Wireless Federated Learning With Hybrid Local and Centralized Training: A Latency Minimization Design

Ning Huang, Minghui Dai, Yuan Wu, *Senior Member, IEEE*, Tony Q. S. Quek, *Fellow, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Wireless federated learning (FL) is a collaborative machine learning (ML) framework in which wireless client-devices independently train their ML models and send the locally trained models to the FL server for aggregation. In this paper, we consider the coexistence of privacy-sensitive client-devices and privacy-insensitive yet computing-resource constrained client-devices, and propose an FL framework with a hybrid centralized training and local training. Specifically, the privacy-sensitive client-devices perform local ML model training and send their local models to the FL server. Each privacy-insensitive client-device can have two options, i.e., (i) conducting a local training and then sending its local model to the FL server, and (ii) directly sending its local data to the FL server for the centralized training. The FL server, after collecting the data from the privacy-insensitive client-devices (which choose to upload the local data), conducts a centralized training with the received datasets. The global model is then generated by aggregating (i) the local models uploaded by the client-devices and (ii) the model trained by the FL server centrally. Focusing on this hybrid FL framework, we firstly analyze its convergence feature with respect to the client-devices' selections of local training or centralized training. We then formulate a joint optimization of client-devices' selections of the local training or centralized training, the FL training configuration (i.e., the number of the local iterations and the number of the global iterations), and the bandwidth allocations to the client-devices, with the objective of minimizing the overall latency for reaching the FL convergence. Despite the non-convexity of the joint optimization problem, we identify its layered structure and propose an efficient algorithm to solve it. Numerical results demonstrate the advantage of our proposed FL framework with the hybrid local and centralized training as well as our proposed algorithm, in comparison with several benchmark FL schemes and algorithms.**

***Index Terms*—Federated learning, hybrid local and centralized training, resource allocation.**

I. Introduction

With the explosive growth in the number of Internet of Things (IoT) devices, a tremendous amount of data is generated by the IoT devices in the wireless access networks. Federated learning (FL), which allows numerous client-devices (CDs) to cooperatively train a common machine learning (ML) model without revealing their private data, has attracted lots of interests in a variety of services and applications [1], [2]. In FL, each CD can perform a local model training based on its local data and then send the locally trained model to the FL server for the global model aggregation. Compared to the traditional centralized training, FL not only preserves the privacy of local data but also reduces the communication resource usage for transmitting data to the FL server.

The advantages of FL have raised lots of applications, in particular, the wireless access networks. Taking into account the limited radio resource (e.g., the access bandwidth and battery capacity of wireless terminals), there have been many studies investigating the joint optimization of radio resource allocation and FL convergence. Many existing studies focus on the balance of the resource utilization and the FL convergence performance, in which the CDs are presumed as privacy-sensitive and can only perform local model training [3], [4], [5]. However, in practice, there exist the privacy-insensitive CDs but with limited computing-resources, which thus prefer to send their local data to the FL server for a centralized training.

Based on the above consideration, different from the existing studies, we investigate a novel FL scheme with the hybrid centralized and local training. Specifically, we consider that there exist two types of CDs, i.e., the privacy-sensitive CDs and the privacy-insensitive CDs. The privacy-sensitive CDs, due to the stringent privacy concern on their local data, have to conduct the local model training individually and then send their

Manuscript received 10 June 2022; revised 21 September 2022; accepted 9 November 2022. Date of publication 21 November 2022; date of current version 17 February 2023. This work was supported in part by National Natural Science Foundation of China under Grant 62072490, in part by FDCT-MOST Joint Project under Grant 0066/2019/AMJ, in part by Science and Technology Development Fund of Macau SAR under Grant 0162/2019/A3, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515011287, in part by Research Grant of University of Macau under Grant MYRG2020-00107-IOTSC, in part by the National Research Foundation, Singapore, in part by Infocomm Media Development Authority under its Future Communications Research and Development Programme, in part by MOE ARF Tier 2 under Grant T2EP20120-0006, and in part by the Natural Sciences and Engineering Research Council of Canada. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Marco Levorato. *(Corresponding author: Yuan Wu.)*

Ning Huang and Minghui Dai are with the State Key Lab of Internet of Things for Smart City, Macau, China, and also with the Department of Computer and Information Science, University of Macau, Macau, China (e-mail: yc07427@umac.mo; minghuidai@um.edu.mo).

Yuan Wu is with the State Key Laboratory of Internet of Things for Smart City and the Department of Computer and Information Science, University of Macau, Macau, China, and also with the Zhuhai UM Science and Technology Research Institute, Zhuhai 519031, China (e-mail: yuanwu@um.edu.mo).

Tony Q. S. Quek is with the Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore 487372, and also with the National Cheng Kung University, Taiwan (e-mail: tonyquek@sutd.edu.sg).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/JSTSP.2022.3223498

trained models to the FL server. The privacy-insensitive CDs, however, do not have any privacy concern. Taking into account its local computing-resource, each privacy-insensitive CD has two options, i.e., (i) conducting a local training and then sending its locally trained model to the FL server, and (ii) directly sending its local data to the FL server for the centralized training. After collecting the data from those privacy-insensitive CDs (which choose to upload their local data), the FL server then conducts a centralized training with the received dataset. After that, the FL server will aggregate the centralized trained model with its received local models from the CDs to update the global model. This hybrid centralized and local training is expected to achieve a higher convergence rate in comparison with the conventional FL without the centralized training, which is regarded as a key advantage and is validated by both theoretic analysis and simulation in this work. There exist two key challenging issues in this considered FL with the hybrid centralized and local training as follows.

- *Coordination between local and centralized training:* To minimize the convergence latency of FL, we need to properly coordinate the following two aspects: (i) the local model training at the CD that consumes the local training latency, and (ii) the data transmission and the centralized training latency at the FL server.
- *Convergence feature of the FL with hybrid training:* We need to quantify the FL convergence feature with respect to the CDs' selections of local training or centralized training, as well as the corresponding resource allocation.

Motivated by the above challenges, we propose a joint optimization of the CDs' selections of local training or centralized training and communication resource allocation. Our detailed contributions in this paper can be summarized as follows.

- *Hybrid training framework:* We propose a novel FL framework with the hybrid local and centralized training in which the privacy-insensitive CD can flexibly choose to either conduct local training or send its local data to the FL server for centralized training. We characterize the connection between the latency and the selection of local training and centralized training as well as resource allocation, and analytically derive the convergence feature for the proposed FL framework.
- *Latency minimization formulation:* We formulate a joint optimization of the CD's selection of local training and centralized training, the FL configuration (i.e., the numbers of local iterations and global iterations) and the bandwidth allocations to the CDs, with the objective of minimizing the overall FL latency.
- *Algorithmic design:* To address the non-convexity of our formulated joint optimization problem, we identify its layered structure and equivalently decompose it into a top problem for optimizing the selection of local training and centralized training, a middle layer problem for optimizing the FL configuration (i.e., the numbers of local iterations and global iterations) and a bottom problem for optimizing the bandwidth allocation. With this layered structure, we propose an efficient algorithm to find the optimal solution of the formulated optimization problem.
- *Numerical validation:* We provide extensive numerical results to validate the effectiveness of our proposed algorithms. The derived convergence feature of our proposed FL framework is validated with the real dataset. We demonstrate the performance advantage of our proposed algorithm in comparison with two benchmark algorithms. We then demonstrate the performance advantage of our proposed FL scheme with the hybrid training.

The remainder of this paper is organized as follows. We review the related studies in Section II. We analyze the convergence feature of the proposed FL with the hybrid training and formulate a joint optimization problem in Section III. We propose an efficient algorithm for solving the formulated problem in Section V. The numerical results are provided in Section VI. We finally conclude this paper in Section VII and discuss the future directions.

II. Related Studies

As a distributed learning framework, FL is expected to enable various learning based services, which has attracted many research efforts [6], [7], [8]. We review the studies related to our paper as follows.

- *FL with single edge-server or multiple edge-servers:* Many research efforts focus on the federated edge learning which aggregates the model at the edge-server, and the latency can be significantly reduced compared to the conventional cloud-centralized FL network. In [9], the authors proposed a communication-efficient asynchronous FL where part of the clients' local models were selected (according to their arrival order) to be aggregated at the FL server. In [10], the authors proposed a novel client selection scheme for FL, based on the learning quality of participants. In [11], Xiao et al. aimed to minimize the cost in the worst case of FL by selecting the proper vehicles and optimizing the corresponding resource allocation. In [12], Chen et al. aimed to minimize the FL loss function by jointly optimizing the user selection and the wireless resource allocation. In [13], the authors proposed a simultaneous wireless information and power transfer aided FL, in which one FL server simultaneously broadcasts the global model and provides wireless power transfer to wireless devices. Since the coverage of a single edge-server is limited, the authors in [14] considered FL with multiple edge-servers and accelerated the training by utilizing the clients located in the overlapping areas among different edge-servers.
- *Hierarchical FL for edge cloud cooperated network:* In the hierarchical FL network, additional edge-servers are deployed as the intermediate helpers between clients and the FL server. Each intermediate helper first aggregates the local models received from nearby clients and then uploads its aggregated model to the remote server. In [15], the authors proposed a hierarchical game framework to study the dynamics of edge association and resource allocation in self-organizing hierarchical FL networks. In [16], Xu et al. provided the upper bound of the average global gradient deviation and jointly optimized the edge aggregation interval and resource allocation in the hierarchical FL. In [17], Luo et al. proposed a novel hierarchical FL framework where the model aggregation can be partially migrated from the cloud to edge-servers. The authors in [18] studied a hierarchical FL framework where only a portion of helpers' aggregated model parameters can be uploaded to the cloud server. In [19], the authors proposed a hierarchical two-level incentive mechanism for the resource allocation in hierarchical FL networks.
- *Fully decentralized FL (or serverless FL):* Fully decentralized FL enables the clients to exchange the locally

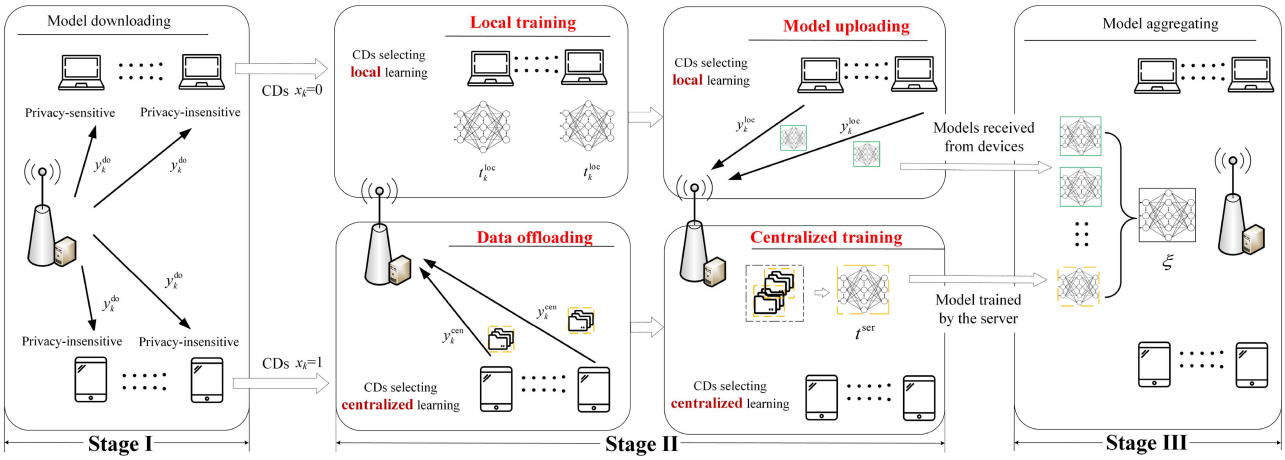

Fig. 1. FL with the hybrid local learning and centralized training.

trained models with the clients nearby, without relaying on any server. The authors in [20] analyzed the convergence behavior of decentralized stochastic gradient descent in FL. In [21], Xiao et al. developed an unmanned aerial vehicle assisted fully decentralized FL framework. The authors in [22] studied the decentralized FL based on the consensus mechanism in the blockchain. In [23], Xing et al. investigated device-to-device communication assisted FL by analyzing the performance of decentralized stochastic gradient descent.

However, it is still an open research problem for the resource allocation in the FL with a hybrid local and centralized training framework, in which some FL clients are allowed to upload their raw data to the FL server for centralized training.

III. System Model and Problem Formulation

We consider the frequency-division multiple access (FDMA) based FL system including one base station (BS) with one single antenna and K CDs with one single antenna as shown in Fig. 1. The BS is attached with an FL server. In particular, we separate the CDs into two groups, i.e., the privacy-sensitive CDs and privacy-insensitive CDs. We denote the subset of the privacy-sensitive CDs as $\mathcal{K}^{\text{sp}}$ (here "sp" means strict-privacy), and the subset of the privacy-insensitive CDs as $\mathcal{K}^{\text{np}}$ (here "np" means non-privacy). We use a binary variable x_k to denote the k-th CD's selection of performing either the local training or the centralized training. For each privacy-sensitive CD $k \in \mathcal{K}^{\text{sp}}$, the value of x_k should be

$$x_k = 0,\ k \in \mathcal{K}^{\text{sp}},$$

which means that the privacy-sensitive CD k can only perform the local training.

For each privacy-insensitive CD $k \in \mathcal{K}^{\text{np}}$, if it selects to perform local training, then $x_k = 0$. If CD k selects to perform the centralized training, then $x_k = 1$, namely,

$$x_k = \begin{cases} 0, & \text{using local training} \\ 1, & \text{using centralized training} \end{cases},\ k \in \mathcal{K}^{\text{np}}.$$

We next illustrate the proposed FL with the hybrid local and centralized training as follows.

