Filling the Missing: Exploring Generative AI for Enhanced Federated Learning Over Heterogeneous Mobile Edge Devices

Peichun Li, Hanwen Zhang, Yuan Wu, *Senior Member, IEEE*, Liping Qian, *Senior Member, IEEE*, Rong Yu, *Member, IEEE*, Dusit Niyato, *Fellow, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Distributed Artificial Intelligence (AI) model training over mobile edge networks encounters significant challenges due to the data and resource heterogeneity of edge devices. The former hampers the convergence rate of the global model, while the latter diminishes the devices' resource utilization efficiency. In this paper, we propose a generative AI-empowered federated learning to address these challenges by leveraging the idea of FIlling the MIssing (FIMI) portion of local data. Specifically, FIMI can be considered as a resource-aware data augmentation method that effectively mitigates the data heterogeneity while ensuring efficient FL training. We first quantify the relationship between the training data amount and the learning performance. We then study the FIMI optimization problem with the objective of minimizing the device-side overall energy consumption subject to required learning performance constraints. The decomposition-based analysis and the cross-entropy searching method are leveraged to derive the solution, where each device is assigned suitable AI-synthetic data and resource utilization policy. Experiment results demonstrate that FIMI can save up to 50% of the device-side energy to achieve the target global test accuracy in comparison with the existing methods. Meanwhile, FIMI can significantly enhance the converged global accuracy under the non-independently-and-identically distribution (non-IID) data.**

***Index Terms*—Federated learning, generative AI, data compensation, resource management.**

Manuscript received 4 September 2023; revised 25 December 2023; accepted 23 February 2024. Date of publication 29 February 2024; date of current version 3 September 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62122069, Grant 62072490, and Grant 62071431, in part by the Science and Technology Development Fund of Macau SAR under Grant 0158/2022/A, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515011287, and in part by the Natural Sciences and Engineering Research Council of Canada. The work of Rong Yu was supported in part by the National Key R&D Program of China under Grant 2020YFB1807802, in part by the National Natural Science Foundation of China under Grant 61971148 and Grant U22A2054. The work of Dusit Niyato was supported in part by the National Research Foundation, Singapore, and Infocomm Media Development Authority under its Future Communications Research and Development Programme, DSO National Laboratories under the AI Singapore Programme (AISG) under Grant AISG2-RP-2020-019 and Grant FCP-ASTAR- TG-2022-003, and MOE Tier 1 under Grant RG87/22. Recommended for acceptance by J. Ren. *(Corresponding author: Yuan Wu.)*

Peichun Li and Hanwen Zhang are with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Macao 999078, China, and also with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: peichunli@um.edu.mo; hw.zhang@connect.um.edu.mo).

Yuan Wu is with the State Key Laboratory of Internet of Things for Smart City, Department of Computer and Information Science, University of Macau, Macau 999078, China, and also with Zhuhai UM Science and Technology Research Institute, Zhuhai 519031, China (e-mail: yuanwu@um.edu.mo).

Liping Qian is with the College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: lpqian@zjut.edu.cn).

Rong Yu is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China (e-mail: yurong@ieee.org).

Dusit Niyato is with the College of Computing and Data Science (CCDS), Nanyang Technological University, Singapore 639798 (e-mail: dniyato@ntu.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/TMC.2024.3371772

I. Introduction

Generative artificial intelligence (AI) has advanced significantly in content creation. Notably, vision models such as stable diffusion and Imagen have demonstrated their impressive capability in generating photo-realistic images [1], [2], [3]. These advancements have sparked new possibilities that generative models can play a significant role in enhancing the centralized training process. Specifically, the synthetic images produced by generative models can be effectively utilized to improve the performance of classification models [4]. In the realm of federated learning (FL), where isolated devices with limited storage collaborate in training a global model, data and resource heterogeneity have emerged as crucial challenges [5], [6], [7], [8], [9]. These factors lead to a slower convergence rate during the global training process. Consequently, an intriguing and pertinent question arises, i.e., can generative AI be harnessed to address these issues and improve the performance of FL ultimately?

In this paper, we first investigate how the distribution of federated local data impacts the training performance of FL. Specifically, we focus on the VGG9 model [10] with 20 Nvidia Jetson Nano devices on CIFAR100 dataset [11]. We regulate the local dataset into different degrees of Dirichlet distribution, leading to varying levels of non-independently-and-identically distribution (non-IID) data [12], [13]. As shown in the top subplot of Fig. 1, we observe that a well-distributed Dir(0.9) significantly improves convergence accuracy compared to Dir(0.3). Here, Dir(z) means the Dirichlet distribution with parameter z, and a larger z means a more homogeneous distribution of the samples (i.e., less non-IID distribution). The observation indicates that augmenting the data to eliminate the non-IID feature of local data is promising to enhance the learning performance

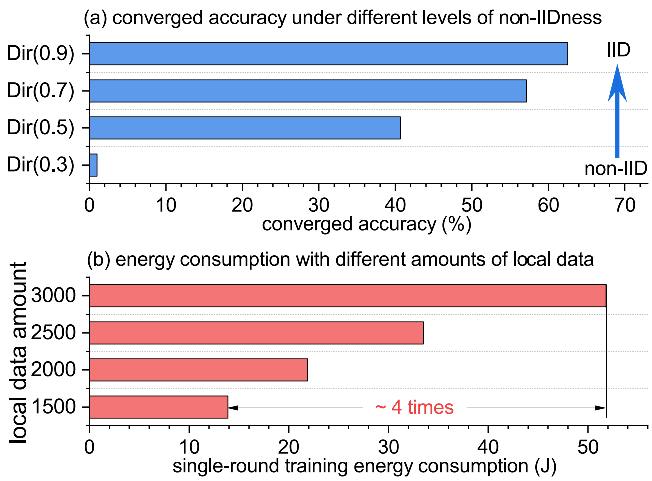


Fig. 1. Impacts of data distribution and data amount on the training performance.

of FL. However, the augmented training data introduces extra computation costs for edge devices. As shown in the bottom subplot of Fig. 1, doubling the amount of data results in about a fourfold increase in energy consumption under a fixed training latency.

