rho)$; $\Delta_J := J(W^{(0)}) - J(W^*)$, $M^2 = R\sigma_2^2 + \sigma_1^2$, and t^* uniformly sampled from $\{0, \ldots, T-1\}$. Consequently, the output of FedU has expected error smaller than ε when

$$T = \mathcal{O}\left(\frac{1}{S\varepsilon} + \frac{\sigma_1}{\varepsilon^{\frac{3}{2}}\sqrt{RS}} + \frac{\sigma_2}{\varepsilon^{\frac{3}{2}}\sqrt{S}} + \frac{\sigma_1^2}{\varepsilon^2 RS} + \frac{\sigma_2^2}{\varepsilon^2 S}\right). \tag{12}$$

Proof: See [47, Appendix E] (full version of this work). □

For illustrative purposes, we compare our rates with those of FL and personalized FL algorithms in IID cases (i.e., $\sigma_2 = 0$ and $\gamma = 1$). The strongly convex rate of FedU becomes $(\sigma_1^2/\alpha RS\varepsilon) + (1/\alpha S)$, which matches the lower bound for the identical case [48], compared with the latest $(\sigma_1^2/\alpha RS\varepsilon) + (1/\alpha)$ by SCAFFOLD [10] and $(\sigma_1^2/\alpha RS\varepsilon) + (\delta/\alpha)$ by LSGD-PFL [45] with $\delta \geq 0$. Our rate improvement comes from the advantage of additional information about the structure of the models of clients that is captured by Laplacian regularization. Also, when no variance ($\sigma_1^2 = 0$) and no client sampling, the nonconvex rate of FedU is $(\sigma_2^2/\varepsilon^2 S) + (\sigma_2/\varepsilon^{3/2}) + (1/\varepsilon)$, which is tighter (without γ) than the rate of SCAFFOLD, and less dependent on σ_2 than that of [49].

VI. Experiments

In this section, we evaluate the performance of FedU when the data are heterogeneous and non-IID in both strongly convex and nonconvex settings. We show the advances of FedU with Laplacian regularization in federated multitask and personalized settings by comparing FedU with cutting-edge learning algorithms, including MOCHA, pFedMe, Per-FedAvg, FedProx [19], SCAFFOLD [10], AFL [50], and the vanilla FedAvg. The experimental results show that FedU achieves appreciable performance improvement over others in terms of test accuracy.

A. Experimental Settings

We consider classification problems using real datasets generated in federated settings, including Human Activity Recognition, Vehicle Sensor, MNIST, and CIFAR-10. noitemsep,nolistsep

1) *Human Activity Recognition:* The set of data gathered from accelerometers and gyroscopes of cell phones from 30 individuals performing six different activities, including lying down, standing, walking, sitting, walking upstairs, and walking downstairs [51]. Each individual is considered as a task (client) classifying six different activities.
2) *Vehicle Sensor:* Data are collected from a distributed wireless sensor network of 23 sensors, including acoustic (microphone), seismic (geophone), and infrared (polarized IR sensor) [52]. It aims to classify types of moving vehicles. We consider each sensor as a separate task (client) performing the binary classification to predict two vehicle types: assault amphibian vehicle (AAV) and dragon wagon (DW).
3) *MNIST:* A handwritten digit dataset [53] includes ten labels and 70 000 instances. The whole dataset is distributed to $N = 100$ clients. Each client has a different local data size and consists of two over ten labels.
4) *CIFAR-10:* An object recognition dataset [54] includes 60 000 color images belonging to ten classes. We partition the dataset to $N = 20$ clients and three labels per client.