A. Stage I for All CDs: Global Model Downloading

In Stage I, the FL server sends the global model to all CDs. We denote the size of the global model as S. We use b_k to denote the bandwidth allocated to CD k. We denote the BS's transmit-power as p_0 and the channel coefficient between CD k and the BS as h_k. The CD k's throughput for downloading the global model is

$$R_k^{\text{do}} = b_k \log_2\left(1 + \frac{p_0 h_k}{\sigma b_k}\right),\ \forall k \in \mathcal{K}, \tag{1}$$

where σ denotes the noise power density.

We use y_k^{do} (here "y" stands for transmission time and "do" stands for "model downloading") to denote the CD k's model downloading time, which can be expressed as

$$y_k^{\text{do}} = \frac{S}{R_k^{\text{do}}},\ \forall k \in \mathcal{K}. \tag{2}$$

B. Stage II for the CDs Performing Local Training: Local Training and Model Uploading

As shown in Fig. 1, the illustration of Stage II can be separated into two cases, i.e., the CD with $x_k = 0$ (i.e., performing local training) and the CD with $x_k = 1$ (i.e., performing centralized training). In this subsection, we firstly illustrate the operations of the CDs which perform local training, including the privacy-sensitive CDs in $\mathcal{K}^{\text{sp}}$ as well as those privacy-insensitive CDs but choosing $x_k = 0$. In the next subsection, we will further illustrate the operations of the CDs which perform centralized training, i.e., CDs choosing $x_k = 1$.

In each round of FL iterations, the operations of CDs performing local training include two parts, i.e., (i) local training and (ii) model uploading, which are illustrated as follows.

(i) Local training: After receiving the global model from the FL server, each CD k (with $x_k = 0$) performs local training to update its local model. We denote f_k as the CPU frequency at CD k, with the unit of HZ which means the number of CPU cycles per second. D_k is the number of data samples on the k-th CD, ι represents the data bits for each sample. We consider the mini-batch stochastic gradient descent (SGD) method which is widely used in FL. Here, we assume that the mini-batch size is ϑD_k, which means that the mini-batch size is proportional to the size of total data on the CD, where $0 < \vartheta < 1$ is a constant

and practically it is a very small value. We use variable m to denote the number of local iterations. The computation time for local training on the k-th CD t_k^{loc} (here "t" stands for training time and "loc" stands for "local training") in each round is

$$t_k^{\text{loc}} = \frac{(1-x_k)m\iota\vartheta D_k\zeta_k}{f_k},\ \forall k \in \mathcal{K}, \tag{3}$$

where ζ_k represents the number of CPU cycles for the k-th CD to process one bit data.

Since CD k's CPU power consumption per second can be expressed as κf_k^3, where κ is a coefficient depending on the CPU architecture, the k-th CD's energy consumption for local training in each round can be given by

$$E_k^{\text{loctr}} = (1-x_k)\kappa\iota D_k\zeta_k m f_k^2,\ \forall k \in \mathcal{K}. \tag{4}$$

(ii) Model uploading: After completing their respective local training, the CDs performing local training send their local models to the FL server via FDMA. We denote the CD k's transmit-power as p_k. The CD k's uploading throughput can be expressed as

$$R_k^{\text{up}} = b_k \log_2\left(1+\frac{p_k h_k}{\sigma b_k}\right),\ \forall k \in \mathcal{K}. \tag{5}$$

We use y_k^{loc} to denote the transmission duration for CD k to upload its model, which can be expressed as

$$y_k^{\text{loc}} = (1-x_k)\frac{S}{R_k^{\text{up}}},\ \forall k \in \mathcal{K}. \tag{6}$$

Correspondingly, the k-th CD's energy consumption for uploading its local model to the FL server is

$$E_k^{\text{locup}} = p_k y_k^{\text{loc}} = p_k(1-x_k)\frac{S}{R_k^{\text{up}}},\ \forall k \in \mathcal{K}. \tag{7}$$

Using (4) and (7), the energy consumption of CD k, which selects to perform the local training, can be expressed as $E_k^{\text{loctr}} + E_k^{\text{locup}}$.

C. Stage II for the CDs Performing Centralized Training: Data Offloading and Centralized Training

In this subsection, we illustrate the operations of the CDs which perform centralized training, i.e., CDs choosing $x_k = 1$. In each round of FL iterations, the operations of CDs performing local training include two parts, i.e., (i) data offloading and (ii) central training, which are illustrated as follows.

(i) Data offloading: After receiving the global model from the FL server, each CD k that selects to perform centralized training offloads a mini-batch of its dataset to the FL server. We use y_k^{cen} (here "cen" stands for "performing centralized training and offloading data") to denote the transmission duration to offload the data, which can be expressed as

$$y_k^{\text{cen}} = x_k\frac{\iota\vartheta D_k}{R_k^{\text{up}}},\ \forall k \in \mathcal{K}, \tag{8}$$

where R_k^{up} is given in (5) before.

Correspondingly, the k-th CD's energy consumption for offloading its data to the FL server is

$$E_k^{\text{cen}} = p_k y_k^{\text{cen}} = p_k x_k\frac{\iota\vartheta D_k}{R_k^{\text{up}}},\ \forall k \in \mathcal{K}. \tag{9}$$

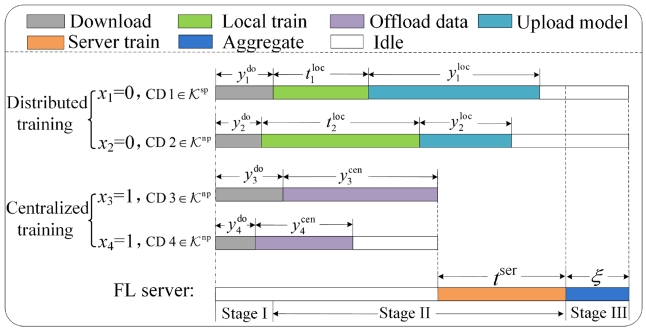

Fig. 2. An illustrative example of the operations in one FL iteration.

(ii) Centralized training: The FL server collects all the offloaded data to form a combined dataset, and then updates the model with the combined dataset. We denote f_0 as the available CPU frequency at the FL server. The centralized training time at the FL server t^{ser} (here "ser" stands for "centralized training by the FL server") can be expressed as

$$t^{\text{ser}} = \sum_{k\in\mathcal{K}}\frac{x_k m\iota\vartheta D_k\zeta_0}{f_0}, \tag{10}$$

where ζ_0 represents the number of CPU cycles for the FL server to process one bit data.

D. Stage III for All CDs: Model Aggregating

In Stage III, the FL server aggregates its trained model (from the centralized training in Stage II) and the collected local models from the CDs which select to perform local training, and further generates an updated global model which will be sent to all CDs in the next round of FL iteration. Since the data size for model aggregating is equivalent at each round, with a fixed computing capacity at the FL server, the time for model aggregating can be considered as a constant which is denoted as ξ.

E. Overall Latency and Energy Consumption of the Proposed FL With Hybrid Local and Centralized Training

An illustrative example of the operations in one FL iteration is demonstrated in Fig. 2. In Fig. 2, CD 1 is privacy-sensitive and performs local training. CD 2 is privacy-insensitive and chooses to perform local training. CD 3 and CD 4 are privacy-insensitive and choose to perform centralized training. By using the example in Fig. 2, we illustrate the single-round latency, including Stage I, Stage II, and Stage III, as follows. In particular, since the latency of Stage III is same for all CDs, we focus on analyzing the latency of Stage I and Stage II below.

- The latency of Stage I and Stage II for the CDs performing local training, e.g., CD 1 and CD 2, includes: i) the model downloading latency y_k^{do}, ii) the local training latency t_k^{loc}, and iii) the model uploading latency y_k^{loc}. Thus, the corresponding latency of Stage I and Stage II for the CDs performing the local training is $\max_{k\in\mathcal{K}}\{y_k^{\text{do}} + t_k^{\text{loc}} + y_k^{\text{loc}}\}$.
- The latency of Stage I and Stage II for the CDs performing centralized training, e.g., CD 3 and CD 4, includes: i) the model downloading latency y_k^{do}, ii) the data uploading latency y_k^{cen}, and iii) the latency of model training on edge server t^{ser}. Thus, the corresponding latency of Stage I and

Stage II for the CDs performing the centralized training is $\max_{k\in\mathcal{K}}\{y_k^{\text{do}}+y_k^{\text{cen}}\}+t^{\text{ser}}$.

Since the FL server can only start Stage III for model aggregating when all CDs complete the procedure of Stage I and Stage II, the single-round latency for the FL can be expressed as

$$T^{\text{single}} = \max\Big\{\max_{k\in\mathcal{K}}\{y_k^{\text{do}}+t_k^{\text{loc}}+y_k^{\text{loc}}\}, \max_{k\in\mathcal{K}}\{y_k^{\text{do}}+y_k^{\text{cen}}\}+t^{\text{ser}}\Big\}+\xi. \tag{11}$$

We use variable n to denote the number of global iterations. The total latency T^{all} for n rounds of global iterations is

$$T^{\text{all}} = nT^{\text{single}}. \tag{12}$$

We next model the energy consumption of each CD as follows. Notice that if $x_k=1$, CD k's energy consumption comes from sending the data to the FL server, i.e., E_k^{cen}. If $x_k=0$, CD k's energy consumption comes from its local training and sending the trained model to the FL server, i.e., $E_k^{\text{loctr}}+E_k^{\text{locup}}$. As a summary, CD k's energy consumption can be expressed as

$$E_k = n(E_k^{\text{loctr}}+E_k^{\text{locup}}+E_k^{\text{cen}}), \forall k\in\mathcal{K}. \tag{13}$$

Reminder that in above (13), either $(E_k^{\text{loctr}}+E_k^{\text{locup}})$ or E_k^{cen} will be positive (and the other one will be zero), according to (4), (7) and (9) before.

F. FL Training Process

We denote $F_k(\varpi)$ as the loss function for CD k's local training, and we use $F_0(\varpi)$ to denote the loss function for centralized training at the FL server with the data collected from the CDs that select to perform centralized training. The objective of FL is to minimize the overall loss function $F(\varpi)$ as

$$\min_{\varpi}\ F(\varpi)=\frac{\sum_{k\in\mathcal{K}}(1-x_k)D_kF_k(\varpi)}{\sum_{k\in\mathcal{K}}D_k}+\frac{F_0(\varpi)\sum_{k\in\mathcal{K}}x_kD_k}{\sum_{k\in\mathcal{K}}D_k}. \tag{14}$$

To analyze the convergence feature of our proposed FL, similar to [24], [25], we make the following assumptions.

Assumption 1: The loss function is L-Lipschitz and γ-strongly convex (the detailed expression can be seen in (54) in Appendix A).

Assumption 2: The stochastic gradient computed on the mini-batch SGD can be bounded by introducing two variables ρ_1 and σ_F^2 (the detailed expression can be seen in (55) in Appendix A).

Assumption 3: We use Γ to quantify the degree of non-independently identical distribution (non-IID) (the detailed expression can be seen in (56) in Appendix A).

The convex and Lipschitz assumptions (i.e., Assumption 1) provide the upper bound and lower bound of $\bigtriangledown^2F_k(\varpi)$, i.e., $\gamma\mathbf{I}\leq\bigtriangledown^2F_k(\varpi)\leq L\mathbf{I}$, which avoids a significant variation on the gradient of the loss function. Moreover, the upper bound and lower bound play an important role in deriving the convergence feature of the proposed FL as illustrated in the proof of Theorem 1. We notice that this similar assumption has also been adopted in [12], [26]. Moreover, although we make the convexity assumption, the convergence feature derived in Theorem 1 can be also expended to non-convex function by using the similar method in [26].

We denote $\mathbf{g}^n$ as the global model at the n-th global iteration. Under the above assumptions, the convergence feature of the proposed FL is derived in Theorem 1.

Theorem 1: We use η_k to represent the decay effect of the loss function for CD k's local training, which can be expressed as

$$\eta_k = 1-\delta\gamma\left(1-\frac{L\delta}{2}-\frac{L\delta}{2}\frac{\rho_1}{\vartheta D_k}\right). \tag{15}$$

Meanwhile, we use η_0 to denote the decay effect of the loss function for the CDs that select to perform the centralized training at the FL server, i.e.,

$$\eta_0 = 1-\delta\gamma\left(1-\frac{L\delta}{2}-\frac{L\delta}{2}\frac{\rho_1}{\vartheta\sum_{k\in\mathcal{K}}x_kD_k}\right). \tag{16}$$

With the learning rate $\delta<\frac{2\vartheta\min\{D_k\}}{L(\vartheta\min\{D_k\}+\rho_1)}$, considering the same number of local iterations m on all CDs and FL server, the convergence feature can be written as

$$F(\mathbf{g}^n)-F^*\leq\underbrace{(1-C_1)^n(F(g^0)-F^*)}_{\text{decay item for global iteration}}+\underbrace{C_3\frac{1-(1-C_1)^n}{C_1}}_{\text{compensation item}}, \tag{17}$$

where

$$C_1=\frac{\beta\gamma^2}{2DL^2}\left(\underbrace{\sum_{k\in\mathcal{K}}(1-x_k)(1-(\eta_k)^m)D_k}_{\text{from local training}}+\underbrace{(1-(\eta_0)^m)\sum_{k\in\mathcal{K}}x_kD_k}_{\text{from centralized training}}\right), \tag{18}$$

$$C_2=\frac{\sigma_F^2}{\vartheta}\left(\underbrace{\sum_{k\in\mathcal{K}}(1-x_k)\frac{1-(\eta_k)^m}{1-\eta_k}}_{\text{from local training}}+\underbrace{\frac{1-(\eta_0)^m}{1-\eta_0}}_{\text{from centralized training}}\right), \tag{19}$$

$$C_3=C_2-C_1\Gamma. \tag{20}$$

Proof: Please refer to Appendix A. ■

Theorem 1 can be intuitively explained as follows. In (17), item $(1-C_1)$ represents the decay rate of global iterations. Since $0<1-C_1<1$, at each FL global iteration, the value of $F(g^n)-F^*$ will decrease with the ratio $1-C_1$. In eq. (17), item $C_3\frac{1-(1-C_1)^n}{C_1}$ represents the compensation for the mini-batch SGD process since we assume the bound of mini-batch SGD (as eq. (55) in Appendix A). In Theorem 1, C_1 can be regarded as the weighted combination of the CDs' and the FL server's training decay items, while C_2 can be regarded as the weighted combination of the CDs' and the FL server's training compensation items. The results of Theorem 1 are demonstrated in Figs. 3 and 4. As demonstrated in Figs. 3 and 4, when more CDs choose to perform centralized training, the FL can achieve a higher convergence rate and the loss function can reach a smaller value.