The observations above offer insights for developing an efficient and effective generative AI-enhanced FL system. To address data heterogeneity, it is recommended to leverage generative AI to compensate for the missing portion of local data. This strategy can bridge the gaps in data distribution across devices and improve convergence accuracy. However, it is crucial to take into account the locally available resources and adapt the amount of synthetic data on different devices accordingly. The majority of current literature on generative AI-enhanced training methods tends to prioritize improving the model accuracy in a centralized training manner, disregarding the computation cost induced by the augmented training dataset [4], [14]. These approaches may shorten the battery lifetime and potentially degrade the user experience when integrated into the FL system directly. Therefore, we propose a resource-aware data augmentation framework, called FIlling the Missing (FIMI), which is designed for generative AI-enhanced FL systems.

The key idea behind FIMI is to leverage the concept of filling the missing portion of data for each device. Our goal is to improve the learning performance of FL by utilizing a pre-trained generative AI model to synthesize customized local data while ensuring the training efficiency of edge devices. To this end, we first outline the generic workflow of our generative AI-empowered FL system. We then empirically measure the relationship between the amount of training data and model learning error to quantitatively analyze the learning performance of FL. Furthermore, we propose an optimization problem for minimizing the device-side energy subject to the learning performance constraints. To solve this formulated problem, we introduce auxiliary variables and decompose the main problem into two sub-problems along with a top-layer searching problem. To find solutions for the sub-problems, the convex optimization methods are leveraged to obtain the optimal solution based on the theoretical analysis. For the top-layer problem, the continuous cross-entropy algorithm is employed to efficiently search the solution. After that, a data entropy maximization problem for mitigating the non-IID feature of local data is conducted to obtain the category-wise training sample amount on each device. In this way, each wireless edge device is assigned a personalized solution, where the data synthesis strategy and resource utilization policy are optimized to minimize the device-side energy consumption while improving the learning performance.

Our main contributions are summarized as follows.

- We propose a generative AI-enhanced FL framework called FIMI to leverage the pre-trained generative models for filling the missing portion of the local data. FIMI aims to re-balance the local data across different edge devices, ultimately enhancing the convergence rate of FL training through calibrated data distribution.
- By combining the theoretical and empirical analysis, we quantitatively establish the relationship between the training data amount and the learning performance of FL. This provides insights for striking the balance between training efficiency and learning performance.
- We formulate a joint optimization of data synthesis strategy and resource utilization policy, with the objective of minimizing the energy consumption of FIMI over wireless networks. We also propose an efficient algorithm to solve the formulated problem and determine the corresponding solution for each device.
- Extensive experiments on two types of real-world datasets demonstrate that our proposed FIMI clearly outperform the existing data augmentation FL methods in terms of the device-side energy utilization and the accuracy of the global model after convergence.

The remainder of this paper is organized as follows. Section II describes related studies. In Section III, we illustrate the system model of FIMI. The theoretical analysis and the corresponding solution are provided in Section IV. The experiment evaluations are presented in Section V. We finally conclude the paper and discuss the future directions in Section VI.

II. Related Work

1) Generative AI over edge networks: Generative AI on edge networks has attracted growing interest for its potential to provide low-latency AI generative content (AIGC) services [3]. Some studies focus on achieving efficient AIGC service by optimizing resource utilization [15] or by improving the freshness of the age of information [16]. In addition, generative AI has emerged as a promising technique to enhance wireless networks from the physical layer perspective [17]. Recently, the study in [18] incorporates the FL framework to enhance the training and inference process of the AIGC model. However, potential benefits of generative AI for FL training remain unexplored.

2) Data augmentation methods for FL: Data augmentation methods provide an effective approach to address the issue of non-IID data distribution in FL. The early study in [19] shows that exchanging a small proportion of local data can alleviate the non-IID issue. However, this approach may raise privacy

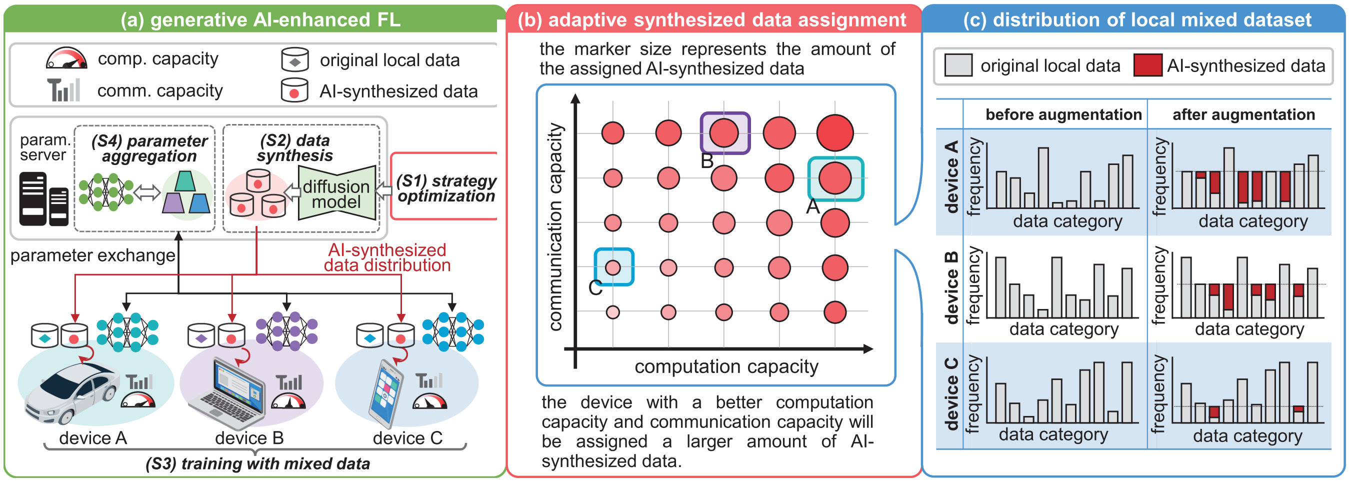


Fig. 2. **left**: FIMI over heterogeneous edge devices. **middle**: the amounts of synthetic data are personalized for different devices to match their locally available resources. **right**: each device optimizes its category-wise synthetic data amount to mitigate the locally non-IID feature of data.

concerns. In addition, semi-supervised FL methods utilize the unlabelled data to enhance learning performance [20], [21], [22]. Recently, generative adversarial networks (GAN) have emerged as a privacy-friendly technique for data compensation in FL [23], [24]. However, the GAN-based methods may fail to produce photo-realistic data, and most of the existing methods overlook the extra training cost induced by the augmented dataset.