In practical FL networks, some clients have significantly limited data sizes and need collaborative learning with others. For each dataset, we, hence, downsample 80% data belonged

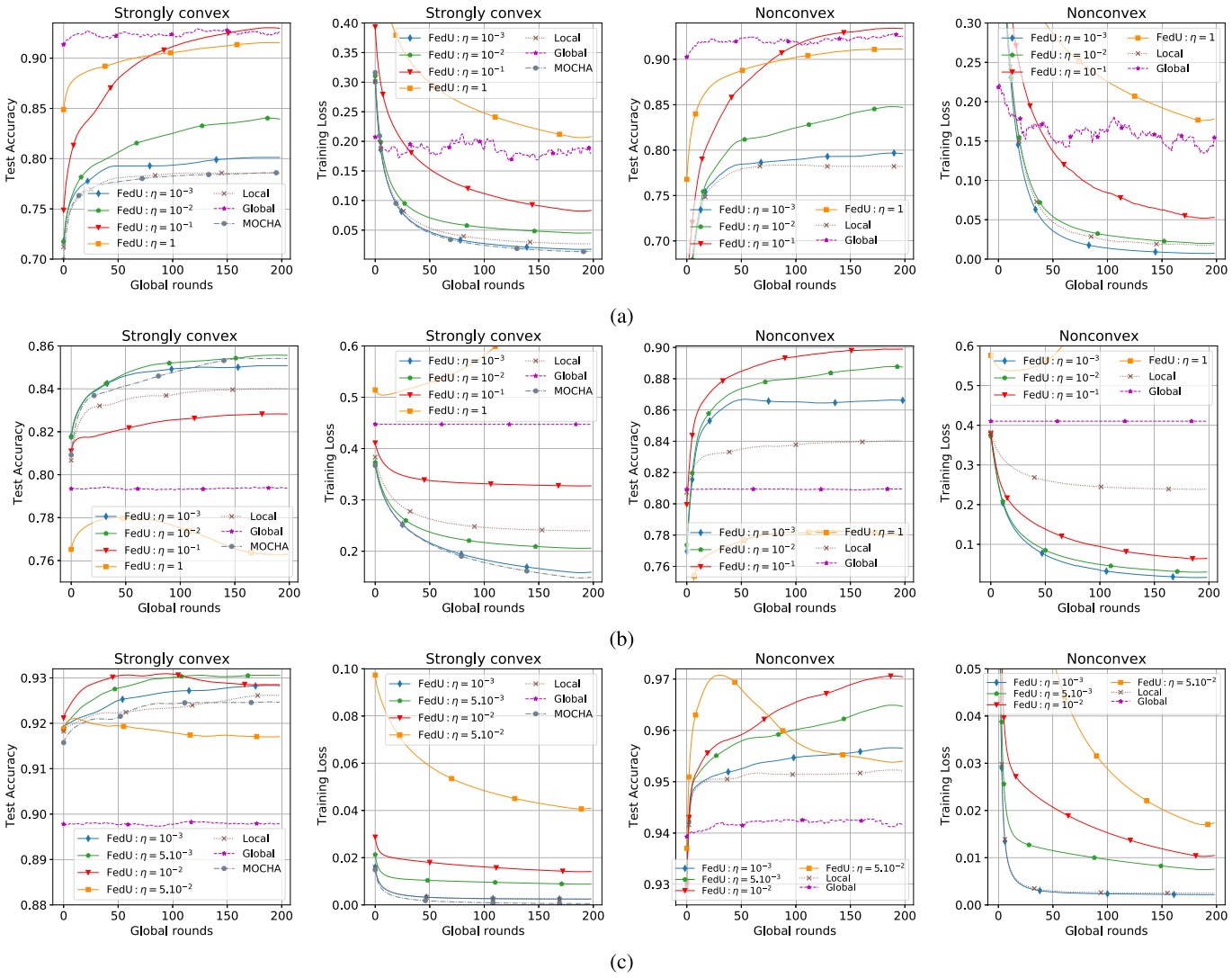

Fig. 3. Performance comparison among MOCHA, local model, global model, and FedU with the various sets of η in both strongly convex and nonconvex settings. (a) Human Activity. (b) Vehicle Sensor. (c) MNIST.

to a half of the total clients to observe behavior of all algorithms. We provide all details about datasets and results without downsampling in [47, Appendix F]. All datasets are split randomly with 75% and 25% for training and testing, respectively.