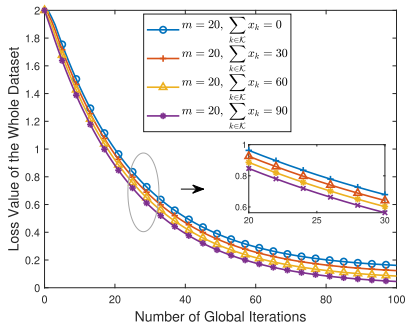

Fig. 3. Convergence behavior for the proposed FL when $m = 20$ ($K = 100$).

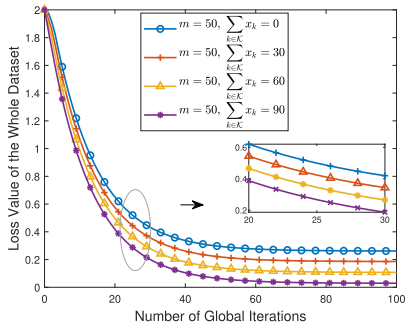

Fig. 4. Convergence behavior for the proposed FL when $m = 50$ ($K = 100$).

We denote $F(\mathbf{g}^0) - F^* = \mu$. According to Theorem 1, we obtain Corollary 1 which is the training loss constraint in our problem formulation as discussed later.

Corollary 1: To achieve the training loss decay rate ε satisfying $F(\mathbf{g}^n) - F^* \leq \varepsilon(F(\mathbf{g}^0) - F^*)$, the number of local iterations m and the number of global iterations n should satisfy

$$-(\varepsilon - (1-C_1)^n)\mu + C_3\frac{1-(1-C_1)^n}{C_1} \leq 0. \tag{21}$$

Notice that m is contained in C_1 and C_3 according to (18), (19) and (20).

Proof: From (17) we obtain

$$(1-C_1)^n\mu + C_3\frac{1-(1-C_1)^n}{C_1} \leq \varepsilon\mu. \tag{22}$$

Form (22), we can obtain the result in Corollary 1. ■

G. Problem Formulation

Based on the above analysis, we aim at minimizing the overall latency of FL and formulate Problem (OLM) as follows (here "OLM" stands for "overall latency minimization").

$$\begin{aligned}
\text{(OLM):}\quad & \min T^{\text{all}} \\
\text{subject to:}\quad & E_k \leq E_k^{\max}, \forall k \in \mathcal{K}, && (23)\\
& \sum_{k\in\mathcal{K}} b_k \leq B, && (24)\\
& x_k \in \{0,1\}, \forall k \in \mathcal{K}^{\text{np}}, && (25)\\
& x_k = 0, \forall k \in \mathcal{K}^{\text{sp}}, && (26)\\
& 1 \leq n \leqslant n_{\max}, n \in \mathbb{Z}^+, && (27)\\
& 1 \leq m \leqslant m_{\max}, m \in \mathbb{Z}^+, && (28)
\end{aligned}$$

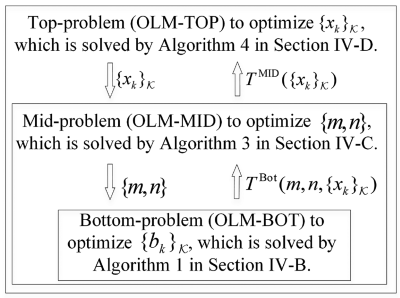

Fig. 5. Illustration of our layered algorithm for solving Problem (OLM).

$$\begin{aligned}
& \text{constraint: } (21),\\
\text{variables:}\quad & m,n,\{x_k\}_{\mathcal{K}},\{b_k\}_{\mathcal{K}}.
\end{aligned}$$

In Problem (OLM), constraint (23) guarantees that each CD's energy consumption cannot exceed its energy budget $E_k^{\max}$. Constraint (24) ensures that the summation of all CDs' bandwidth cannot exceed the available bandwidth denoted by B. Constraint (21) ensures that the training loss decay satisfies the training requirement.

However, due to jointly invoking the series of variables $\{m,n,\{x_k\}_{\mathcal{K}},\{b_k\}_{\mathcal{K}}\}$, Problem (OLM) is a non-convex optimization problem, which means that there exists no general algorithm that can solve Problem (OLM) efficiently. We will propose an efficient algorithm to solve Problem (OLM) in the next section.

IV. Proposed Algorithm for Problem (OLM)

A. Layered Structure of Problem (OLM)

As demonstrated in Fig. 5, to solve the complicated non-convex Problem (OLM), we exploit its layered structure as follows.

Bottom-problem to optimize $\{b_k\}_{\mathcal{K}}$ under given $\{x_k\}_{\mathcal{K}}, m$ and n. We firstly consider that CDs' selections $\{x_k\}_{\mathcal{K}}$ and the FL configuration (i.e., the number of global iterations and local iterations) are given, and aim at optimizing the bandwidth allocations $\{b_k\}_{\mathcal{K}}$ to minimize the overall latency. This leads to the bottom problem (i.e., Problem (OLM-BOT)) as follows:

$$\begin{aligned}
\text{(OLM-BOT):}\quad & T^{\text{BOT}}(\{x_k\}_{\mathcal{K}},n,m) = \min T^{\text{all}}\\
\text{subject to:}\quad & \text{constraints: } (23),(24),\\
\text{variables:}\quad & \{b_k\}_{\mathcal{K}}.
\end{aligned}$$

Mid-problem to optimize m and n under given $\{x_k\}_{\mathcal{K}}$. With the optimal value of $T^{\text{BOT}}(\{x_k\}_{\mathcal{K}},n,m)$ by solving the bottom problem (OLM-BOT), we then continue to optimize the number of local iterations m and the number of global iterations n. This leads to the middle layer problem (i.e., Problem (OLM-MID)) as follows:

$$\begin{aligned}
\text{(OLM-MID):}\quad & T^{\text{MID}}(\{x_k\}_{\mathcal{K}}) = \min\ T^{\text{BOT}}(\{x_k\}_{\mathcal{K}},n,m)\\
\text{subject to:}\quad & \text{constraints: } (21),(23),(27),(28),\\
\text{variables:}\quad & m,n.
\end{aligned}$$

Top-problem to optimize $\{x_k\}_{\mathcal{K}}$. After obtaining $T^{\text{MID}}(\{x_k\}_{\mathcal{K}})$ by solving Problem (OLM-MID), we continue to optimize CDs' selections $\{x_k\}_{\mathcal{K}}$. This leads to the top problem

(i.e., Problem (OLM-TOP)) as follows:

$$\begin{array}{ll}\text{(OLM-TOP):} & T^* = \min\ T^{\text{MID}}(\{x_k\}_{\mathcal{K}}) \\ \text{subject to:} & x_k \in \{0,1\}, \forall k \in \mathcal{K}^{\text{np}},\ x_k = 0, \forall k \in \mathcal{K}^{\text{sp}}, \\ \text{variables:} & \{x_k\}_{\mathcal{K}}.\end{array}$$

B. Proposed Algorithms for Solving Problem (OLM-BOT)

We use $\hat{T}_k$ to denote the latency of CD k for Stage I and Stage II in one round, specifically,

$$\hat{T}_k = \begin{cases} y_k^{\text{do}} + t_k^{\text{loc}} + y_k^{\text{loc}},\ x_k = 0, \\ y_k^{\text{do}} + y_k^{\text{cen}} + t^{\text{ser}},\ x_k = 1. \end{cases} \tag{29}$$

The objective of Problem (OLM-BOT) can be rewritten as

$$T^{\text{all}} = n\left(\max_{k\in\mathcal{K}}\{\hat{T}_k\} + \xi\right). \tag{30}$$

Recall that ξ denotes the model aggregating time which is described in Section III.D before. Thus, to minimize the objective of Problem (OLM-BOT) is equivalent to minimize $\max_{k\in\mathcal{K}}\{\hat{T}_k\}$. We introduce a variable $\hat{T} = \max_{k\in\mathcal{K}}\{\hat{T}_k\}$. With $\hat{T}$, we transform Problem (OLM-BOT) into the following one:

$$\begin{array}{lll}\text{(OLM-BOT-E):} & \min\ \hat{T} & \\ \text{subject to:} & \hat{T}_k \le \hat{T},\ \forall k \in \mathcal{K}, & (31) \\ & \text{constraints: } (23), (24), & \\ \text{variables:} & \{b_k\}_{\mathcal{K}}. & \end{array}$$

To solve Problem (OLM-BOT-E), we firstly identify the monotonic property of $\hat{T}_k$ and E_k as Lemma 1 and Lemma 2.

Lemma 1: For each CD k, $\hat{T}_k$ is monotonically decreasing with b_k.

Proof: Please refer to Appendix B. ■

Lemma 2: For each CD k, E_k is monotonically decreasing with b_k.

Proof: Please refer to Appendix C. ■

We use $\{b_k^{\text{low}}\}_{\mathcal{K}}$ to denote the lower bound of the feasible region constrained by the energy budget constraint (23). Specifically, there exists

$$b_k^{\text{low}} = \min\{b_k | E_k \le E_k^{\max}\},\ \forall k \in \mathcal{K}. \tag{32}$$

Reminder that E_k can be regarded as an implicit function of b_k. Moreover, according to Lemma 2, E_k is decreasing with b_k. Thus, for each CD k, we can use the bisection search method to determine the value b_k^{low} in (32).

Lemma 3: Problem (OLM-BOT-E) is a convex problem.

Proof: Please refer to Appendix D. ■

Lemma 3 enables us to use Karush-Kuhn-Tucker (KKT) conditions to solve Problem (OLM-BOT-E). We use λ_k to denote the Lagrangian multiplier for constraint $\hat{T}_k \le \hat{T}$, $\forall k \in \mathcal{K}$. We use u to denote the Lagrangian multiplier for constraint $\sum_{k\in\mathcal{K}} b_k \le B$. We use υ_k to denote the Lagrangian multiplier for constraint $E_k \le E_k^{\max}$, $\forall k \in \mathcal{K}$. Thus, we can derive the Lagrangian function as follows:

$$L(b_k,\lambda_k,u,\upsilon_k) = \hat{T} + \sum_{k\in\mathcal{K}} \lambda_k(\hat{T}_k - \hat{T}) + u\left(\sum_{k\in\mathcal{K}} b_k - B\right)$$

Algorithm 1: BA Algorithm to Solve Problem (OLM-BOT).

Input: the values of $\{x_k\}_{\mathcal{K}}$, m.
1: Initialize $\mathcal{S}^0 = \varnothing$. Initialize $i = 0$.
2: **while** $\mathcal{K} \setminus \mathcal{S}^i \neq \varnothing$ or $\mathcal{S}^{i+1} \neq \mathcal{S}^i$ **do**
3: With $\mathcal{S}^i$, use Subroutine-BA to compute $\{\hat{b}_k\}_{\mathcal{K}}^i$, which are the solutions of (38), (39) and (40).
4: Update $\hat{\mathcal{S}} = \{j | \hat{b}_j^i < b_j^{\text{low}}\}$. Set $b_j^* = b_j^{\text{low}}$, $\forall j \in \hat{\mathcal{S}}$.
5: Update $\mathcal{S}^{i+1} = \mathcal{S}^i \cup \hat{\mathcal{S}}$.
6: Update $i = i + 1$.
7: **end while**
Output:
$b_k^* = \hat{b}_k^i$, $\forall k \in \mathcal{K} \setminus \mathcal{S}^{i+1}$. $b_k^* = b_j^{\text{low}}$, $\forall j \in \mathcal{S}^{i+1}$.

Algorithm 2: Subroutine-BA to Compute $\{\hat{b}_k\}_{\mathcal{K}}^i$ (i.e., solutions of (38), (39) and (40) under given $\mathcal{S}^i$).

Input: the values of $\{x_k\}_{\mathcal{K}}$, m, and set $\mathcal{S}^i$.
1: Initialize ε with a small value. Initialize $\hat{T}^{\text{up}}$ with a large value. Initialize $\hat{T}^{\text{low}} = 0$. Initialize $b_k^{\text{cur}} = b_k^{\text{low}}$, $\forall k \in \mathcal{K}$.
2: **while** $\sum_{k\in\mathcal{K}} b_k^{\text{cur}} > (1+\varepsilon)B$ or $\sum_{k\in\mathcal{K}} b_k^{\text{cur}} < (1-\varepsilon)B$ **do**
3: Update $\hat{T}^{\text{cur}} = \frac{1}{2}(\hat{T}^{\text{up}} + \hat{T}^{\text{low}})$.
4: Compute $\{b_k^{\text{cur}}\}_{\mathcal{K}\setminus\mathcal{S}^i}$ satisfying (38) with the bisection search.
5: **if** $\sum_{k\in\mathcal{K}} b_k^{\text{cur}} \le (1-\varepsilon)B$ **then**
6: Update $\hat{T}^{\text{up}} = \hat{T}^{\text{cur}}$.
7: **else**
8: Update $\hat{T}^{\text{low}} = \hat{T}^{\text{cur}}$.
9: **end if**
10: **end while**
Output: $\hat{b}_k^i = b_k^{\text{cur}}, \forall k \in \mathcal{K}$.

$$+ \sum_{k\in\mathcal{K}} \upsilon_k(E_k - E_k^{\max}). \tag{33}$$

From the KKT conditions, we obtain

$$\frac{\partial L(b_k,\lambda_k,u,\upsilon_k)}{\partial b_k} = -\lambda_k\frac{\partial T_k}{\partial b_k} + u + \upsilon_k\frac{\partial E_k}{\partial b_k} = 0, \forall k \in \mathcal{K}. \tag{34}$$

Based on Lemma 1 and Lemma 2, we conclude that $\frac{\partial T_k}{\partial b_k} < 0$ and $\frac{\partial E_k}{\partial b_k} < 0$. We then consider u. Reminder that $u \ge 0$. If $u = 0$, then $\lambda_k = 0$, $\forall k \in \mathcal{K}$, $\upsilon_k = 0$, $\forall k \in \mathcal{K}$. It implies that for any values of $\{b_k\}_{\mathcal{K}}$, there always exists $\frac{\partial L(b_k,\lambda_k,u,\upsilon_k)}{\partial b_k} = 0$, $\forall k \in \mathcal{K}$. As a result, the objective value does not change for any values of $\{b_k\}_{\mathcal{K}}$ when $u = 0$, which incurs a contradiction. Thus, we can conclude that $u > 0$.