3) Efficient training methods for FL: Efficient training methods for FL have been widely studied in recent years. One popular approach is the gradient compression, which reduces the amount of data by compressing the gradients [25], [26]. Another approach is topology-aware methods, such as hierarchical aggregation [27], [28] and device-to-device communication [29], which consider the network topology to optimize the communication pattern and reduce the communication overhead. Furthermore, energy-aware methods, such as wireless power transfer [30] and resource management [31], [32], [33], offer a green and sustainable framework for FL training. However, most of the existing studies do not consider the heterogeneity of data distribution among different participating devices, which can result in biased model updates.

III. System Model

In this section, we first introduce the structure of FIMI, and then present the mathematical modeling and optimization problems for the data augmentation-based FL training.

A. Structure of FIMI

We consider an FL scenario comprising a set of devices labeled as $\{1, 2, \ldots, I\}$. Each device, denoted as i, holds an original local dataset $\mathcal{D}_i^{\text{loc}}$, where $|\mathcal{D}_i^{\text{loc}}| = D_i^{\text{loc}}$ represents the number of local training samples. Our goal is to leverage a pre-trained generative model to create additional synthetic data $\mathcal{D}_i^{\text{gen}}$ for each device i, with $|\mathcal{D}_i^{\text{gen}}| = D_i^{\text{gen}}$. In traditional FL systems, device i solely relies on its local dataset $\mathcal{D}_i^{\text{loc}}$ for the local model update. Alternatively, the proposed FIMI adopts a mixed local dataset $\mathcal{D}_i^{\text{mix}} = \mathcal{D}_i^{\text{loc}} \cup \mathcal{D}_i^{\text{gen}}$ to accelerate the convergence.

We focus on the FL task for image classification across C categories (or label types). The data synthesis strategy of device i can be represented as $\{d_{i,c}^{\text{gen}}\}_{\forall c}, (c = 1, \ldots, C)$. This strategy should meet the constraint $\sum_c d_{i,c}^{\text{gen}} = D_i^{\text{gen}}$, where $d_{i,c}^{\text{gen}}$ denotes the amount of synthetic data for the c-th category on device i. Before the beginning of FL training, a collaborative optimization is undertaken between the server and edge devices to obtain the data synthesis strategy. Following this, the server will employ the corresponding label as input for the diffusion model to generate the required data. After the synthetic data is distributed to each device, the FL training with the mixed dataset is initiated. As shown in Fig. 2, the FIMI framework operates as follows.

- *(S1) Strategy optimization:* The server conducts a centralized optimization, as shown in the middle subplot of Fig. 2, to determine the synthetic data amount for each device. Each device then calculates the *category-wise* synthetic data amount $\{d_{i,c}^{\text{gen}}\}_{\forall c}, (c = 1, \ldots, C)$, as shown in the right subplot of Fig. 2, to rebalance the data distribution and mitigate the non-IID feature of local data.
- *(S2) Data synthesis:* Each device sends its data synthesis request, specifying the category-wise amount of synthetic data, to the server. The server infers the generative model and distributes the required synthetic data to the corresponding devices in parallel.
- *(S3) Training with mixed data:* The server distributes the global model to each device. Then, the local device performs the local training with the mixed local dataset, and then uploads its model update back to the server.
- *(S4) Parameter aggregation:* The server collects and aggregates the local updates from all devices to renew the global model. The weight coefficient of device i during aggregation is calculated as $\frac{D_i^{\text{loc}} + D_i^{\text{gen}}}{\sum_{j=1}^{I} D_j^{\text{loc}} + D_j^{\text{gen}}}$. The training process jumps to (S3) until the predetermined maximum global iterations are reached.

Since the servers are usually equipped with sufficient computational resources and resilient power supply, we do not account for the latency and energy consumption during the data synthesis process. Empirical computation and communication overheads for data synthesis and distribution will be illustrated in Section V-A.

B. Learning Performance Analysis

In this subsection, we quantitatively analyze the relationship between the number of local samples and the overall training performance. We first establish the mathematical formulation that links the amount of local data to the individual learning performance. We then determine the unknown parameters in the formulation through the empirical fitting. Finally, we investigate how the isolated local data from multiple devices can influence the global learning performance.

1) Local Data Versus Local Model Performance: Let δ_i denote the local learning error of device i when training on mixed data $\mathcal{D}_i^{\text{mix}}$. According to the previous studies in [34], [35], [36], the lower bound of the achievable local error can be modeled as the power-law function with respect to the training sample amount of $|\mathcal{D}_i^{\text{mix}}|$. The relationship between $D_i^{\text{loc}} + D_i^{\text{gen}}$ and δ_i can be expressed as

$$\delta_i = \alpha(D_i^{\text{loc}} + D_i^{\text{gen}})^{-\beta} - \gamma, \tag{1}$$

where α, β, and γ are three positive hyper-parameters that can be determined through experiments.

2) Parameter Fitting on Proxy Task: We next focus on determining the values of $\{\alpha, \beta, \gamma\}$. However, since the local dataset is not accessible to the server, direct parameter fitting on the target learning task by the server can raise privacy concerns. To address this, we propose an alternative approach by determining these parameters using publicly available datasets. Specifically, we provide the fitting results on two classification tasks representing varying complexities: CINIC10 [37] for a more intricate task and Fashion-MNIST (FMNIST) [38] for a simpler one. As shown in Fig. 3, the fitting results of two different datasets exhibit a similar pattern. These findings underscore the reliability and robustness of the model in capturing local learning error as defined in (1). It is noticed that this *one-time* parameter fitting process can be conducted offline by the server, and the results of fitting can be generalized to *many* learning tasks with different datasets. During the practice implementation, we can directly select the according fitting result for $\{\alpha, \beta, \gamma\}$ based on the complexity of the target tasks.

3) Global Data Versus Global Model Performance: After estimating the local learning performance $\{\delta_i\}_{\forall i}$, the average local training error $\overline{\delta}$ can be calculated as

$$\overline{\delta} = \frac{1}{I}\sum_{i=1}^{I} \delta_i. \tag{2}$$

Based on the theoretical analysis in [39], [40], when training with the average local training error $\overline{\delta}$, the number of global iterations N to reach a pre-determined global learning error Δ is bounded by

$$N = \frac{\zeta \ln(1/\Delta)}{1-\overline{\delta}}, \tag{3}$$

where ζ is a positive constant parameter. Furthermore, the relationship between the average local learning error $\overline{\delta}$ and the global learning error Δ can be captured by

$$\Delta = \exp\left(\frac{N(\overline{\delta}-1)}{\zeta}\right). \tag{4}$$

By combing (1), (2) and (4), we bridge the relationship between the amount of local mixed dataset and the global learning performance.