We use a multinomial logistic regression (MLR) model with cross-entropy loss functions and L_2-regularization term as the strongly convex model for Human Activity Recognition, Vehicle Sensor, and MNIST. For nonconvex setting, we use a simple DNN with one hidden layer, an ReLU activation function, and a softmax layer at the end of the network for Human Activity and Vehicle Sensor datasets. The size of hidden layer is 100 for Human Activity and 20 for Vehicle Sensor. In the case of MNIST, we use DNN with two hidden layers, and both layers have the same size of 100. For CIFAR-10, we follow the CNN structure of [1].

The structural dependence matrix $\mathbf{\Omega}$ of MOCHA is chosen as $\mathbf{\Omega} = (\mathbf{I}_{N\times N} - (1/N)\mathbf{1}\mathbf{1}^T)^2$ following the settings of [16] and [24], where $\mathbf{I}_{N\times N}$ is the identity matrix with size $N \times N$, and $\mathbf{1}$ is a vector of all ones size N. Here, $\mathbf{\Omega}$ is exactly the Laplacian matrix L in problem (1) when all the weights $a_{k\ell} = 1, \forall k, \ell$. As both FedU and dFedU have the same performance when there is no client sampling, in our experiments, we only evaluate the performance of FedU. When comparing FedU with other algorithms, we conduct fivefold cross validation to figure out the combination of hyperparameters allowing each algorithm to achieve the highest test accuracy. All experiments are implemented using PyTorch [55] version 1.6. We follow the implementations of [14] for pFedMe, FedAvg, and Per-FedAvg, and [24] for MOCHA. All experiments are run on **NVIDIA Tesla T4** GPU. All code and data are published at https://github.com/dual-grp/FedU_FMTL. The accuracy is reported with mean and standard deviation over ten runs.

B. Performance of FedU in FMTL

We first show the benefits of FedU in FMTL setting by comparing FedU with local model (training one separate model per client), global model (training one single model on centralized data), and MOCHA, the conventional