From the complementary slackness condition that $u(\sum_{k\in\mathcal{K}} b_k - B) = 0$, we conclude that the optimal solutions of $\{b_k\}_{\mathcal{K}}$ should satisfy

$$\sum_{k\in\mathcal{K}} b_k^* - B = 0. \tag{35}$$

We next consider υ_k. Since $\upsilon_k \ge 0$, $\forall k \in \mathcal{K}$, we divide set $\mathcal{K}$ into two sub-sets $\mathcal{S}$ and $\mathcal{K} \setminus \mathcal{S}$, where $\mathcal{S} = \{j | \upsilon_j > 0\}$. For CD

$k \in \mathcal{K} \setminus \mathcal{S}$, since $v_k = 0$, from (34), we obtain

$$-\lambda_k \frac{\partial T_k}{\partial b_k} + u = 0, \forall k \in \mathcal{K} \setminus \mathcal{S}.$$

Since $u > 0$ and $\frac{\partial T_k}{\partial b_k} < 0$, we obtain

$$\lambda_k > 0, \forall k \in \mathcal{K} \setminus \mathcal{S}.$$

From the complementary slackness condition that $\lambda_k(\hat{T}_k - \hat{T}) = 0$, $\forall k \in \mathcal{K}$, we conclude that

$$\hat{T}_k = \hat{T}, \forall k \in \mathcal{K} \setminus \mathcal{S}. \tag{36}$$

For CD $j \in \mathcal{S}$, since $v_j > 0$, from the complementary slackness condition that $v_j(E_j - E_j^{\max}) = 0$, $\forall j \in \mathcal{K}$, and recalling that b_j^{low} is the value satisfying $E_j = E_j^{\max}$, we conclude that

$$b_j^* = b_j^{\text{low}}, \forall j \in \mathcal{S}. \tag{37}$$

Based on the above analysis, we conclude that the optimal solutions of $\{b_k^*\}_{\mathcal{K}}$ should satisfy Proposition 1 as follows.

Proposition 1: The optimal solutions $\{b_k^*\}_{\mathcal{K}}$ of Problem (OLM-BOT-E) should satisfy the equation group as

$$\begin{cases} \hat{T}_k = \hat{T}, \forall k \in \mathcal{K} \setminus \mathcal{S}, & (38) \\ \sum_{k \in (\mathcal{K} \setminus \mathcal{S})} b_k^* + \sum_{j \in \mathcal{S}} b_j^{\text{low}} = B, & (39) \\ b_j^* = b_j^{\text{low}}, \forall j \in \mathcal{S}. & (40) \end{cases}$$

Proof: Combining (35), (36) and (37), we can derive the conclusion in Proposition 1. ■

With Proposition 1, we propose BA Algorithm (here "BA" stands for "bandwidth allocation") as Algorithm 1 to solve Problem (OLM-BOT). The main idea of BA Algorithm is to recursively update $\mathcal{S}^{i+1} = \mathcal{S}^i \cup \{j | \hat{b}_j^i < b_j^{\text{low}}\}$ until either (i) there remains no elements in $\mathcal{K} \setminus \mathcal{S}^i$ or (ii) $\mathcal{K} \setminus \mathcal{S}^i$ keeps unchanged.

Proposition 2: The solution yielded by BA Algorithm suffices to be the optimal solution for Problem (OLM-BOT).

Proof: BA Algorithm is based on the recursion method and we conduct the proof by recursively revealing the relationship of the i-th and $(i+1)$-th recursions. Since the optimal solutions of $\{b_k^*\}_{\mathcal{S}^i}$ are obtained in the $(i-1)$-th recursion, we focus on the analysis of $\{b_k^*\}_{\mathcal{K} \setminus \mathcal{S}^i}$. In the i-th recursion, after obtaining $\{\hat{b}_k\}_{\mathcal{K}}^i$ by solving (38) and (39) under $\mathcal{S}^i$, we denote $\hat{\mathcal{S}} = \{j | \hat{b}_j^i < b_j^{\text{low}}\}$. For CD $j \in \hat{\mathcal{S}}$, to satisfy the energy budget constraint (23), the solution should satisfy

$$b_j^* \geq b_j^{\text{low}} > \hat{b}_j^i, \ \forall j \in \hat{\mathcal{S}}. \tag{41}$$

From the monotonic property of $\hat{T}_k$ ($\hat{T}_k$ is implicit function of b_k), we obtain

$$\hat{T}_j(b_j^*) \leq \hat{T}_j(b_j^{\text{low}}) < \hat{T}_j(\hat{b}_j^i), \forall j \in \hat{\mathcal{S}}. \tag{42}$$

Recall that $\{\hat{b}_k^i\}_{\mathcal{K} \setminus \mathcal{S}^i}$ are the points satisfying $\hat{T}_k(\hat{b}_k^i) = \hat{T}$, $\forall k \in \mathcal{K} \setminus \mathcal{S}^i$. (41) implies that more bandwidth (more than $\hat{b}_k^i$) should be allocated to the CDs in set $\hat{\mathcal{S}}$. Thus, the bandwidth remained for CD $k \in \mathcal{K} \setminus (\mathcal{S}^i \cup \hat{\mathcal{S}})$ is non-increasing (smaller than or equal to $\hat{b}_k^i$), which implies that

$$b_k^* \leq \hat{b}_k^i, \forall k \in \mathcal{K} \setminus (\mathcal{S}^i \cup \hat{\mathcal{S}}).$$

Also from the monotonic property of $\hat{T}_k$, we obtain

$$\hat{T}_k(\hat{b}_k^i) \leq \hat{T}_k(b_k^*), \forall k \in \mathcal{K} \setminus (\mathcal{S}^i \cup \hat{\mathcal{S}}). \tag{43}$$

Since $\{\hat{b}_k^i\}_{\mathcal{K} \setminus \mathcal{S}^i}$ are the points satisfying

$$\hat{T}_j(\hat{b}_j^i) = \hat{T}_k(\hat{b}_k^i), \forall j \in \mathcal{S}^i, \forall k \in \mathcal{K} \setminus (\mathcal{S}^i \cup \hat{\mathcal{S}}),$$

by combing (42) and (43), we obtain

$$\hat{T}_j(b_j^*) < \hat{T}_k(b_k^*), \forall j \in \hat{\mathcal{S}}, \forall k \in \mathcal{K} \setminus (\mathcal{S}^i \cup \hat{\mathcal{S}}),$$

which implies that the solutions of $\{b_k^*\}_{\mathcal{K} \setminus \mathcal{S}^i}$ satisfy $\max_{j \in \hat{\mathcal{S}}} \{\hat{T}_j\} < \max_{k \in \mathcal{K} \setminus (\mathcal{S}^i \cup \hat{\mathcal{S}})} \{\hat{T}_k\}$. Thus, to minimize the objective of $\max_{k \in \mathcal{K} \setminus \mathcal{S}^i} \{\hat{T}_k\}$ is equivalent to minimize the following one:

$$\max \left\{ \max_{j \in \hat{\mathcal{S}}} \left\{ \hat{T}_j \right\}, \max_{k \in \mathcal{K} \setminus (\mathcal{S}^i \cup \hat{\mathcal{S}})} \left\{ \hat{T}_k \right\} \right\} = \max_{k \in \mathcal{K} \setminus (\mathcal{S}^i \cup \hat{\mathcal{S}})} \left\{ \hat{T}_k \right\}. \tag{44}$$

Then, we conduct the analysis as follows.

- For CD $j \in \hat{\mathcal{S}}$, to minimize $\max_{k \in \mathcal{K} \setminus (\mathcal{S}^i \cup \hat{\mathcal{S}})} \{\hat{T}_k\}$, we should allocate more bandwidth to set $\mathcal{K} \setminus (\mathcal{S}^i \cup \hat{\mathcal{S}})$. Specifically, we decrease the bandwidth of each CD $j \in \hat{\mathcal{S}}$ until it reaches its low bound b_j^{low}. Thus, we conclude that $b_j^* = b_j^{\text{low}}, \forall j \in \hat{\mathcal{S}}$.
- For CD $k \in \mathcal{K} \setminus (\mathcal{S}^i \cup \hat{\mathcal{S}})$, to minimize $\max_{k \in \mathcal{K} \setminus (\mathcal{S}^i \cup \hat{\mathcal{S}})} \{\hat{T}_k\}$, we update $\mathcal{S}^{i+1} = \mathcal{S}^i \cup \hat{\mathcal{S}}$ and continue to minimize $\max_{k \in \mathcal{K} \setminus \mathcal{S}^{i+1}} \{\hat{T}_k\}$ in the next recursion.

This completes the proof. ■

Notice that in Step 3 of BA Algorithm, we use a subroutine (i.e., Subroutine-BA as Algorithm 2) to compute the value of $\{\hat{b}_k\}_{\mathcal{K}}$. Subroutine-BA can be regarded as a two-layer bisection search method. In Step 4, given the value of $\hat{T}$, the value of $\hat{b}_k^i$ satisfying $\hat{T}_k = \hat{T}$ as (38) can be computed by the bisection search method according to the monotonic property of $\hat{T}_k$ in Lemma 1. Then, from Step 5 to Step 10, we check whether the obtained $\hat{b}_k^i$ can satisfy $\sum_{k \in \mathcal{K}} \hat{b}_k^i = B$ as (39). If yes, we output the final solution. Otherwise, we update the value of $\hat{T}$ with the bisection search method, since the value of $\sum_{k \in \mathcal{K}} \hat{b}_k^i$ is monotonically decreasing with $\hat{T}$ according to Lemma 1.

C. Proposed Algorithm for Solving Problem (OLM-MID)

After obtaining the optimal solutions of $\{b_k\}_{\mathcal{K}}$ in the previous subsection, we next optimize the solutions of the number of local iterations m and the number of global iterations n. The two coupling variables m and n incur the difficulty for solving Problem (OLM-MID). To address this difficulty, we firstly determine the optimal value of n under given m, and then optimize m. The details are as follows.

i) Determining the optimal value of n under given m: Since T^{single} is given under the given value of m, the objective $T^{\text{all}} = nT^{\text{single}}$ is monotonically decreasing with n. Thus, to minimize the overall latency T^{all}, we should decrease the value of n until it reaches the lower bound of its feasible region. We denote n^{lower} and n^{upper} as the lower and upper bounds of the feasible region for n, which is constrained by the training loss constraint

Algorithm 3: Computing the Optimal Values of m and n.

Input: the value of $\{x_k\}_{\mathcal{K}}$.
1: Initialize set $\mathcal{T} = \varnothing$.
2: **for** $m = 1 : m_{\max}$ **do**
3: Compute n^{lower} satisfying (47) with the bisection search method (and the $\lceil \cdot \rceil$ operation).
4: Solve Problem (OLM-BOT) with BA algorithm. Compute $\{E_k\}_{\mathcal{K}}$.
5: **if** $\max_{k\in\mathcal{K}} \{E_k - E_k^{\max}\} \leq 0$ and $n^{\text{lower}} \leq n_{\max}$ **then**
6: Compute T^{all}, and add the tuple $(n^{\text{lower}}, m, T^{\text{all}})$ into set $\mathcal{T}$.
7: **end if**
8: **end for**
Output: the optimal solution is the tuple with the smallest value of T^{all} in set $\mathcal{T}$.

(21) and the energy budget constraint (23) [1]. The training loss constraint (21) comes from

$$F(\mathbf{g}^n) - F^* \leq \varepsilon\mu. \tag{45}$$

The energy budget constraint $E_k \leq E_k^{\max}, \forall k \in \mathcal{K}$ can be rewritten as

$$\max_{k\in\mathcal{K}} \{E_k - E_k^{\max}\} \leq 0. \tag{46}$$

Lemma 4: In constraint (45), $F(\mathbf{g}^n) - F^*$ is monotonically decreasing with n.

Proof: It is observed in (17) that the more rounds of the global iterations, the larger decay of the FL training loss. Thus, we can obtain the conclusion in Lemma 4. ■

Lemma 4 implies that constraint (45) gives a lower bound of the feasible region for n. Specifically,

$$n^{\text{lower}} = \min\{n \in \mathbb{Z}^+ | F(\mathbf{g}^n) - F^* \leq \varepsilon\mu\}. \tag{47}$$

Reminder that $F(\mathbf{g}^n) - F^*$ can be regarded as an implicit function of n. Moreover, according to Lemma 4, $F(\mathbf{g}^n) - F^*$ is decreasing with n. To compute n^{lower}, we firstly treat n^{lower} as a continuous value, which can be determined by the bisection search method. Then, we obtain the integer solution of n^{lower} by taking the nearest upper integer of its corresponding continuous solution according to the property of discrete monotonic optimization [27].