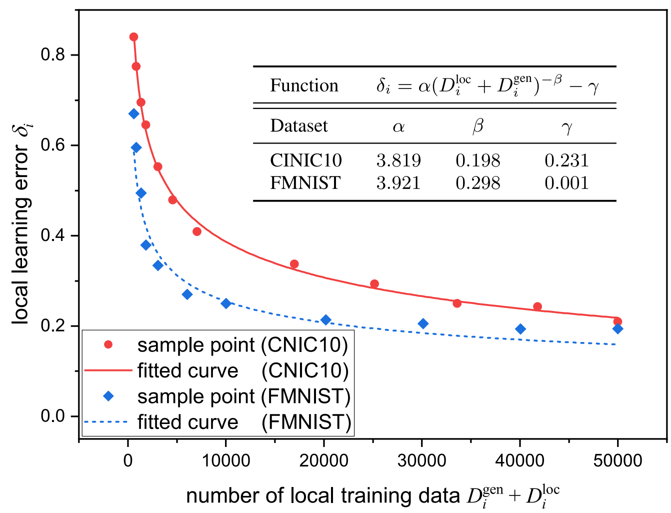


Fig. 3. Local learning error δ_i with respect to local training data amount.

C. FIMI Over Wireless Networks

1) Computation Model: Let ω denote the computation workload required for a single training sample and τ represent the local epoch. According to [31], [40], [41], the computation-related configurations of device i can be profiled by two parameters, including the computing frequency f_i and the hardware energy coefficient ϵ_i. Thus, the single-round energy consumption for the local model training at device i can be estimated by

$$E_i^{\text{cmp}} = \tau\epsilon_i\omega(D_i^{\text{loc}} + D_i^{\text{gen}})f_i^2. \tag{5}$$

Furthermore, the required latency for a single-round local model training of device i is

$$T_i^{\text{cmp}} = \frac{\tau\omega(D_i^{\text{loc}} + D_i^{\text{gen}})}{f_i}. \tag{6}$$

2) Communication Model: We next focus on the device-side uplink parameter transmission. We consider the frequency division multiple access (FDMA) scheme with a preset total bandwidth of B. For the device i allocated with a sub-bandwidth of b_i, its achievable transmitting rate for sending the local update to the server can be given by

$$r_i = b_i \log_2\left(1 + \frac{g_i P_i}{N_0 b_i}\right), \tag{7}$$

where g_i is the device-to-server channel gain, P_i is the transmitting power, and N_0 is the power spectral density of background Gaussian noise. Here, we utilize S to denote the data size of the model update. The single-round latency required for device i to send its model update to the server is estimated by

$$T_i^{\text{com}} = \frac{S}{r_i}. \tag{8}$$

Moreover, the single-round energy consumption for device i to send its model update to the server can be calculated by

$$E_i^{\text{com}} = \frac{SP_i}{r_i}. \tag{9}$$

3) Problem Formulation: Based on the above modeling, the energy consumption of all the devices for one-round training can be expressed as

$$E^{\text{round}} = \sum_{i=1}^{I} E_i^{\text{cmp}} + \sum_{i=1}^{I} E_i^{\text{com}}. \tag{10}$$

Meanwhile, the system-wise learning latency for one-round training is calculated by

$$T^{\text{round}} = \max_i \left\{ T_i^{\text{cmp}} + T_i^{\text{com}} \right\}. \tag{11}$$

To optimize the proposed FIMI, we investigate an energy minimization problem with a pre-determined learning performance constraint. Specifically, given multiple devices with heterogeneous resources in terms of computation, communication, and data distribution, our objective is to optimize the optimal data synthesis strategy (i.e., $\{D_i^{\text{gen}}\}, i = 1, \ldots, I$) and resource utilization policy (i.e., $\{b_i, f_i, P_i\}, i = 1, \ldots, I$) for each device. Mathematically, we aim at studying the following problem for each global round.

$$\begin{aligned}
&\text{(P1)} && \min E^{\text{round}} && (12)\\
&\text{subject to:} && \Delta \leq \Delta^{\max}, && (12a)\\
& && T^{\text{round}} \leq T^{\max}, && (12b)\\
& && 0 \leq D_i^{\text{gen}} \leq D^{\max}, \forall i, && (12c)\\
& && 0 \leq f_i \leq f_i^{\max}, \forall i, && (12d)\\
& && \sum_{i=1}^{I} b_i \leq B, && (12e)\\
& && 0 \leq P_i \leq P_i^{\max}, \forall i, && \\
&\text{variables:} && \{D_i^{\text{gen}}, f_i, b_i, P_i\}_{\forall i}, && (12f)
\end{aligned}$$

where $\Delta^{\max}$, $T^{\max}$, and $D^{\max}$ are the pre-determined maximum allowable global error, the maximum latency for single round learning, and the maximum amount of synthetic data for each device, respectively. Note that the data synthesis procedure is conducted by the server with powerful computing capabilities. Therefore, we solely focus on the device-side energy consumption in Problem (P1).

D. Analysis and Decomposition

Before proposing the algorithms to solve Problem (P1) in Section IV, we present the analysis and decomposition in this subsection. We first leverage two lemmas to simplify the main problem. After that, we decompose the simplified problem into three sub-problems.

To facilitate the analysis, we present the following lemmas.

Lemma 1: The equality in Constraint (12a) always holds when the optimal data synthesis strategy $\{D_i^{\text{gen},*}\}_{\forall i}$ is achieved.