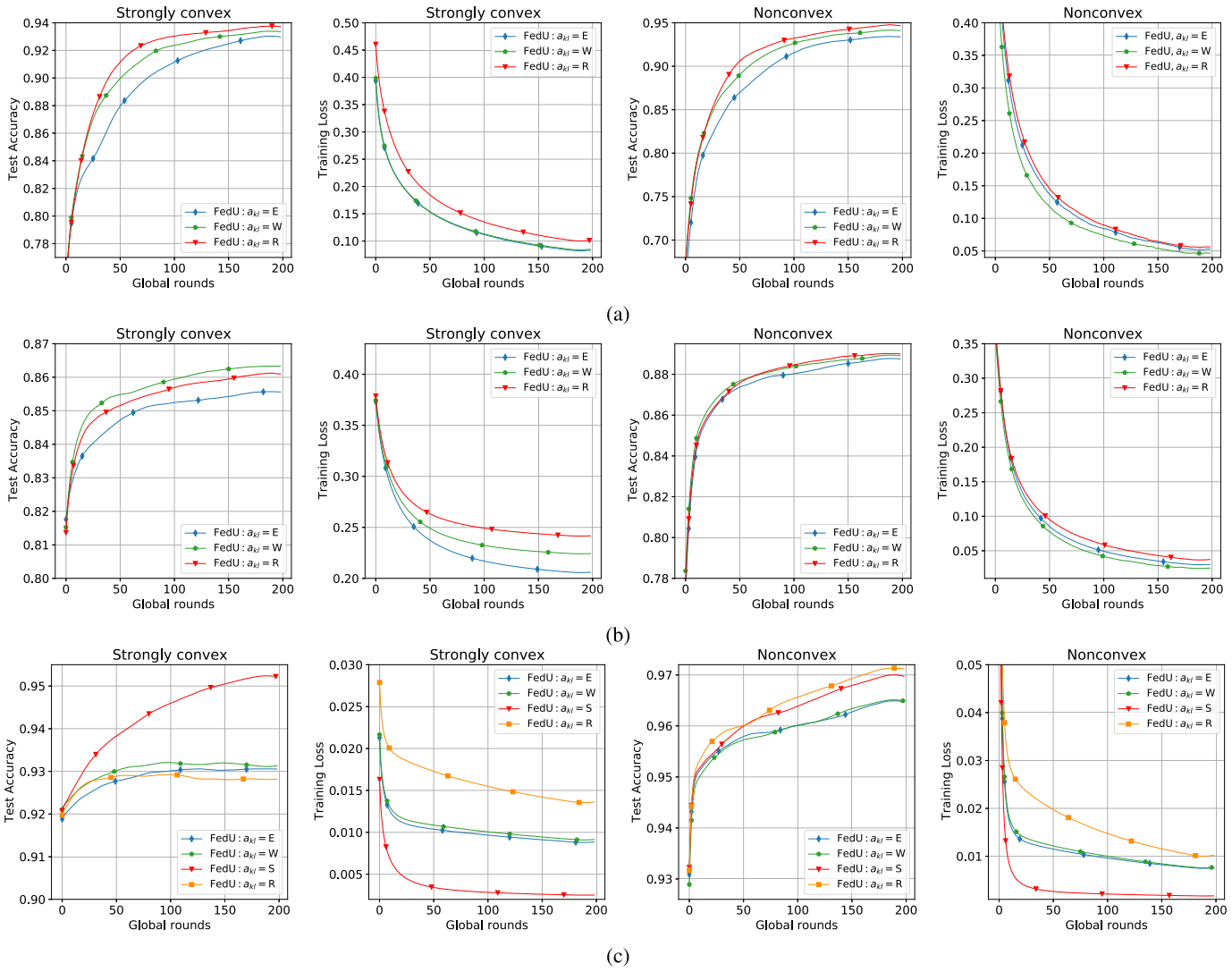

Fig. 4. Effects of graph information $\{a_{kl}\}$ on the convergence of FedU in both convex and nonconvex settings. (a) Human Activity. (b) Vehicle Sensor. (c) MNIST.

FMTL algorithm [16]. Note that the performance results of the FMTL algorithm in [29] and MOCHA are reported similar. We evaluate FedU on a wide range of values of $\eta \in \{5.10^{-3}, 10^{-3}, 5.10^{-2}, 10^{-2}, 10^{-1}, 1\}$ and compare with others using their best fine-tuned parameters. In FMTL, each client represents a separate task. All clients have the same weight connection $\{a_{kl}\}$ with others and no client sampling in order to make fair comparisons with local, global models, and MOCHA. We also provide details on how to choose the different values of $\{a_{kl}\}$ in Section VI-C. We only report the convex setting for MOCHA according to its assumption as stated in Section III-A [16].

The results in Fig. 3 show that FedU achieves the highest performance, followed by MOCHA, local model, and global model. While the local model at individual client learns only its own data without any contribution from the model of other clients, the global model only does a single task that is not well generalized on highly non-IID data. We also recognize that the local model suffers overfitting when the data size at clients is small. By contrast, MOCHA and FedU have the ability to learn models for multiple related tasks simultaneously and capture relationships among clients. Especially, in the case of FedU, using Laplacian regularization allows utilizing additional information about the structures of clients' models to increase the learning performance, and the contribution from clients having the large data size to those having smaller ones becomes more significant.

Observing different values of η, we found that the larger the value of η is, the more the coordination from other clients are; then, FedU performs better when η is increased. However, when η reaches a certain threshold, it slows down the convergence of FedU, for example, $\eta = 5.10^{-2}$ in Fig. 3. η then should be chosen carefully depends on the dataset.

C. Effect of the Graph Information $\{a_{kl}\}$

For the above experiments, we assume that all relationships among a client and its neighbors are equal. However, in practice, the connection weights may have different values, and they need to be known in advance. We then evaluate the

TABLE I

PERFORMANCE COMPARISON OF CENTRALIZED SETTING ($R = 5$, $S = 0.1\ N$, $B = 20$, AND $T = 200$). THERE IS NO CONVEX MODEL FOR CIFAR-10; WE THEN ONLY REPORT THE NONCONVEX CASE