Lemma 5: In constraint (46), $\max_{k\in\mathcal{K}} \{E_k - E_k^{\max}\}$ is monotonically increasing with n.

Proof: From the expression of E_k as (13), we obtain

$$\max_{k\in\mathcal{K}} \{E_k - E_k^{\max}\} = n \left(\max_{k\in\mathcal{K}} \{E_k^{\text{loctr}} + E_k^{\text{locup}} + E_k^{\text{cen}} - E_k^{\max}\} \right). \tag{48}$$

In (48), since E_k^{loctr}, E_k^{locup} and E_k^{cen} are given under the given value of m, we can get the conclusion in Lemma 5. ■

Lemma 5 implies that constraint (46) gives an upper bound of the feasible region for n, i.e.,

$$n^{\text{upper}} = \max\{n \in \mathbb{Z}^+ | \max_{k\in\mathcal{K}} \{E_k - E_k^{\max}\} \leq 0,\ n \leq n_{\max}\}.$$

[1]It is noticed that both n^{lower} and n^{upper} depend on the currently given value of m. For the sake of clear presentation, we do not explicitly include m in the notations of n^{lower} and n^{upper}.

Algorithm 4: SL-CE Algorithm for Problem (OLM-TOP).

1: Initialize $\mathbf{q}^1$ and $\mathbf{q}^0$ and ϵ. Initialize $i = 0$.
2: **while** $\|\mathbf{q}^{i+1} - \mathbf{q}^i\| \geq \epsilon$ **do**
3: Update L samples of $\{\mathbf{x}_l\}_{l=1,2,\ldots,L}$. Specifically, in each sample $\mathbf{x} = [x_1, x_2, \ldots x_K]$, for CD $k \in \mathcal{K}^{\text{np}}$, the value of x_k is generated following the probabilistic distribution in eq. (50). For CD $k \in \mathcal{K}^{\text{sp}}$, we set $x_k = 0$.
4: For each sample $\mathbf{x}_l$, solve Problem (OLM-MID) with Algorithm 3 and obtain the optimal objective value $T^{\text{MID}}(\mathbf{x}_l)$.
5: Re-order the samples $\{\mathbf{x}_l\}_{l\in\mathcal{L}}$ as
$$T^{\text{MID}}(\mathbf{x}_1) \leq T^{\text{MID}}(\mathbf{x}_2) < \ldots < T^{\text{MID}}(\mathbf{x}_L).$$
6: Choose $\hat{L}$ best samples from the re-ordered set $\{\mathbf{x}_l\}_{l\in\hat{\mathcal{L}}}$.
7: Update the Bernoulli distribution parameter in eq. (50) as
$$\mathbf{q}^{i+1} = \frac{1}{\hat{L}} \sum_{l=1}^{\hat{L}} \mathbf{x}_l. \tag{49}$$
8: Update $i = i + 1$.
9: **end while**
Output: the optimal solutions of $\{x_k^*\}_{\mathcal{K}}$.

To guarantee a non-empty feasible region, the values of n^{lower} and n^{upper} should satisfy $n^{\text{lower}} \leq n^{\text{upper}}$. However, if a given value of m leads to $n^{\text{lower}} > n^{\text{upper}}$, which implies that the given value of m is infeasible, then we will change another value of m for a new search for the solution of n. Checking whether $n^{\text{lower}} \leq n^{\text{upper}}$ is equivalent to Step 5 in Algorithm 3.

ii) Computing the optimal solution of m: The solution of m is obtained by the exhaustive search. The exhaustive searching in a proper range $[0, m_{\max}]$ only consumes a small complexity since m is an integer.

The detailed procedures of our algorithm to compute the solutions of m and n are demonstrated in Algorithm 3. Step 5 is equivalent to checking whether $n^{\text{lower}} \leq n^{\text{upper}}$. If $n^{\text{lower}} \leq n^{\text{upper}}$, we add the current tuple $(n^{\text{lower}}, m, T^{\text{all}})$ into set $\mathcal{T}$ as shown in Step 6 in Algorithm 3. Otherwise, we continue to evaluate the next value of m for a new round of iteration for evaluating the corresponding value of n.

D. Proposed Algorithm for Solving Problem (OLM-TOP)

Problem (OLM-TOP) can be regarded as a typically combinatorial optimization problem. To solve this problem, we adopt a stochastic learning based algorithm which is based on the measure of cross-entropy [28], [29], i.e., our SL-CE Algorithm in Algorithm 4. Based on the principle of CE, the key idea of our SL-CE Algorithm is to find the best probabilistic distribution for the matchment of the input-output relationship via adaptive sampling. In particular, our SL-CE Algorithm involves an iterative procedure including two phases: i) generating a profile of random samples according to the current probabilistic distribution, and ii) updating the parameters of the probabilistic distribution to produce improved samples for the next iteration. We illustrate the details as follows.

For modeling the probabilistic learning, the CDs' selections $\{x_k\}_{\mathcal{K}}^{\text{np}}$ can be regarded as random variables. We generate L samples of $\{x_k\}_{\mathcal{K}^{\text{np}}}$ following the Bernoulli distributions with parameters $\{q_k\}_{\mathcal{K}^{\text{np}}}$ as

$$\Phi_k(x_k) = q_k^{x_k}(1-q_k)^{(1-x_k)},\ \forall k \in \mathcal{K}^{\text{np}}. \tag{50}$$

At each iteration, $\mathbf{q} = [q_1, \ldots q_{\mathcal{K}^{\text{np}}}]$ is updated for improving the matching between the probabilistic distribution and the input-output relationship. The key steps of Algorithm 4 are illustrated as follows.

- *Step 3:* A batch of L samples is updated with the current Bernoulli distribution parameter $\mathbf{q}$. In each sample, the values of $\{x_k\}_{\mathcal{K}^{\text{np}}}$ for the privacy-insensitive CDs are generated according to eq. (50). The values of $\{x_k\}_{\mathcal{K}^{\text{sp}}}$ for the privacy-sensitive CDs are set to zeros.
- *Step 6 to Step 7:* The $\hat{L}$ best samples are exploited to obtain an improvement parameter q_k of the Bernoulli distribution. In Step 7, we utilize the criterion of CE to update the probabilistic distributions, which is equivalent to solve the problem as follows.

$$q_k^* = \arg \max_{0 \le q_k \le 1} \mathbb{E}(q_k^{x_k}(1-q_k)^{(1-x_k)}), \tag{51}$$

where $\mathbb{E}$ denotes the expectation.

E. Complexity Analysis

We firstly discuss the complexity of BA Algorithm for solving Problem (OLM-BOT). BA Algorithm requires at most K rounds (K rounds corresponds to the worst case) for invoking Subroutine-BA to compute $\{\hat{b}_k^i\}_{\mathcal{K}\setminus\mathcal{S}^i}$. Subroutine-BA can be regarded as a two-layer bisection search method. The bottom-layer bisection search requires at most $A^i \log_2(\frac{b^{\text{ini}}}{\varepsilon})$ iterations, where b^{ini} is the initial value of the upper bound of b_k^{up} and ε is the tolerable computation-error which is used as the stopping criterion. A^i represents the number of elements in the set $\mathcal{K} \setminus \mathcal{S}^i$ at the i-th invocation of Subroutine-BA. The top-layer bisection search requires at most $\log_2(\frac{T^{\text{ini}}}{\varepsilon})$ iterations, where T^{ini} is the initial value of the upper bound of $\hat{T}^{\text{up}}$. The Subroutine-BA requires at most $A^i \log_2(\frac{b^{\text{ini}}}{\varepsilon})\log_2(\frac{T^{\text{ini}}}{\varepsilon})$ iterations. For the worst case, we need K rounds invocation of Subroutine-BA, and at the i-th invocation of Subroutine-BA, the value of A^i is $K-i$. To solve Problem (OLM-BOT), at most $I_{\text{BOT}} = \frac{1}{2}K(K+1)\log_2(\frac{b^{\text{ini}}}{\varepsilon})\log_2(\frac{T^{\text{ini}}}{\varepsilon})$ iterations are required.

We next analyze the complexity of Algorithm 3 for solving Problem (OLM-MID). In Algorithm 3, to compute the optimal value of n under given m, the bisection search is utilized to compute n^{low}, which requires at most $\log_2(\frac{n^{\text{ini}}}{\varepsilon})$ iterations, where n^{ini} is the initial value of the upper bound of n^{low}. Then, we utilize the exhaustive search to find the optimal solution of m, which requires $m_{\max}$ times of enumeration. Thus, to solve Problem (OLM-MID), at most $I_{\text{MID}} = m_{\max}\log_2(\frac{n^{\text{ini}}}{\varepsilon})I_{\text{BOT}}$ iterations are required, where I_{BOT} denotes the number of iterations required for the convergence of Problem (OLM-BOT) as demonstrated before.

We finally analyze the complexity of SL-CE Algorithm for solving Problem (OLM-TOP). In each round of iterations, we need to generate a total number of L samples of $\mathbf{x}_l$. For each sample of $\mathbf{x}_l$, we invoke Algorithm 3 to solve Problem (OLM-TOP). Let Q denote the total number of iterations required for our SL-CE Algorithm to converge. Then, to solve Problem (OLM-TOP), QLI_{MID} iterations are required, where I_{MID} denotes the number of iterations required for the convergence of Problem (OLM-MID) as demonstrated before.

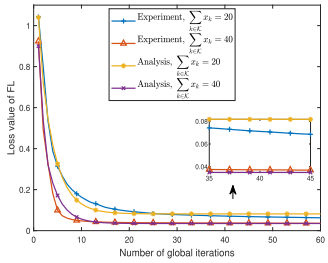

Fig. 6. Evaluation on real dataset of MINST.

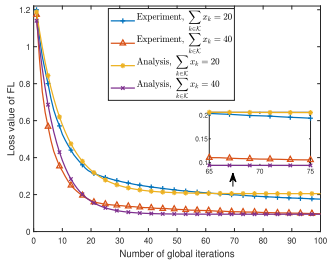

Fig. 7. Evaluation on real dataset of Fashion-MINST.

V. Numerical Results

In this section, we firstly evaluate the derived convergence feature of the proposed FL with the hybrid training. We then demonstrate the performance advantage of our proposed algorithm in comparison with two benchmark algorithms. We next demonstrate the performance advantage of the FL with hybrid training.

We assume that the BS is located at $(0,0)$m, and the CDs are randomly located in a circular region with the radius of 200 m. The path loss between the BS and CD k is modeled as $\text{PL}_k = \text{PL}_0 - 10\alpha\log_{10}\frac{d_k}{d_0}$, where PL_0 denotes the path loss at the reference distance d_0, and α denotes the path loss exponent. We set $\text{PL}_0 = 30$ dB, $\alpha = 2$, and $d_0 = 1$m. The percentage of privacy-sensitive CDs is set as 20%, i.e., the number of privacy-sensitive CDs is $0.2K$. Other parameters are set similar to those in [26], [30] as follows. $K = 50$, $S = 0.1 \times 10^6$ bits, $\zeta_k = 0.3 \times 10^4$ cycles/bit, $\iota = 1 \times 10^4$ bits/sample, $\vartheta = 0.005$, $\xi = 1$ s, $\sigma = 4 \times 10^{-21}$ W/HZ, $B = 5 \times 10^6$ HZ, $f_0 = 3 \times 10^{10}$ HZ, $\kappa = 1 \times 10^{-28}$, $f_k \in [1,6] \times 10^8$ HZ, $E_k^{\max} \in [1,2] \times 10^4$ J, $p_k \in [0.1,0.2]$ W.

A. Convergence Feature of FL With Hybrid Training

We run the FL with real datasets of MNIST and Fashion-MNIST [31]. MNIST is a dataset comprising of handwritten images of the numbers 0 to 9. Fashion-MNIST contains images of fashion products of 10 categories. Here, we focus on the non-IID case since the practical data distribution is usually non-IID.

Figs. 6 and 7 demonstrate the convergence feature of the proposed FL on the dataset of MNIST and Fashion-MNIST. In Figs. 6 and 7, "analysis" means the derived convergence feature of the proposed FL scheme according to Theorem 1, while "experiment" means the FL training with the real dataset.

TABLE I
PERFORMANCE GAIN VERSUS DIFFERENT NUMBERS OF THE CDS.

Number of CDs	10	20	30	40	50
Performance gain	20.8%	24.7%	26.3%	29.4%	17.6%
Number of CDs	60	70	80	90	100
Performance gain	26.2%	16.7%	20.7%	13.3%	13.9%

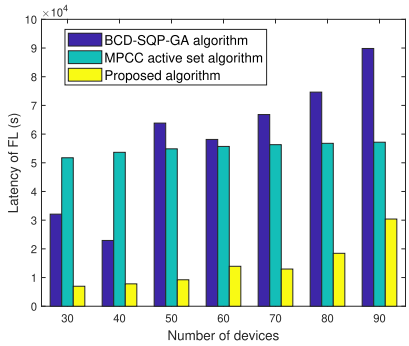

Fig. 8. Impact of the number of CDs.

The results in Figs. 6 and 7 show that the training performance on the real dataset is improved when more CDs choose to perform centralized training, which is consistent with the convergence feature described in Theorem 1. The results in Figs. 6 and 7 also show that the difference of our analysis of the convergence feature and the experimental results based on the real dataset is very small.

We also evaluate the scalability of the proposed FL with hybrid training by demonstrating its performance versus different numbers of the CDs, and the results are shown in Table I. In Table I, the performance gain represents how much percent our proposed FL with hybrid local and centralized training can outperform the conventional FL without using the centralized training. The results in Table I show that our proposed FL with hybrid local and centralized training can achieve an obvious performance gain when the number of the CDs increases.

B. Performance of the Proposed Algorithm

We evaluate the performance of the proposed algorithm by comparing it with two benchmarks as follows.