Proof: The lemma can be proved by showing contradiction. Given the optimal data synthesis strategy $\{D_i^{\text{gen},*}\}_{\forall i}$, suppose that the achieved global learning error is less than the maximum allowable error, i.e., $\Delta(\{D_i^{\text{gen},*}\}_{\forall i}) < \Delta^{\max}$. Based on (1) and (4), the global learning error Δ decreases with D_i^{gen}. Therefore, a feasible solution can always be constructed by reducing the synthetic data of device i_0 from $D_{i_0}^{\text{gen},*}$ to $D_{i_0}^{\text{gen},\prime}$, such that $\Delta(\{D_i^{\text{gen},*}\}_{\forall i, i\neq i_0} \cup D_{i_0}^{\text{gen},\prime}) = \Delta^{\max}$. As E^{round} increases with D_i^{gen}, the energy consumption of the new solution $\{D_i^{\text{gen},*}\}_{\forall i, i\neq i_0} \cup D_{i_0}^{\text{gen},\prime}$ will be lower than that of the optimal one. This completes the proof. □

Lemma 2: The equality in Constraint (12b) always holds under the optimal resource utilization policy $\{b_i^*, f_i^*, P_i^*\}_{\forall i}$.

Proof: The lemma can be proved by showing contradiction. Suppose that under the optimal resource utilization policy $\{b_i^*, f_i^*, P_i^*\}_{\forall i}$, the single-round latency required is smaller than the maximum latency, i.e., $T^{\text{round}} < T^{\max}$. According to (6), the training latency increases with f_i. Therefore, a feasible solution can always be constructed by reducing the computing frequency of device i_0 from $f_{i_0}^*$ to $f_{i_0}^{\prime}$. Similar to the proof of Lemma 1, the new solution will consume less energy than the optimal one. This completes the proof. □

By combing (4) and the above two lemmas, Problem (P1) can be simplified as

$$\begin{aligned}
&\text{(P2)} && \min E^{\text{round}} && (13)\\
&\text{subject to:} && \sum_{i=1}^{I} \delta_i = I + \frac{\zeta I}{N} \ln\left(\Delta^{\max}\right), && (13a)\\
& && T_i^{\text{cmp}} + T_i^{\text{com}} = T^{\max}, && (13b)\\
& && \text{Constraints (12c)–(12f)}, && \\
&\text{variables:} && \{D_i^{\text{gen}}, f_i, b_i, P_i\}_{\forall i}. && (13c)
\end{aligned}$$

Moreover, by determining the latency for both local model training and local update transmission for each device (i.e., $\{T_i^{\text{cmp}}, T_i^{\text{com}}\}_{\forall i}$), Problem (P2) can be decomposed into two distinct sub-problems as described in the following.

The first sub-problem, denoted as Problem (P3), aims at minimizing all devices' energy consumption for their local model training, which is expressed as follows:

$$\text{(P3)} \quad \min_{\{D_i^{\text{gen}}, f_i\}_{\forall i}} \sum_{i=1}^{I} \tau \epsilon_i \omega (D_i^{\text{loc}} + D_i^{\text{gen}}) f_i^2 \tag{14}$$

$$\text{subject to:} \quad \frac{\tau\omega(D_i^{\text{loc}} + D_i^{\text{gen}})}{f_i} = T_i^{\text{cmp}}, \forall i, \tag{14a}$$

$$\text{Constraints (12c), (12d), and (13a).} \tag{14b}$$

The second sub-problem, denoted as Problem (P4), aims at minimizing all devices' energy consumption for uploading their locally trained models as follows:

$$\text{(P4)} \quad \min_{\{b_i, P_i\}_{\forall i}} \sum_{i=1}^{I} P_i T_i^{\text{com}} \tag{15}$$

$$\text{subject to:} \quad \frac{S}{b_i \log_2\left(1 + \frac{g_i P_i}{N_0 b_i}\right)} = T_i^{\text{com}}, \forall i, \tag{15a}$$

$$\text{Constraints (12e) and (12f).} \tag{15b}$$

After solving the above two sub-problems, we can obtain $\{D_i^{\text{gen}}, f_i, b_i, P_i\}_{\forall i}$ with given $\{T_i^{\text{cmp}}, T_i^{\text{com}}\}$. Then, we continue to determine the optimal $\{T_i^{\text{cmp},*}, T_i^{\text{com},*}\}_{\forall i}$ for each device. Instead of directly optimizing $\{T_i^{\text{cmp}}, T_i^{\text{com}}\}$, we introduce an auxiliary variable for each device to reduce the dimension of optimization variables. Let $\eta_i \in (0, 1)$ denote the time splitting factor for device i. Then, we have $T_i^{\text{cmp}} = \eta_i T^{\max}$ and $T_i^{\text{com}} = (1 - \eta_i)T^{\max}$. Thus, our goal can be transformed into a top-layer searching problem that optimizes the splitting factor $\{\eta_i\}_{\forall i}$ for each device as follows:

$$\text{(P5)} \quad \min E^{\text{round}}\left(\{\eta_i\}_{\forall i}\right) \tag{16}$$

$$\text{subject to:} \quad \eta_i^{\min} \le \eta_i \le \eta_i^{\max}, \forall i,$$

$$\text{variables:} \quad \{\eta_i\}_{\forall i}, \tag{16a}$$

where $\eta_i^{\min}$ and $\eta_i^{\max}$ are the lower and the upper bounds of the time splitting factor, respectively. Specifically, by combing (6), (7) and (8), the value of $\eta_i^{\min}$ can be obtained by

$$\eta_i^{\min} = \frac{\tau\omega D_i^{\text{loc}}}{T^{\max} f_i^{\max}}, \tag{17}$$

and $\eta_i^{\max}$ can be calculated by

$$\eta_i^{\max} = 1 - \frac{S}{T^{\max} B \log_2(1 + \frac{g_i P_i^{\max}}{N_0 B})}. \tag{18}$$

After decomposing the main problem into two sub-problems, we will proceed to present the solution for each problem in Section IV.

IV. Proposed Algorithm for FIMI

A. Solution for Problem (P3)

The procedure for solving Problem (P3) is outlined as follows. We first simplify the problem by variable substitutions. After that, we leverage the convex optimization technique to obtain the optimal solution.