Dataset	Algorithm	Test Accurancy Convex	Non Convex
CIFAR-10	FedU		**75.41** $\pm$ 0.29
	pFedMe		74.10 $\pm$ 0.89
	Per-FedAvg		64.70 $\pm$ 1.91
	FedAvg		34.48 $\pm$ 5.34
	FedProx		42.31 $\pm$ 4.21
	SCAFFOLD		45.12 $\pm$ 3.38
	AFL		49.07 $\pm$ 3.35
MNIST	FedU	**96.95** $\pm$ 0.11	97.81 $\pm$ 0.01
	MOCHA	96.18 $\pm$ 0.09	
	pFedMe	93.73 $\pm$ 0.40	**98.64** $\pm$ 0.17
	Per-FedAvg	90.33 $\pm$ 0.84	96.38 $\pm$ 0.40
	FedAvg	87.75 $\pm$ 1.31	91.48 $\pm$ 1.05
	FedProx	88.70 $\pm$ 1.18	91.60 $\pm$ 0.23
	SCAFFOLD	89.45 $\pm$ 0.37	92.15 $\pm$ 0.43
	AFL	89.79 $\pm$ 1.23	92.01 $\pm$ 1.21
Vehicle Sensor	FedU	**88.47** $\pm$ 0.21	**91.79** $\pm$ 0.31
	MOCHA	87.31 $\pm$ 0.23	
	pFedMe	81.38 $\pm$ 0.41	90.62 $\pm$ 0.41
	Per-FedAvg	81.07 $\pm$ 0.71	86.92 $\pm$ 1.3
	FedAvg	79.84 $\pm$ 0.91	84.04 $\pm$ 2.69
	FedProx	82.06 $\pm$ 0.91	87.65 $\pm$ 2.34
	SCAFFOLD	81.97 $\pm$ 0.91	88.48 $\pm$ 0.34
	AFL	82.25 $\pm$ 0.91	87.88 $\pm$ 1.08
Human Activity	FedU	**95.76** $\pm$ 0.46	**95.86** $\pm$ 0.36
	MOCHA	92.33 $\pm$ 0.67	
	pFedMe	95.41 $\pm$ 0.38	95.72 $\pm$ 0.32
	Per-FedAvg	94.78 $\pm$ 0.37	94.80 $\pm$ 0.60
	FedAvg	93.41 $\pm$ 0.95	93.74 $\pm$ 1.01
	FedPro	93.69 $\pm$ 0.84	94.65 $\pm$ 0.72
	SCAFFOLD	93.61 $\pm$ 0.37	94.78 $\pm$ 0.85
	AFL	93.92 $\pm$ 0.34	94.42 $\pm$ 0.34

effect of graph information shown in Fig. 4 by normalizing the values of $\{a_{kl}\}$ in the range of $[0, 1]$ and simulate four different scenarios of $\{a_{kl}\}$ as follows.

1) *Random (R):* All values of $\{a_{kl}\}$ are generated randomly $\{a_{kl}\} \sim \mathcal{N}(0, 1)$.
2) *Equal (E):* When all clients have the same value for $\{a_{kl}\}$, we can choose any value of $\{a_{kl}\}$ in the range of $[0, 1]$. However, there will be one value of $\eta * \{a_{kl}\}$ allowing FedU to achieve the highest accuracy. So, whenever $\{a_{kl}\}$ is large, we can choose a small η, and vice versa. In this experiment, we fix $\{a_{kl}\} = 0.5$ and adjust η accordingly.
3) *Weighted (W):* As there are various clients having significantly small data sizes, we set $\{a_{kl}\} = 0$ on the connection between these clients. We then set $\{a_{kl}\} = 0.5$ on the connection among clients having small data sizes and those having large data sizes, and $\{a_{kl}\} = 1$ for all other connections.
4) *Similar (S):* This scenario is only for MNIST. When distributing data to all clients, each client has two labels over ten. Hence, clients may share only one, two similar labels, or none of them. We set $\{a_{kl}\} = 0$, $\{a_{kl}\} = 0.5$, and $\{a_{kl}\} = 1$ for the connections among clients having no similar label, one similar label, and two similar labels, respectively.