- BCD-SQP-GA algorithm. Specifically, we utilize the block coordinate descent (BCD) method to divide Problem (OLM) into three subproblems as follows. i) The first subproblem is to optimize the continuous variable $\{b_k\}_{\mathcal{K}}$, which is solved by the sequential quadratic programming (SQP) algorithm [32]. ii) The second subproblem is to optimize the integer variable $\{x_k\}_{\mathcal{K}}$, which is solved by the genetic algorithm (GA) [33]. iii) The third subproblem is to optimize m and n, which is solved by Algorithm 3 in Section IV-C.
- MPCC active set algorithm. Specifically, we firstly transform the original problem with integer constraints to mathematical programs with complementary constraints (MPCC) [34], and then utilize the active set method [35] to solve it.

Fig. 8 demonstrates the performance of the proposed algorithm versus the number of CDs. The results in Fig. 8 show that the latency of FL is increasing with the number of CDs. The results in Fig. 8 also show that our proposed algorithm can outperform both BCD-SQP-GA algorithm and MPCC active set algorithm.

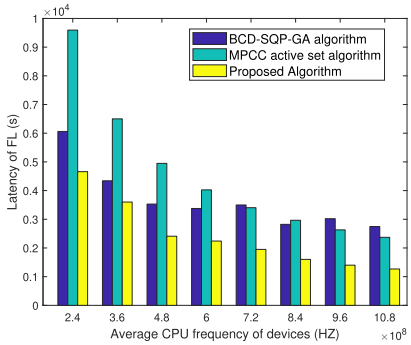

Fig. 9. Impact of the average CPU frequency of CDs.

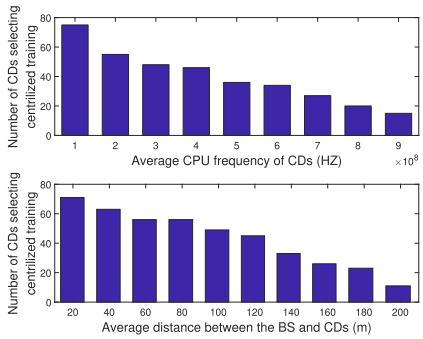

Fig. 10. Evaluating the number of CDs selecting centralized training.

Fig. 9 demonstrates the performance of the proposed algorithm versus the CDs' average CPU frequency. The results in Fig. 9 show that the latency of FL is decreasing with the average CPU frequency of CDs. The results in Fig. 9 also show that our proposed algorithm can outperform both BCD-SQP-GA algorithm and MPCC active set algorithm. According to the results shown in Fig. 3 and Fig. 9, our algorithm can on average outperform BCD-SQP-GA algorithm and MPCC active set algorithm by 52% and 63%, respectively.

Fig. 10 demonstrates the number of CDs selecting centralized training versus the average computing capacity of all CDs and the average distance between the BS and the CDs. Here, we set $K = 100$. Considering the same transmit-power and energy budget of all CDs, we set $E_k = 1 \times 10^4$ J, $p_k = 0.1$ W, $\forall k \in \mathcal{K}$. When evaluating the impact of the average computing capacity of CDs, we set the same value of the distances between the CDs and the BS ($d_k = 140$ m, $\forall k \in \mathcal{K}$). When evaluating the impact of the average distance between the BS and the CDs, we set the same computing capacity of all CDs ($f_k = 4 \times 10^8$ HZ, $\forall k \in \mathcal{K}$). The results in Fig. 10 show that the number of CDs selecting centralized training is decreasing with the average computing capacity of CDs and the average distance between the BS and the CDs.

C. Evaluation of the Proposed FL With Hybrid Training

We evaluate the performance of the proposed FL scheme with the hybrid training in comparison with three benchmark schemes as follows.

- Conventional FL (i.e., pure local training) scheme, in which all CDs are privacy-sensitive and preform local learning.

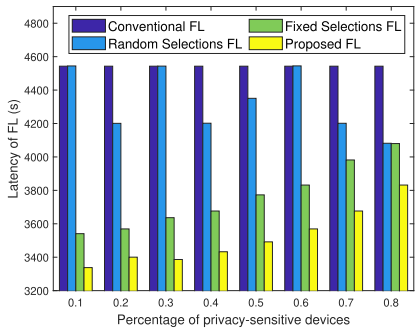

Fig. 11. Impact of the percentage of privacy-sensitive CDs.

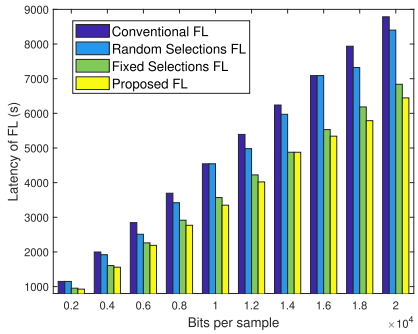

Fig. 12. Impact of the number of bits per sample.

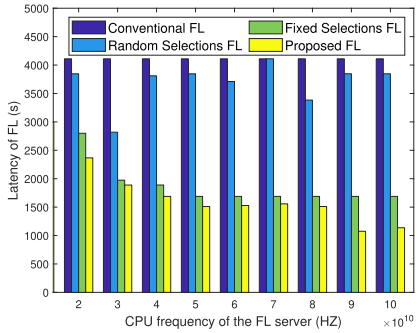

Fig. 13. Impact of the FL server's CPU frequency.

- Random selection FL scheme, in which the privacy-insensitive CDs randomly select to perform local training or centralized training.
- Fixed selection FL scheme, in which a fixed group of the CDs are selected to perform centralized training while others are fixed to perform local training.

Fig. 11 demonstrates the latency of the proposed FL scheme with the hybrid training versus the percentage of privacy-sensitive CDs. The results in Fig. 11 show that the latency of the proposed FL is increasing with the percentage of privacy-sensitive CDs. The results in Fig. 11 also show that our proposed FL scheme can outperform the benchmark schemes and the performance gain is decreasing with the percentage of privacy-sensitive CDs.

Fig. 12 demonstrates the latency of the proposed FL scheme with the hybrid training versus the number of bits per sample. The results in Fig. 12 show that the latency of the proposed FL is increasing with the number of bits per sample. The results in Fig. 12 also show that our proposed FL scheme can outperform the other benchmark schemes.

Fig. 13 demonstrates the latency of the proposed FL scheme with the hybrid training versus the FL server's CPU frequency. The results in Fig. 13 show that the latency of the proposed FL is decreasing with the FL server's CPU frequency. The results in Fig. 13 again validate that our proposed FL scheme can outperform the other benchmark schemes. According to the results shown in Figs. 11, 12, and 13, our proposed FL framework can save the latency by more than 30% compared to both conventional FL scheme and the FL with the randomized selection of training types.

VI. Conclusion

In this paper, we have proposed an FL framework with hybrid training, in which each privacy insensitive client-device has option to perform either the local model training or the centralized training. We have analyzed the convergence feature under this framework, and formulated a joint optimization of the client-devices' selections of training types, the FL configurations (i.e., the numbers of local iterations and global iterations) and the bandwidth allocations for different client-devices. Despite the non-convexity of the joint optimization problem, we have exploited its layered structure and proposed an efficient algorithm to solve it. Numerical results have been provided to validate the accuracy and efficiency of our proposed algorithm and demonstrate the advantage the FL framework. The results demonstrate that our proposed FL framework can save the latency by more than 30% compared to both conventional FL scheme and the FL with the randomized selection of training types. Meanwhile, our proposed algorithm can outperform two benchmark algorithms (i.e., BCD-SQP-GA algorithm and MPCC active set algorithm) by more than 50%. For the future work, we will investigate the scenario of multiple FL-servers in which different client-devices can flexibly select different servers for their model aggregations.

Appendix A
Proof of Theorem 1

To minimize the overall loss function and solve eq. (14), similar to [26], we adopt distributed approximate Newton algorithm (DANE). For DANE, the CDs and the FL server solve the optimization problem as

$$\min_{\mathbf{v}_k} \Omega_k(\mathbf{g}^n + \mathbf{v}_k) = F_k(\mathbf{g}^n + \mathbf{v}_k) - (\bigtriangledown F_k(\mathbf{g}^n) - \beta \bigtriangledown F(\mathbf{g}^n))^T \mathbf{v}_k, \quad k = 0,1,2,\ldots K, \tag{52}$$

where β is a constant and $k = 0$ corresponds to the centralized training on the FL server. $\mathbf{v}_k$ denotes the difference between the global FL model and the model of CD k or the FL server. $\varpi_k = \mathbf{g}^n + \mathbf{v}_k$. To solve the optimization problem (52), we use the gradient method as

$$\mathbf{v}_k^{m+1} = \mathbf{v}_k^m - \delta \bigtriangledown \Omega_k(\mathbf{g}^n + \mathbf{v}_k^m), \tag{53}$$

where δ denotes the learning rate and $\mathbf{v}_k^m$ denotes the value of $\mathbf{v}_k$ at the m-th local iteration.

To analyze the convergence feature of FL, similar to [25], we assume that $F_k(\varpi)$ is L-Lipschitz and γ-strongly convex as

$$\gamma \mathbf{I} \leq \bigtriangledown^2 F_k(\varpi) \leq L\mathbf{I}, \tag{54}$$

which avoids a significant variation on the gradient of the loss function.

In our paper, we consider the mini-batch SGD. We use $\bigtriangledown\tilde{\Omega}_k(\varpi)$ to denote the stochastic gradient computed on the mini-batch. Similar to [24], we assume that the stochastic gradient's variance is bounded as

$$\mathbb{E}\left(\left\|\bigtriangledown\tilde{\Omega}_k(\varpi)-\bigtriangledown\Omega_k(\varpi)\right\|^2\right)\leq\mathbb{E}(\frac{\rho_1}{\vartheta D_k}\|\bigtriangledown\Omega_k(\varpi)\|^2)+\frac{\sigma_F^2}{\vartheta D_k},\quad k=0,1,2,\ldots K, \tag{55}$$

where ρ_1 and σ_F^2 are constant, and $\mathbb{E}$ means the expectation corresponding to the stochastic batch. In (55), $D_0=\sum_{k\in\mathcal{K}}x_kD_k$.

Similar to [25], the degree of non-IID and the heterogeneity of the data distribution can be quantified by Γ as

$$\Gamma=F^*-\frac{1}{D}\left(\sum_{k\in\mathcal{K}}(1-x_k)D_kF_k^*+F_0^*\sum_{k\in\mathcal{K}}x_kD_k\right), \tag{56}$$

where $D=\sum_{k\in\mathcal{K}}D_k$ denotes the total data samples of all CDs.

Considering the training loss decay on mini-batch, we obtain

$$\begin{aligned}
&\mathbb{E}(\Omega_k(\mathbf{g}^n+\mathbf{v}_k^{m+1})-\Omega_k(\mathbf{g}^n+\mathbf{v}_k^m))\\
&=\mathbb{E}\left(\Omega_k(\mathbf{g}^n+\mathbf{v}_k^m-\delta\bigtriangledown\tilde{\Omega}_k(\mathbf{g}^n+\mathbf{v}_k^m))-\Omega_k\left(\mathbf{g}^n+\mathbf{v}_k^m\right)\right)\\
&\leq\mathbb{E}\left(\frac{L\delta^2}{2}\left\|\bigtriangledown\tilde{\Omega}_k(\mathbf{g}^n+\mathbf{v}_k^m)\right\|^2-\delta\bigtriangledown\Omega_k(\mathbf{g}^n+\mathbf{v}_k^m)^T\right.\\
&\quad\left.\times\bigtriangledown\tilde{\Omega}_k(\mathbf{g}^n+\mathbf{v}_k^m)\right)\\
&=\mathbb{E}(\frac{\delta}{2}(\left\|\bigtriangledown\Omega_k(\mathbf{g}^n+\mathbf{v}_k^m)-\bigtriangledown\tilde{\Omega}_k(\mathbf{g}^n+\mathbf{v}_k^m)\right\|^2\\
&\quad-\|\bigtriangledown\Omega_k(\mathbf{g}^n+\mathbf{v}_k^m)\|^2\\
&\quad-\left\|\bigtriangledown\tilde{\Omega}_k(\mathbf{g}^n+\mathbf{v}_k^m)\right\|^2)+\frac{L\delta^2}{2}\left\|\bigtriangledown\tilde{\Omega}_k(\mathbf{g}^n+\mathbf{v}_k^m)\right\|^2)\\
&\overset{①}{=}\frac{L\delta^2}{2}\mathbb{E}\left(\left\|\bigtriangledown\Omega_k(\mathbf{g}^n+\mathbf{v}_k^m)-\bigtriangledown\tilde{\Omega}_k(\mathbf{g}^n+\mathbf{v}_k^m)\right\|^2\right)\\
&\quad-\left(\delta-\frac{L\delta^2}{2}\right)\mathbb{E}(\|\bigtriangledown\Omega_k(\mathbf{g}^n+\mathbf{v}_k^m)\|^2)\\
&\leq-\delta\left(1-\frac{L\delta}{2}-\frac{L\delta}{2}\frac{\rho_1}{\vartheta D_k}\right)\mathbb{E}(\|\bigtriangledown\Omega_k(\mathbf{g}^n+\mathbf{v}_k^m)\|^2)+\frac{L\delta^2\sigma_F^2}{2\vartheta D_k}.
\end{aligned} \tag{57}$$

① is due to $\mathbb{E}(\|\bigtriangledown\tilde{\Omega}_k(\mathbf{g}^n+\mathbf{v}_k^m)-\mathbb{E}(\bigtriangledown\tilde{\Omega}_k(\mathbf{g}^n+\mathbf{v}_k^m))\|^2)=\mathbb{E}(\|\bigtriangledown\tilde{\Omega}_k(\mathbf{g}^n+\mathbf{v}_k^m)\|^2)-\mathbb{E}(\|\bigtriangledown\tilde{\Omega}_k(\mathbf{g}^n+\mathbf{v}_k^m)\|)^2$, where $\mathbb{E}(\bigtriangledown\tilde{\Omega}_k(\mathbf{g}^n+\mathbf{v}_k^m))=\mathbb{E}(\bigtriangledown\Omega_k(\mathbf{g}^n+\mathbf{v}_k^m))$ since we assume that the data on the k-th CD is unbiased.