Based on (1), D_i^{gen} can be expressed as a function of δ_i as follows:

$$D_i^{\text{gen}} = \left(\frac{\gamma + \delta_i}{\alpha}\right)^{-\frac{1}{\beta}} - D_i^{\text{loc}}. \tag{19}$$

Moreover, by combing Constraint (14a) and (19), f_i can be further represented by δ_i as

$$f_i = \frac{\tau\omega\left(\frac{\gamma+\delta_i}{\alpha}\right)^{-\frac{1}{\beta}}}{T_i^{\text{cmp}}}. \tag{20}$$

Next, by substituting (19) and (20) into Problem (P3), the problem is simplified as follows:

$$\text{(P6)} \quad \min_{\{\delta_i\}_{\forall i}} \sum_{i=1}^{I} \rho_i(\gamma + \delta_i)^{-\frac{3}{\beta}} \tag{21}$$

$$\text{subject to:} \quad \sum_{i=1}^{I} \delta_i = I + \frac{\zeta I}{N} \ln\left(\Delta^{\max}\right), \tag{21a}$$

$$\delta_i^{\min} \le \delta_i \le \delta_i^{\max}, \forall i, \tag{21b}$$

where ρ_i is an intermediate constant calculated by

$$\rho_i = \frac{\epsilon_i(\tau\omega)^3}{\left(T_i^{\text{cmp}}\right)^2 \alpha^{-\frac{3}{\beta}}} > 0. \tag{22}$$

Here, the lower bound of δ_i can be obtained by

$$\delta_i^{\min} = \alpha\left(\min\left\{\frac{f_i^{\max} T_i^{\text{cmp}}}{\tau\omega}, D_i^{\text{loc}} + D^{\max}\right\}\right)^{-\beta} - \gamma, \tag{23}$$

and the upper bound of δ_i is calculated as

$$\delta_i^{\max} = \alpha(D_i^{\text{loc}})^{-\beta} - \gamma. \tag{24}$$

Note that the solution to Problem (P6) relies on Constraint (21a). If $\sum_{i=1}^{I} \delta_i^{\min} > I + \frac{\zeta I}{N}\ln(\Delta^{\max})$ or $\sum_{i=1}^{I} \delta_i^{\max} < I + \frac{\zeta I}{N}\ln(\Delta^{\max})$, then Problem (P6) is infeasible. In the following, we focus on the practical case that $\sum_{i=1}^{I} \delta_i^{\min} \le I + \frac{\zeta I}{N}\ln(\Delta^{\max}) \le \sum_{i=1}^{I} \delta_i^{\max}$.

Theorem 1 (Solution for Problem (P6)): Suppose that a feasible scenario where inequality $\sum_{i=1}^{I} \delta_i^{\min} \le I + \frac{\zeta I}{N}\ln(\Delta^{\max}) \le \sum_{i=1}^{I} \delta_i^{\max}$ holds. The optimal solution to Problem (P6) can be expressed as[1]

$$\delta_i^* = \left[\left(\frac{3\rho_i}{\beta\nu^*}\right)^{\frac{\beta}{\beta+3}}\right]_{\delta_i^{\min}}^{\delta_i^{\max}}, \ \forall i, \tag{25}$$

where the variable ν^* is determined through a search process in order to find the optimal solution $\{\delta_i^*\}_{\forall i}$ that satisfies Constraint (21a).

Proof: It can be verified that Problem (P6) is a convex optimization problem with respect to δ_i. We next employ Karush-Kuhn-Tucker (KKT) the necessary condition for achieving the optimality. By introducing $\{\lambda_i\}_{\forall i}$ and $\{\psi_i\}_{\forall i}$ as the Lagrange multipliers for inequality constraints, and ν as the multiplier for the equality constraint, we have

$$\begin{cases} \lambda_i \ge 0, \ \lambda_i(\delta_i^{\min} - \delta_i) = 0, \forall i, & (26\text{a}) \\ \psi_i \ge 0, \ \psi_i(\delta_i - \delta_i^{\max}) = 0, \forall i, & (26\text{b}) \\ -\frac{3\rho_i}{\beta(\delta_i+\gamma)^{\frac{\beta+3}{\beta}}} - \lambda_i + \psi_i + \nu = 0, \forall i, & (26\text{c}) \\ \text{Constraints (21a) and (21b).} & (26\text{d}) \end{cases}$$

[1]The operation of $[x]_a^b = \min\{b, \max\{x, a\}\}$.

Algorithm 1: Bisection Search for Problem (P3).

Input: $\nu^{\min}$, $\nu^{\max}$, and a small positive number ε.
repeat
  Update $\nu = (\nu^{\max} + \nu^{\min})/2$;
  Compute $\{\delta_i\}_{\forall i}$ based on Eqn. (25);
  if $\sum_{i=1}^{I} \delta_i > I + \frac{\zeta I}{N} \ln\left(\Delta^{\max}\right)$ **then** update $\nu^{\min} = \nu$ **else** update $\nu^{\max} = \nu$;
until $|\nu^{\max} - \nu^{\min}| \leqslant \varepsilon$;
Set $\nu^* = \nu$, and acquire $\{\delta_i^*\}_{\forall i}$ based on Eqn. (25);
Compute $\{D_i^{\text{gen},*}\}_{\forall i}$ based on Eqn. (19), and obtain $\{f_i^*\}_{\forall i}$ based on Eqn. (20);
return $\{D_i^{\text{gen},*}, f_i^*\}_{\forall i}$

By putting (26c) into (26a) and (26c), we have

$$\begin{cases} \lambda_i \geq 0,\ \psi_i \geq 0, \forall i, & \text{(27b)} \\ \Bigg(\underbrace{-\frac{3\rho_i}{\beta(\delta_i+\gamma)^{\frac{\beta+3}{\beta}}} + \psi_i + \nu}_{A_1}\Bigg)(\delta_i^{\min} - \delta_i) = 0, \forall i, & \text{(27a)} \\ \Bigg(\underbrace{\frac{3\rho_i}{\beta(\delta_i+\gamma)^{\frac{\beta+3}{\beta}}} + \lambda_i - \nu}_{A_2}\Bigg)(\delta_i - \delta_i^{\max}) = 0, \forall i. & \text{(27c)} \end{cases}$$

Furthermore, the solution to Problem (P6) can be categorized into the following three cases, depending on the value of ν.

- When $\nu > \frac{3\rho_i}{\beta(\delta_i^{\min}+\gamma)^{\frac{\beta+3}{\beta}}}$, we have $A_1 > 0$. Based on (27a), we obtain $\delta_i = \delta_i^{\min}$.
- When $\nu < \frac{3\rho_i}{\beta(\delta_i^{\max}+\gamma)^{\frac{\beta+3}{\beta}}}$, we have $A_2 > 0$. Thus, we get $\delta_i = \delta_i^{\max}$.
- When $\frac{3\rho_i}{\beta(\delta_i^{\max}+\gamma)^{\frac{\beta+3}{\beta}}} \leq \nu \leq \frac{3\rho_i}{\beta(\delta_i^{\min}+\gamma)^{\frac{\beta+3}{\beta}}}$, we have $\lambda_i = 0$ and $\psi_i = 0$. Thus $\nu = \frac{3\rho_i}{\beta(\delta_i+\gamma)^{\frac{\beta+3}{\beta}}}$.