In most of the cases, the performance of FedU with random $\{a_{kl}\}$ is better than that with equal $\{a_{kl}\}$. When the values of $\{a_{kl}\}$ are weighted, FedU performs better than when all $\{a_{kl}\}$ are equal. Especially for MNIST, when the values of $\{a_{kl}\}$ are weighted based on the similarity of clients, FedU achieves the highest performance compared with other scenarios. Therefore, given knowing the relationship between client's data distribution, for example, in a weather forecasts application, clients in the same geographical location may have similar or close weather data, and we can set higher values of weight connection for those clients than clients are in different locations to takes advantages of FedU.

D. Comparison With Personalized FL Algorithms

Finally, we compare FedU with the conventional FL algorithms FedAvg, FedProx, SCAFFOLD, AFL, and MOCHA, and with the state-of-the-art personalized FL algorithms pFedMe and Per-FedAvg. The results are shown in Table I. We fix the subset of clients $S = 0.1N$ and perform the comparison on all four real datasets. Overall, FedU almost maintains the top performance in all scenarios.

VII. CONCLUSION

This work has formulated an FMTL problem using Laplacian regularization to capture the relationships among the models of clients. The formulated problem has been proven to be used for traditional FL and personalized FL. We have also proposed both communication-centralized and decentralized algorithms to solve the formulated problem with guaranteed convergence to the optimal solution. Theoretical results show that our algorithms FedU and dFedU achieve the state-of-the art convergence rates. Experimental results with real datasets in both convex and nonconvex objectives demonstrate that the proposed algorithms outperform the conventional MOCHA in FMTL settings, the vanilla FedAvg in FL settings, and pFedMe and Per-FedAvg in personalized FL settings.

ACKNOWLEDGMENT

The research of Minh N. Dao benefited from the support of the FMJH Program Gaspard Monge for optimization and operations research and their interactions with data science.

REFERENCES

[1] 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.*, Apr. 2017, pp. 1273–1282.
[2] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1, pp. 1–76, 2021.
[3] 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.
[4] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3710–3722, Aug. 2021.
[5] N. Rieke et al., "The future of digital health with federated learning," *NPJ Digit. Med.*, vol. 3, no. 1, pp. 1–7, 2020.
[6] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *J. Healthcare Informat. Res.*, vol. 5, no. 1, pp. 1–19, Mar. 2021.