From (52) and (54), after some mathematical manipulations, we can conclude that Ω_k is also L-Lipschitz and γ strong. Similar to [25], the item in (57) can be bounded as

$$\mathbb{E}(\|\bigtriangledown\Omega_k(\mathbf{g}^n+\mathbf{v}_k^m)\|^2)\geq\mathbb{E}(\gamma(\Omega_k(\mathbf{g}^n+\mathbf{v}_k^m)-\Omega_k(\mathbf{g}^n+\mathbf{v}_k^*))). \tag{58}$$

By substituting (58) into (57), and choosing $\delta<\frac{2\vartheta\min\{D_k\}}{L(\vartheta\min\{D_k\}+\rho_1)}$ which implies $1-\frac{L\delta}{2}-\frac{L\delta}{2}\frac{\rho_1}{\vartheta D_k}>0$, we obtain

$$\begin{aligned}
&\mathbb{E}(\Omega_k(\mathbf{g}^n+\mathbf{v}_k^{m+1})-\Omega_k(\mathbf{g}^n+\mathbf{v}_k^m))\\
&\leq-\delta\gamma\left(1-\frac{L\delta}{2}-\frac{L\delta}{2}\frac{\rho_1}{\vartheta D_k}\right)\mathbb{E}(\Omega_k(\mathbf{g}^n)-\Omega_k(\mathbf{g}^n+\mathbf{v}_k^*))\\
&\quad+\frac{L\delta^2\sigma_F^2}{2\vartheta D_k}.
\end{aligned} \tag{59}$$

By subtracting $\Omega_k(\mathbf{g}^n+\mathbf{v}_k^*)$ at the left and right sides of (59) simultaneously, we obtain

$$\begin{aligned}
&\mathbb{E}(\Omega_k(\mathbf{g}^n+\mathbf{v}_k^{m+1})-\Omega_k(\mathbf{g}^n+\mathbf{v}_k^*))\\
&=\frac{L\delta^2}{2}\frac{\sigma_F^2}{\vartheta D_k}+\left(1-\delta\gamma\left(1-\frac{L\delta}{2}-\frac{L\delta}{2}\frac{\rho_1}{\vartheta D_k}\right)\right)\\
&\quad\mathbb{E}\left(\Omega_k\left(\mathbf{g}^n+\mathbf{v}_k^m\right)-\Omega_k\left(\mathbf{g}^n+\mathbf{v}_k^*\right)\right)\\
&\leq\left(1-\delta\gamma\left(1-\frac{L\delta}{2}-\frac{L\delta}{2}\frac{\rho_1}{\vartheta D_k}\right)\right)^{m+1}\mathbb{E}(\Omega_k(\mathbf{g}^n)-\Omega_k(\mathbf{g}^n+\mathbf{v}_k^*))\\
&\quad+\frac{(1-\delta\gamma(1-\frac{L\delta}{2}-\frac{L\delta}{2}\frac{\rho_1}{\vartheta D_k}))^{m+1}}{\delta\gamma(1-\frac{L\delta}{2}-\frac{L\delta}{2}\frac{\rho_1}{\vartheta D_k})}\frac{\sigma_F^2}{\vartheta D_k}.
\end{aligned} \tag{60}$$

By introducing $\eta_k=1-\delta\gamma(1-\frac{L\delta}{2}-\frac{L\delta}{2}\frac{\rho_1}{\vartheta D_k})$, $k=0,1,2,\ldots K$, from (60), the local training loss decay rate at the k-th CD or the FL server ($k=0$) can be written as follows.

$$\begin{aligned}
&\mathbb{E}(\Omega_k(\mathbf{g}^n+\mathbf{v}_k^{m+1})-\Omega_k(\mathbf{g}^n+\mathbf{v}_k^*))\\
&\leq(\eta_k)^{m+1}\mathbb{E}(\Omega_k(\mathbf{g}^n)-\Omega_k(\mathbf{g}^n+\mathbf{v}_k^*))+\frac{(\eta_k)^{m+1}}{1-\eta_k}\frac{\sigma_F^2}{\vartheta D_k}.
\end{aligned}$$

Now, we consider the training loss of the global model. Since we do not use the relationship between the expectation of the mini-batch and that of the whole dataset in the following proof, for the sake of simplicity, we omit the symbol $\mathbb{E}$. The training loss of the global model at $(n+1)$-th round can be bounded as follows.

$$\begin{aligned}
&F(\mathbf{g}^{n+1})\\
&=F\left(\frac{1}{D}\sum_{k\in\mathcal{K}}(1-x_k)D_k\mathbf{g}_k^{n+1}+\frac{1}{D}\sum_{k\in\mathcal{K}}x_kD_k\mathbf{g}_0^{n+1}\right)\\
&\overset{②}{=}F\left(\mathbf{g}^n+\frac{1}{D}\sum_{k\in\mathcal{K}}(1-x_k)D_k\mathbf{v}_k^m+\frac{1}{D}\sum_{k\in\mathcal{K}}x_kD_k\mathbf{v}_0^m\right)\\
&\leq\underbrace{\frac{1}{D}\bigtriangledown F(\mathbf{g}^n)\sum_{k\in\mathcal{K}}(1-x_k)D_k\mathbf{v}_k^m}_{A_1}+\underbrace{\frac{1}{D}\bigtriangledown F(g^n)\mathbf{v}_0^m\sum_{k\in\mathcal{K}}x_kD_k}_{A_2}
\end{aligned}$$

$$+\underbrace{\frac{L}{2}\left\|\frac{1}{D}\sum_{k\in\mathcal{K}}(1-x_k)D_k\mathbf{v}_k^m+\frac{1}{D}\sum_{k\in\mathcal{K}}x_kD_k\mathbf{v}_0^m\right\|^2}_{A_3}+F(\mathbf{g}^n).$$

② comes from that $\mathbf{g}^{n+1}=\mathbf{g}^n+\frac{1}{D}\sum_{k\in\mathcal{K}}D_k\mathbf{v}_k^m$.

Considering the item A_1, we obtain

$$\begin{aligned}A_1&=\frac{1}{D\beta}\sum_{k\in\mathcal{K}}(1-x_k)\\&\quad\times D_k(\Omega_k(\mathbf{g}^n+\mathbf{v}_k^m)-F_k(\mathbf{g}^n+\mathbf{v}_k^m)+\bigtriangledown F_k(\mathbf{g}^n)\mathbf{v}_k^m)\\&\le\frac{1}{D\beta}\sum_{k\in\mathcal{K}}((1-x_k)D_k(\underbrace{\Omega_k(\mathbf{g}^n+\mathbf{v}_k^m)-F_k(\mathbf{g}^n)}_{A_4}-\frac{\gamma}{2}\|\mathbf{v}_k^m\|^2).\end{aligned}$$

Similarly, considering the item A_2, we obtain

$$A_2\le\frac{1}{D\beta}\sum_{k\in\mathcal{K}}x_kD_k(\underbrace{\Omega_0(\mathbf{g}^n+\mathbf{v}_k^m)-F_0(\mathbf{g}^n)}_{A_5}-\frac{\gamma}{2}\|\mathbf{v}_0^m\|^2).$$

We analyze the item A_4 in A_1 and obtain

$$\begin{aligned}A_4&=\Omega_k(\mathbf{g}^n+\mathbf{v}_k^m)-\Omega_k(\mathbf{g}^n+\mathbf{v}_k^*)-(\Omega_k(\mathbf{g}^n)-\Omega_k(\mathbf{g}^n+\mathbf{v}_k^*))\\&\le((\eta_k)^m-1)(\Omega_k(\mathbf{g}^n)-\Omega_k(\mathbf{g}^n+\mathbf{v}_k^*))+\frac{1-(\eta_k)^m}{1-\eta_k}\frac{\sigma_F^2}{\vartheta D_k}\\&=((\eta_k)^m-1)(F_k(\mathbf{g}^n)-F_k(\mathbf{g}^n+\mathbf{v}_k^*)\\&\quad+(\bigtriangledown F_k(\mathbf{g}^n)-\beta\bigtriangledown F(\mathbf{g}^n))^T\mathbf{v}_k^m+\frac{1-(\eta_k)^m}{1-\eta_k}\frac{\sigma_F^2}{\vartheta D_k}\\&=((\eta_k)^m-1)(F_k(\mathbf{g}^n)-F_k(\mathbf{g}^n+\mathbf{v}_k^*)+\bigtriangledown F_k(\mathbf{g}^n+\mathbf{v}_k^*)\mathbf{v}_k^*)\\&\quad+\frac{1-(\eta_k)^m}{1-\eta_k}\frac{\sigma_F^2}{\vartheta D_k}\\&\le((\eta_k)^m-1)\frac{\gamma}{2}\|\mathbf{v}_k^*\|^2+\frac{1-(\eta_k)^m}{1-\eta_k}\frac{\sigma_F^2}{\vartheta D_k}.\end{aligned}$$

Similarly to above analysis, the item A_5 can be bounded as

$$A_5\le((\eta_0)^m-1)\frac{\gamma}{2}\|\mathbf{v}_0^*\|^2+\frac{1-(\eta_0)^m}{1-\eta_0}\frac{\sigma_F^2}{\vartheta\sum_{k\in\mathcal{K}}x_kD_k}.$$

Since the norm function is convex, from the property of convex function (for convex function $f(x)$, if $\sum a_i=1$, then there exists $f(\sum a_ix_i)\le a_i\sum f(x_i)$), we obtain

$$A_3\le\frac{L}{2D}\sum_{k\in\mathcal{K}}(1-x_k)D_k\|\mathbf{v}_k^m\|^2+\frac{L}{2D}\sum_{k\in\mathcal{K}}x_kD_k\|\mathbf{v}_0^m\|^2.$$

Choosing $\beta>\frac{L}{\gamma}$, we obtain $\frac{1}{2D}(L-\frac{\gamma}{\beta})\sum_{k\in\mathcal{K}}x_kD_k\|\mathbf{v}_0^m\|^2<0$ and $\frac{1}{2D}\sum_{k\in\mathcal{K}}(1-x_k)D_k(L-\frac{\gamma}{\beta})\|\mathbf{v}_k^m\|^2<0$. Then, combing the above analysis on A_1,A_2,A_3,A_4,A_5, we obtain

$$\begin{aligned}&F(\mathbf{g}^{n+1})-F(\mathbf{g}^n)\\&\le\underbrace{-\frac{\gamma}{2D\beta}\sum_{k\in\mathcal{K}}((1-x_k)D_k(1-(\eta_k)^m)\|\mathbf{v}_k^*\|^2)}_{A_6}+\frac{1-(\eta_0)^m}{1-\eta_0}\frac{\sigma_F^2}{\vartheta}\\&\underbrace{-\frac{\gamma\|v_0^*\|^2}{2D\beta}(1-(\eta_0)^m)\sum_{k\in\mathcal{K}}x_kD_k}_{A_7}+\sum_{k\in\mathcal{K}}(1-x_k)\frac{1-(\eta_k)^m}{1-\eta_k}\frac{\sigma_F^2}{\vartheta}.\end{aligned}$$

For the item A_6, according to the Lagrange median theorem, there always exists a ϖ such that $\bigtriangledown F_k(\mathbf{g}^n+\mathbf{v}_k)-\bigtriangledown F_k(\mathbf{g}^n)=\bigtriangledown^2F_k(\varpi)\mathbf{v}_k$. Combining the L-Lipschitz and γ-strong assumption, we obtain $\|\mathbf{v}_k^*\|^2\ge\frac{1}{L^2}\|\bigtriangledown F_k(\mathbf{g}^n+\mathbf{v}_k^*)-\bigtriangledown F_k(\mathbf{g}^n)\|^2$. Form $\bigtriangledown\Omega_k(\mathbf{g}^n+\mathbf{v}_k^*)=\bigtriangledown F_k(\mathbf{g}^n+\mathbf{v}_k^*)-(\bigtriangledown F_k(\mathbf{g}^n)-\beta\bigtriangledown F(\mathbf{g}^n))=0$, we obtain $\beta\bigtriangledown F(\mathbf{g}^n)=\bigtriangledown F_k(\mathbf{g}^n)-\bigtriangledown F_k(\mathbf{g}^n+\mathbf{v}_k^*)$. Then, the item A_6+A_7 can be bounded as

$$\begin{aligned}A_6+A_7&\le-\frac{\beta\gamma}{2DL^2}\sum_{k\in\mathcal{K}}(1-x_k)D_k(1-(\eta_k)^m)\left\|\bigtriangledown F(\mathbf{g}^n)\right\|^2\\&\quad-\frac{\beta\gamma}{2DL^2}(1-(\eta_0)^m)\|\bigtriangledown F(\mathbf{g}^n)\|^2\sum_{k\in\mathcal{K}}x_kD_k\\&\le-\frac{\beta\gamma^2}{2DL^2}(\sum_{k\in\mathcal{K}}(1-x_k)(1-(\eta_k)^m)D_k\\&\quad+(1-(\eta_0)^m)\sum_{k\in\mathcal{K}}x_kD_k)(F(\mathbf{g}^n)-F(\mathbf{g}^*)).\end{aligned}\tag{61}$$

We introduce two variables as

$$\begin{aligned}C_1&=\frac{\beta\gamma^2}{2DL^2}(\sum_{k\in\mathcal{K}}(1-x_k)(1-(\eta_k)^m)D_k\\&\quad+(1-(\eta_0)^m)\sum_{k\in\mathcal{K}}x_kD_k)),\\C_2&=\sum_{k\in\mathcal{K}}((1-x_k)\frac{1-(\eta_k)^m}{1-\eta_k}\frac{\sigma_F^2}{\vartheta}+\frac{1-(\eta_0)^m}{1-\eta_0}\frac{\sigma_F^2}{\vartheta}.\end{aligned}$$

With these newly introduced variables, we obtain

$$F(\mathbf{g}^{n+1})\le F(\mathbf{g}^n)-C_1(F(\mathbf{g}^n)-F(\mathbf{g}^*))+C_2.$$

With the non-IID assumption in (56), we obtain

$$F(\mathbf{g}^{n+1})\le F(\mathbf{g}^n)-C_1(F(\mathbf{g}^n)-(F^*-\Gamma))+C_2.\tag{62}$$

Then, from (62), the convergence feature of the proposed FL can be expressed as

$$\begin{aligned}&F(\mathbf{g}^{n+1})-F^*\le(1-C_1)(F(\mathbf{g}^n)-F^*)-C_1\Gamma+C_2\\&\le(1-C_1)^{n+1}(F(\mathbf{g}^0)-F^*)+(C_2-C_1\Gamma)((1-C_1)^n\\&\quad+\cdots+(1-C_1)+1)\\&=(1-C_1)^{n+1}(F(\mathbf{g}^0)-F^*)+(C_2-C_1\Gamma)\frac{1-(1-C_1)^{n+1}}{C_1}.\end{aligned}$$

This completes our proof.