By summarizing the above three cases into one equation, we obtain the solution in (25). Thus, we complete the proof. □

We next aim to find the value of ν^* that satisfies Constraint (21a) according to Theorem 1. Based on (25), the optimal value ν^* can be expressed as a function of δ_i^* as follows:

$$\nu^* = \frac{3\rho_i}{\beta}(\delta_i^*)^{-\frac{\beta+3}{\beta}}. \tag{28}$$

Note that ν^* decreases as δ_i^* increases for $\delta_i^* > 0$. Thus, the search range $[\nu^{\min}, \nu^{\max}]$ for ν^* can be given by

$$\begin{cases} \nu^{\min} = \min\limits_i \left\{\frac{3\rho_i}{\beta}(\delta_i^{\max})^{-\frac{\beta+3}{\beta}}\right\}, & \text{(29a)} \\ \nu^{\max} = \max\limits_i \left\{\frac{3\rho_i}{\beta}\left(\delta_i^{\min}\right)^{-\frac{\beta+3}{\beta}}\right\}. & \text{(29b)} \end{cases}$$

According to (25), the value of $\sum_{i=1}^{I} \delta_i$ is non-increasing with respect to ν. Therefore, the value of ν^* can be efficiently obtained by the bi-section search. Furthermore, the algorithm for solving Problem (P3) is shown in Algorithm 1. Specifically, the computational complexity for Algorithm 1 can be approximated by $\mathcal{O}(\log_2(\nu^{\max} - \nu^{\min}))$.

B. Solution for Problem (P4)

To solve Problem (P4), we begin by transforming the problem into a simplified form that optimizes the bandwidth allocation.

Based on Constraint (15a), P_i can be represented by b_i as

$$P_i = \frac{N_0 b_i}{g_i}\left(2^{\frac{S}{b_i T_i^{\text{com}}}} - 1\right). \tag{30}$$

Note that P_i is decreasing with respect to b_i. Thus, by incorporating Constraint (12f), the lower bound of b_i can be expressed as

$$b_i \geq b_i^{\min} = -\frac{S \ln 2}{T_i^{\text{com}} W\left(-\frac{\kappa_i}{T_i^{\text{com}}}\exp\left(-\frac{\kappa_i}{T_i^{\text{com}}}\right)\right) + \kappa_i}, \tag{31}$$

where $\kappa_i = \frac{N_0 S \ln 2}{g_i P_i^{\max}}$ is an intermediate variable and $W(\cdot)$ is the Lambert function.

By putting (30) into (15) and replacing Constraint (12f) to (31), Problem (P4) can be simplified as follows:

$$\text{(P7)} \quad \min_{\{b_i\}_{\forall i}} \sum_{i=1}^{I} \frac{N_0 b_i T_i^{\text{com}}}{g_i}\left(2^{\frac{S}{b_i T_i^{\text{com}}}} - 1\right) \tag{32}$$

$$\text{subject to:} \quad \sum_{i=1}^{I} b_i = B, \tag{32a}$$

$$b_i \geq b_i^{\min}, \forall i, \quad . \tag{32b}$$

Theorem 2 (Solution for Problem (P7)): The optimal solution to Problem (P7) can be expressed as

$$b_i^* = \max\left\{b_i^{\min}, b_i(\varpi^*)\right\}, \ \forall i, \tag{33}$$

where the value of ϖ^* is determined through a search process, and the value of $b_i(\varpi^*)$ is obtained by solving the equation $Q(b_i) + \varpi^* = 0$. Here, the function $Q(b_i)$ is defined as

$$Q(b_i) = \frac{N_0 T_i^{\text{com}}\left(2^{\frac{S}{T_i^{\text{com}} b_i}} - 1\right)}{g_i} - \frac{\ln(2) N_0 S \cdot 2^{\frac{S}{T_i^{\text{com}} b_i}}}{g_i b_i}. \tag{34}$$

The optimal solution $\{b_i^*\}_{\forall i}$ under ϖ^* should satisfy Constraint (32a).

Proof: It can be verified that Problem (P7) is a convex optimization problem. We further utilize KKT conditions to obtain the necessary condition for achieving the optimality. By introducing $\{\varphi_i\}_{\forall i}$ as Lagrange multipliers for inequality constraints and ϖ as the multiplier for the equality constraint, we have

$$\begin{cases} \varphi_i \geq 0,\ \varphi_i(b_i^{\min} - b_i) = 0, \forall i, & \text{(35a)} \\ \frac{N_0 T_i^{\text{com}}\left(2^{\frac{S}{T_i^{\text{com}} b_i}} - 1\right)}{g_i} - \frac{\ln(2) N_0 S \cdot 2^{\frac{S}{T_i^{\text{com}} b_i}}}{g_i b_i} \\ -\varphi_i + \varpi = 0, \forall i, & \text{(35b)} \\ \text{Constraints (32a) and (32b).} & \text{(35c)} \end{cases}$$

By putting (35b) into (35a), we have

$$(Q(b_i) + \varpi) \cdot (b_i^{\min} - b_i) = 0, \tag{36}$$

Algorithm 2: Hierarchical Bisection Search.

Input: $\varpi^{\min}$, $\varpi^{\max}$, and error bound ε.
repeat
  Update $\varpi = (\varpi^{\max} + \varpi^{\min})/2$;
  Invoke `BandWidSearch`(ϖ) to obtain $\{b_i(\varpi)\}_{\forall i}$;
  if $\sum_{i=1}^{I} b_i(\varpi) > B$ **then** update $\varpi^{\min} = \varpi$ **else** update $\varpi^{\max} = \varpi$;
until $|\varpi^{\max} - \varpi^{\min}| \leqslant \varepsilon$;
Set $\varpi^* = \varpi$, and recall `BandWidSearch`(ϖ^*) to obtain $\{b_i^*\}_{\forall i}$;
Compute $\{P_i^*\}_{\forall i}$ based on Eqn. (30);
return $\{b_i^*, P_i^*\}_{\forall i}$
/* Function for searching $\{b_i(\varpi)\}_{\forall i}$. */
Function `BandWidSearch`(ϖ).
for each device $i = 1, 2, \ldots, I$ **in parallel do**
  Set $b_i^{\max} = B, \forall i$;
  repeat
    Update $b_i = (b_i^{\max} + b_i^{\min})/2$;
    Compute $Q(b_i)$ based on Eqn. (34);
    if $Q(b_i) + \varpi > 0$ **then** update $b_i^{\max} = b_i$ **else** update $b_i^{\min} = b_i$;
  until $|b_i^{\max} - b_i^{\min}| \leqslant \varepsilon$;
  Set $b_i(\varpi) = b_i$;
end
return $\{b_i(\varpi)\}_{\forall i}$