[7] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," *Int. J. Med. Inform.*, vol. 112, pp. 59–67, Jan. 2018.
[8] J. C. Jiang, B. Kantarci, S. Oktug, and T. Soyata, "Federated learning in smart city sensing: Challenges and opportunities," *Sensors*, vol. 20, no. 21, p. 6230, Oct. 2020.
[9] L. Ahmed, K. Ahmad, N. Said, B. Qolomany, J. Qadir, and A. Al-Fuqaha, "Active learning based federated learning for waste and natural disaster image classification," *IEEE Access*, vol. 8, pp. 208518–208531, 2020.
[10] S. P. Karimireddy et al., "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, vol. 119, 2020, pp. 1–12.
[11] F. Haddadpour and M. Mahdavi, "On the convergence of local descent methods in federated learning," 2019, *arXiv:1910.14425*.
[12] D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," 2019, *arXiv:1910.03581*.
[13] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," 2020, *arXiv:2003.13461*.
[14] C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized federated learning with Moreau envelopes," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 1–12.
[15] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–12.
[16] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1–11.
[17] A. Kumar and H. Daumé, "Learning task grouping and overlap in multi-task learning," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1–15.
[18] Y. Zhang and D.-Y. Yeung, "A convex formulation for learning task relationships in multi-task learning," in *Proc. 26th Conf. Uncertainty Artif. Intell.*, 2010, pp. 733–742.
[19] T. Li et al., "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, 2020, pp. 429–450.
[20] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, *arXiv:1806.00582*.
[21] 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.*, Apr. 2020, pp. 1–26.
[22] A. Khaled, K. Mishchenko, and P. Richtarik, "Tighter theory for local SGD on identical and heterogeneous data," in *Proc. Int. Conf. Artif. Intell. Statist.*, vol. 108, Aug. 2020, pp. 26–28.
[23] F. Hanzely and P. Richtárik, "Federated learning of a mixture of global and local models," 2020, *arXiv:2002.05516*.
[24] P. P. Liang et al., "Think locally, act globally: Federated learning with local and global representations," 2020, *arXiv:2001.01523*.
[25] 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. 1–12.
[26] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," 2019, *arXiv:1909.12488*.
[27] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018, *arXiv:1803.02999*.
[28] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," 2019, *arXiv:1912.00818*.
[29] Y. SarcheshmehPour, Y. Tian, L. Zhang, and A. Jung, "Networked federated learning," 2021, *arXiv:2105.12769*.
[30] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *Proc. 38th Int. Conf. Mach. Learn.*, Jul. 2021, pp. 1–12.
[31] J. Shen, X. Zhen, M. Worring, and L. Shao, "Variational multi-task learning with Gumbel-Softmax priors," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 1–12.
[32] R. Li, F. Ma, W. Jiang, and J. Gao, "Online federated multitask learning," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 215–220.
[33] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Mach. Learn.*, vol. 73, no. 3, pp. 243–272, 2008.
[34] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *J. Mach. Learn. Res.*, vol. 6, pp. 1817–1853, Nov. 2005.
[35] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, Jul. 1997.
[36] A. Jung and Y. SarcheshmehPour, "Local graph clustering with network lasso," *IEEE Signal Process. Lett.*, vol. 28, pp. 106–110, 2021.
[37] A. Jung and N. Tran, "Localized linear regression in networked data," *IEEE Signal Process. Lett.*, vol. 26, no. 7, pp. 1090–1094, Jul. 2019.
[38] D. Hallac, J. Leskovec, and S. Boyd, "Network lasso: Clustering and optimization in large graphs," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 387–396.
[39] F. Hanzely, S. Hanzely, S. Horváth, and P. Richtarik, "Lower bounds and optimal algorithms for personalized federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–12.
[40] P. Vanhaesebrouck, A. Bellet, and M. Tommasi, "Decentralized collaborative learning of personalized models over networks," in *Proc. 20th Int. Conf. Artif. Intell. Statist. (AISTATS)*, Fort Lauderdale, FL, USA, Apr. 2017, pp. 509–517.
[41] R. Nassif, S. Vlaski, C. Richard, and A. H. Sayed, "Learning over multitask graphs—Part I: Stability analysis," *IEEE Open J. Signal Process.*, vol. 1, pp. 28–45, 2020.
[42] R. Nassif, S. Vlaski, C. Richard, and A. H. Sayed, "Learning over multitask graphs—Part II: Performance analysis," *IEEE Open J. Signal Process.*, vol. 1, pp. 46–63, 2020.
[43] J. Tuck, S. Barratt, and S. Boyd, "A distributed method for fitting Laplacian regularized stratified models," 2019, *arXiv:1904.12017*.
[44] J. Tuck and S. Boyd, "Eigen-stratified models," *Optim. Eng.*, vol. 23, pp. 397–419, Jan. 2021.
[45] F. Hanzely, B. Zhao, and M. Kolar, "Personalized federated learning: A unified framework and universal optimization techniques," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–12.
[46] Y. Nesterov, *Lectures on Convex Optimization*, vol. 137. Berlin, Germany: Springer, 2018.
[47] C. T. Dinh, T. T. Vu, N. H. Tran, M. N. Dao, and H. Zhang, "A new look and convergence rate of federated multi-task learning with Laplacian regularization," 2021, *arXiv:2102.07148*.
[48] B. E. Woodworth, J. Wang, A. Smith, B. McMahan, and N. Srebro, "Graph Oracle models, lower bounds, and gaps for parallel stochastic optimization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–15.
[49] 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.*, Jul. 2019, vol. 33, no. 1, pp. 5693–5700.
[50] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," 2019, *arXiv:1902.00146*.
[51] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proc. 21st Int. Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn.*, 2013, pp. 437–442.
[52] M. F. Duarte and Y. H. Hu, "Vehicle classification in distributed sensor networks," *J. Parallel Distrib. Comput.*, vol. 64, no. 7, pp. 826–838, Jul. 2004.
[53] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[54] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
[55] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30. Vancouver, BC, Canada, 2019, pp. 1–12.