APPENDIX B
PROOF OF LEMMA 1

The first-order derivative of R_k^{up} can be expressed as

$$\frac{\mathrm{d}\,R_k^{\text{up}}}{\mathrm{d}\,b_k}=\log_2\left(1+\frac{p_kh_k}{\sigma b_k}\right)-\frac{p_kh_k}{(\sigma b_k+p_kh_k)\ln 2}.\tag{63}$$

From the inequality that $\ln(1+x) > \frac{x}{1+x}$ for $x > 0$, there exists

$$\log_2\left(1+\frac{p_k h_k}{\sigma b_k}\right) > \frac{p_k h_k}{(\sigma b_k + p_k h_k)\ln 2}. \tag{64}$$

By substituting (64) into (63), we obtain $\frac{\mathrm{d}\, R_k^{\mathrm{up}}}{\mathrm{d}\, b_k} > 0$. Thus, the value of R_k^{up} is monotonically increasing with b_k, which implies that y_k^{loc} and y_k^{cen} are monotonically decreasing with b_k.

Similarly, we can conclude that the value of y_k^{do} is monotonically decreasing with b_k. From the expression of $\hat{T}_k$ as (29), since t_k^{loc} and t^{ser} are fixed, we can obtain the conclusion in Lemma 1. This completes our proof.

Appendix C
Proof of Lemma 2

Based on the modeling of the energy consumption in (7), (9) and (13), E_k can be rewritten as

$$E_k = \begin{cases} \kappa\iota D_k \zeta_k m f_k^2 + p_k y_k^{\mathrm{loc}}, \; x_k = 0, \\ p_k y_k^{\mathrm{cen}}, \; x_k = 1. \end{cases} \tag{65}$$

In (65), all items are fixed except y_k^{loc} and y_k^{cen}. In the proof in Lemma 1, we conclude that y_k^{loc} and y_k^{cen} are monotonically decreasing with b_k. Thus, we get the conclusion in Lemma 2 and complete the proof.

Appendix D
Proof of Lemma 3

The second-order derivative of R_k^{up} can be expressed as

$$\frac{\mathrm{d}^2 R_k^{\mathrm{up}}}{\mathrm{d}\, b_k^2} = \frac{p_k h_k(-(p_k h_k)^2 - \sigma p_k h_k b_k))}{(\sigma(b_k)^2 + p_k h_k b_k)(\sigma b_k + p_k h_k)^2 \ln 2} < 0,$$

which implies that R_k^{up} is concave. Since $y_k^{\mathrm{loc}} = (1 - x_k)\frac{S}{R_k^{\mathrm{up}}}$, $\forall k \in \mathcal{K}$, according to the operation rules of preserving convexity for composition function, we conclude that y_k^{loc} is convex. Similarly, we can conclude that the values of y_k^{do} and y_k^{cen} are monotonically decreasing with b_k. From (29) we know that $\hat{T}_k$ is convex, which leads constraint (31) convex. From (65), E_k is convex, which leads the energy budget constraint (23) convex. Constraint (24) is linear. Thus, Problem (OLM-BOT) is a convex problem. This completes the proof.

References

[1] W. Xu, Z. Yang, D. W. K. Ng, M. Levorato, and Y. C. Eldar, "Edge learning for B5G networks with distributed signal processing: Semantic communication, edge computing, and wireless sensing," 2022, *arXiv:2206.00422*.

[2] Z. Yang, M. Chen, K.-K. Wong, H. V. Poor, and S. Cui, "Federated learning for 6G: Applications, challenges, and opportunities," *Engineering*, vol. 8, pp. 33–41, 2022.

[3] Y. Luo, J. Xu, W. Xu, and K. Wang, "Sliding differential evolution scheduling for federated learning in bandwidth-limited networks," *IEEE Commun. Lett.*, vol. 25, no. 2, pp. 503–507, Feb. 2021.

[4] S. Wang, Y. Hong, R. Wang, Q. Hao, Y.-C. Wu, and D. W. K. Ng, "Edge federated learning via unit-modulus over-the-air computation," *IEEE Trans. Commun.*, vol. 70, no. 5, pp. 3141–3156, May 2022.

[5] P. Tehrani, F. Restuccia, and M. Levorato, "Federated deep reinforcement learning for the distributed control of nextG wireless networks," in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw.*, 2021, pp. 248–253.

[6] W. Ni, Y. Liu, Z. Yang, H. Tian, and X. Shen, "Federated learning in multi-RIS-aided systems," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9608–9624, Jun. 2022.

[7] H. Yang et al., "Lead federated neuromorphic learning for wireless edge artificial intelligence," *Nature Commun.*, vol. 13, no. 1, pp. 1–12, 2022.

[8] Y. Wu, Y. Song, T. Wang, L. Qian, and T. Q. S. Quek, "Non-orthogonal multiple access assisted federated learning via wireless power transfer: A cost-efficient approach," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2853–2869, Apr. 2022.

[9] J. Liu et al., "Communication-efficient asynchronous federated learning in resource-constrained edge computing," *Comput. Netw.*, vol. 199, 2021, Art. no. 108429.

[10] Y. Deng et al., "Auction: Automated and quality-aware client selection framework for efficient federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 8, pp. 1996–2009, Aug. 2022.

[11] H. Xiao, J. Zhao, Q. Pei, J. Feng, L. Liu, and W. Shi, "Vehicle selection and resource optimization for federated learning in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11073–11087, Aug. 2022.

[12] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[13] Y. Li, Y. Wu, Y. Song, L. Qian, and W. Jia, "Dynamic user-scheduling and power allocation for SWIPT aided federated learning: A deep learning approach," *IEEE Trans. Mobile Comput.*, early access, Aug. 25, 2022, doi: 10.1109/TMC.2022.3201622.

[14] D.-J. Han, M. Choi, J. Park, and J. Moon, "FedMes: Speeding up federated learning with multiple edge servers," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3870–3885, Dec. 2021.

[15] W. Y. B. Lim, J. S. Ng, Z. Xiong, D. Niyato, C. Miao, and D. I. Kim, "Dynamic edge association and resource allocation in self-organizing hierarchical federated learning networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3640–3653, Dec. 2021.

[16] B. Xu, W. Xia, W. Wen, P. Liu, H. Zhao, and H. Zhu, "Adaptive hierarchical federated learning over wireless networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 2070–2083, Feb. 2022.

[17] S. Luo, X. Chen, Q. Wu, Z. Zhou, and S. Yu, "HFEL: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6535–6548, Oct. 2020.

[18] W. Wen, Z. Chen, H. H. Yang, W. Xia, and T. Q. S. Quek, "Joint scheduling and resource allocation for hierarchical federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 21, no. 8, pp. 5857–5872, Aug. 2022.

[19] J. S. Ng et al., "A hierarchical incentive design toward motivating participation in coded federated learning," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 359–375, Jan. 2022.

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

[21] Y. Xiao et al., "Fully decentralized federated learning-based on-board mission for UAV swarm system," *IEEE Commun. Lett.*, vol. 25, no. 10, pp. 3296–3300, Oct. 2021.

[22] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, Jun. 2020.

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

[24] F. Haddadpour and M. Mahdavi, "On the convergence of local descent methods in federated learning," 2019, *arXiv:1910.14425*.

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

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

[27] Y. J. Zhang, L. Qian, and J. Huang, "Monotonic optimization in communication and networking systems," *Found. Trends Netw.*, vol. 7, no. 1, pp. 1–75, 2013.

[28] S. Zhu, W. Xu, L. Fan, K. Wang, and G. K. Karagiannidis, "A novel cross entropy approach for offloading learning in mobile edge computing," *IEEE Wireless Commun. Lett.*, vol. 9, no. 3, pp. 402–405, Mar. 2020.

[29] X. Huang, W. Xu, G. Xie, S. Jin, and X. You, "Learning oriented cross-entropy approach to user association in load-balanced hetnet," *IEEE Wireless Commun. Lett.*, vol. 7, no. 6, pp. 1014–1017, Dec. 2018.

[30] X. Huang, R. Yu, D. Ye, L. Shu, and S. Xie, "Efficient workload allocation and user-centric utility maximization for task scheduling in collaborative vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3773–3787, Apr. 2021.
[31] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[32] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming for large-scale nonlinear optimization," *J. Comput. Appl. Math.*, vol. 124, no. 1/2, pp. 123–137, 2000.
[33] D. Goldberg, *Genetic Algorithms In Search, Optimization, and Machine Learning*. Massachusetts, USA: Addison-Wesley Pub. Co., 1989.
[34] Y. Li, T. Tan, and X. Li, "Convergence of a continuous approach for zero-one programming problems," *Appl. Math. Comput.*, vol. 217, no. 9, pp. 4691–4698, 2011.
[35] A. Kadrani, J. P. Dussault, and A. Benchakroun, "A globally convergent algorithm for MPCC," *EURO J. Comput. Optim.*, vol. 3, no. 3, pp. 263–296, 2015.

Ning Huang received the B.Sc. degree in electronic science and technology, and the M.S. degree in optical engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2009 and 2012, respectively. He is currently working toward the Ph.D. degree with the Department of Computer and Information Science, the University of Macau, Macau, China. His current research interests include intelligent reflecting surface, mobile edge computing, and federated learning.

Minghui Dai received the Ph.D. degree from Shanghai University, Shanghai, China, in 2021. He is currently a Postdoctoral Fellow with the State Key Laboratory of Internet of Things for Smart City, the University of Macau, Macau, China. His research interests include the general area of wireless network architecture and vehicular networks.

Yuan Wu (Senior Member, IEEE) received the Ph.D. degree in electronic and computer engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2010. He is currently an Associate Professor with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Macau, China, and also with the Department of Computer and Information Science, University of Macau. From 2016 to 2017, he was a Visiting Scholar with Department of Electrical and Computer Engineering, the University of Waterloo, Waterloo, ON, Canada. His research interests include resource management for wireless networks, green communications and computing, edge computing and edge intelligence, and energy informatics. He was the recipient of the Best Paper Award from the IEEE ICC'2016, WCSP'2016, IEEE TCGCC'2017, and IWCMC'2021. Dr. Wu is currently on the editorial board of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, and IEEE INTERNET OF THINGS JOURNAL.

Tony Q. S. Quek (Fellow, IEEE) received the B.E. and M.E. degrees in electrical and electronics engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1998 and 2000, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2008. He is currently the Cheng Tsang Man Chair Professor with Singapore University of Technology and Design (SUTD), Singapore. He is also the Director of the Future Communications R&D Programme, the Head of ISTD Pillar, and the Deputy Director of the SUTD-ZJU IDEA. His current research interests include wireless communications and networking, network intelligence, internet-of-things, URLLC, and 6G. He has been actively involved in organizing and chairing sessions, and has served as a Member of the Technical Program Committee as well as symposium chairs in a number of international conferences. He is currently an Area Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and an Elected Member of IEEE Signal Processing Society SPCOM Technical Committee. He was an Executive Editorial Committee Member for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, an Editor for IEEE TRANSACTIONS ON COMMUNICATIONS, and IEEE WIRELESS COMMUNICATIONS LETTERS. Dr. Quek was the recipient of the 2008 Philip Yeo Prize for Outstanding Achievement in Research, the 2012 IEEE William R. Bennett Prize, the 2015 SUTD Outstanding Education Awards – Excellence in Research, the 2016 IEEE Signal Processing Society Young Author Best Paper Award, the 2017 CTTC Early Achievement Award, the 2017 IEEE ComSoc AP Outstanding Paper Award, the 2020 IEEE Communications Society Young Author Best Paper Award, the 2020 IEEE Stephen O. Rice Prize, the 2020 Nokia Visiting Professor, and the 2016–2020 Clarivate Analytics Highly Cited Researcher. He is a Fellow of the Academy of Engineering, Singapore.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular networks. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society. Dr. Shen received the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory (CSIT) in 2021, the R.A. Fessenden Award in 2019 from IEEE, Canada, Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society (ComSoc), and Technical Recognition Award from Wireless Communications Technical Committee (2019) and AHSN Technical Committee (2013). He was also the recipient of the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. He was the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, IEEE Globecom'07, and the Chair for the IEEE ComSoc Technical Committee on Wireless Communications. He is the President of the IEEE ComSoc. He was the Vice President for Technical and Educational Activities, Vice President for Publications, Member-at-Large on the Board of Governors, Chair of the Distinguished Lecturer Selection Committee, and Member of IEEE Fellow Selection Committee of the ComSoc. He was the Editor-in-Chief of IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, and *IET Communications*.