Canh T. Dinh received the B.E. degree in electronics and telecommunication from the Ha Noi University of Science and Technology, Ha Noi City, Vietnam, in 2015, and the Master of Data Science degree from Université Grenoble Alpes, Grenoble, France, in 2019. He is currently pursuing the Ph.D. degree in computer science with The University of Sydney, Sydney, NSW, Australia.

His research interests include federated learning and privacy machine learning.

Tung T. Vu (Member, IEEE) received the Ph.D. degree in wireless communications from The University of Newcastle, Callaghan, NSW, Australia, in 2021.

In 2019, he visited the Broadband Communications Research Laboratory, McGill University, Montreal, QC, Canada. From 2021 to 2022, he was a Research Fellow with Queen's University Belfast, Belfast, U.K. He is currently a Post-Doctoral Researcher with the Department of Electrical Engineering (ISY), Linköping University, Linköping, Sweden. His research interests include optimization, information theories, and machine learning applications for 5G-and-beyond wireless networks, especially with massive MIMO, cell-free massive MIMO, federated learning, full-duplex communications, physical layer security, and low-earth orbit satellite communications.

Dr. Vu was an IEEE WIRELESS COMMUNICATIONS LETTERS Exemplary Reviewer in 2020 and 2021, and an IEEE TRANSACTIONS ON COMMUNICATIONS Exemplary Reviewer in 2021. He is serving as an Editor for *Elsevier Physical Communication (PHYCOM)*.

Nguyen H. Tran (Senior Member, IEEE) received the B.S. degree in electrical and computer engineering from the Ho Chi Minh City (HCMC) University of Technology, Ho Chi Minh City, Vietnam, in 2005, and the Ph.D. degree in electrical and computer engineering from Kyung Hee University (KHU), Seoul, South Korea, in 2011.

He was an Assistant Professor with the Department of Computer Science and Engineering, KHU, from 2012 to 2017. Since 2018, he has been with the School of Computer Science, The University of Sydney, Sydney, NSW, Australia, where he is currently a Senior Lecturer. His research interests include distributed computing, machine learning, and networking.

Dr. Tran received the best KHU thesis award in engineering in 2011 and several best paper awards, including the IEEE ICC 2016 and ACM MSWiM 2019. He receives the Korea NRF Funding for Basic Science and Research 2016–2023 and ARC Discovery Project 2020–2023. He was an Editor of the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING from 2016 to 2020, and an Associate Editor of the IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS 2020 in the area of distributed machine learning/federated learning.

Minh N. Dao received the B.Sc. (Hons.) and M.Sc. degrees in mathematics from the Hanoi National University of Education, Hanoi, Vietnam, in 2004 and 2006, respectively, and the Ph.D. degree in applied mathematics from the University of Toulouse, Toulouse, France, in 2014.

He was a Lecturer with the Hanoi National University of Education, a Post-Doctoral Fellow with The University of British Columbia, Vancouver, BC, Canada, a Research Associate with The University of Newcastle, Callaghan, NSW, Australia, and the University of New South Wales, Sydney, NSW, and a Lecturer with Federation University Australia, Ballarat, VIC, Australia. He is currently a Senior Lecturer with RMIT University, Melbourne, VIC. His research interests include mathematical optimization, convex and variational analysis, control theory, signal processing, and machine learning.

Dr. Dao received the Annual Best Paper Award from the *Journal of Global Optimization* in 2017.

Hongyu Zhang (Senior Member, IEEE) received the Ph.D. degree from the National University of Singapore, Singapore, in 2003.

He was a Lead Researcher with Microsoft Research Asia, Beijing, China, and an Associate Professor with Tsinghua University, Beijing. He is currently an Associate Professor with The University of Newcastle, Callaghan, NSW, Australia. He is interested in intelligent software engineering, AIOps, software maintenance, and software quality. He has authored or coauthored more than 180 research papers in leading international journals and conferences.

Dr. Zhang is a Distinguished Member of ACM and CCF. He is a fellow of Engineers Australia (FIEAust). He received five ACM distinguished paper awards and several best paper awards. He has also served as a program committee member/track chair for many international conferences. He is an Associate Editor of the *ACM Computing Surveys* and *Automated Software Engineering*.