where $Q(b_i)$ is a function of b_i, and it is expressed as

$$Q(b_i) = \frac{N_0 T_i^{\text{com}}\left(2^{\frac{S}{T_i^{\text{com}} b_i}} - 1\right)}{g_i} - \frac{\ln(2) N_0 S \cdot 2^{\frac{S}{T_i^{\text{com}} b_i}}}{g_i b_i}. \tag{37}$$

Let $b_i(\varpi)$ denote the solution of $Q(b_i) + \varpi = 0$. Based on (36), the optimal solution to Problem (P6) is given by

$$b_i^* = \max\left\{b_i^{\min}, b_i(\varpi^*)\right\}, \ \forall i. \tag{38}$$

Thus, we complete the proof. □

We next investigate how the solution of $Q(b_i) + \varpi = 0$ is affected by ϖ. Specifically, the first-order derivative of $Q(b_i)$ with respect to b_i is expressed as

$$\frac{\partial H(b_i)}{\partial b_i} = \frac{\ln^2(2) S^2 N_0 \cdot 2^{\frac{S}{T_i^{\text{com}} b_i}}}{T_i^{\text{com}} g_i b_i^3} > 0. \tag{39}$$

Since $Q(b_i)$ is an increasing function with respect to b_i, the solution for $Q(b_i) + \varpi = 0$, denoted as $b_i(\varpi)$, will decrease with the increase of ϖ. Therefore, the value of $b_i(\varpi) \in [b_i^{\min}, B]$ can be obtained through the bisection search.

After that, our goal turns to search for ϖ^* that satisfies Constraint (32a). Since ϖ decreases with $b_i(\varpi)$, the search range $[\varpi^{\min}, \varpi^{\max}]$ for ϖ can be calculated by

$$\varpi^{\min} = Q(B), \text{and } \varpi^{\max} = \max_i \left\{Q(b_i^{\min})\right\}. \tag{40}$$

By combing (33) and the monotonicity of $b_i(\varpi)$, the value of $\sum_{i=1}^{I} b_i$ is non-increasing with respect to ϖ. Thus, the value of ϖ^* can be obtained by an outer bisection search.

Finally, by first solving Problem (P7) and then putting the optimal $\{b_i^*\}$ into (30), we acquire the optimal solution for Problem (P4). The workflow for solving Problem (P4) is presented in Algorithm 2, which involves a hierarchical bi-section search to obtain the optimal solution. For the inner search, we aim to solve $Q(b_i) + \varpi = 0, \forall i$ and acquire $\{b_i(\varpi)\}_{\forall i}$. For the outer search, we aim to search for ϖ^* that satisfies Constraint (32a). Regarding the computational complexity, the number of iteration steps required for inner and outer search are dominated by $\mathcal{O}(\log_2(\max\{B - b_i^{\min}, \forall i\}))$ and $\mathcal{O}(\log_2(\varpi^{\max} - \varpi^{\min}))$, respectively. Thus, the overall computational complexity of Algorithm 2 is measured as $\mathcal{O}(\log_2(\varpi^{\max} - \varpi^{\min}) \cdot \log_2(\max_i\{B - b_i^{\min}\}))$.

C. Cross-Entropy Searching for Problem (P5)

In this subsection, we aim at solving the top-layer problem that optimizes the time split factor (i.e., $\{\eta_i^*\}_{\forall i}$) for each device. However, directly obtaining the closed-form expression for function $E^{\text{round}}(\{\eta_i\}_{\forall i})$ is very difficult, hindering the subsequent theoretical analysis. Inspired by [42], [43], we employ the learning-based cross-entropy (CE) algorithm as a promising method to address this issue.

To leverage the CE method for solving Problem (P5), we incorporate the optimization variables into a vector denoted as $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_I)$. Additionally, we consider $\boldsymbol{\eta} \in \mathbb{R}^I$ as a random vector, which follows the normal distribution $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$. Instead of directly tuning $\boldsymbol{\eta}$, we focus on the joint optimization of the vector for mean value $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_I)$ and the vector for standard deviation $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_I)$.

Let j denote the iteration index and J denote the predetermined maximum iteration step for the CE algorithm. At the j-th iteration, the algorithm samples M feasible solutions $\{\boldsymbol{\eta}_{j,m}\}_{\forall m}$ from the distribution profile of $\{\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j\}$, where $\boldsymbol{\eta}_{j,m} \in \mathbb{R}^I$ represents the m-th generated solution at the j-th iteration. The fundamental concept behind the CE algorithm is to select the top K solutions with the best performance from $\{\boldsymbol{\eta}_{j,m}\}_{\forall m}$ for evolving the distribution profile. By iteratively converging towards the optimal distribution with the expectation value of $\boldsymbol{\mu}^*$, we obtain the numerical result for Problem (P5). The detailed procedure of the CE-based method is presented in Algorithm 3, which is explained as follows.

- (Line 1: Initialization): The algorithm initializes $\boldsymbol{\mu}_0 = (0.5, \ldots, 0.5)$ and $\boldsymbol{\sigma}_0 = (1, \ldots, 1)$.
- (Line 3: Sampling): During the j-th iteration, the algorithm generates M samples from distribution $\mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j^2)$ to form the solution set denoted as $\{\boldsymbol{\eta}_{j,m}\}_{\forall m}$.
- (Line 5: Selection): After invoking Algorithms 1 and 2, we obtain the corresponding objective values $\{E_{j,m}^{\text{round}}\}_{\forall m}$ for all the generated solutions $\{\boldsymbol{\eta}_{j,m}\}_{\forall m}$, where $E_{j,m}^{\text{round}} = E^{\text{round}}(\boldsymbol{\eta}_{j,m})$. By sorting $\{E_{j,m}^{\text{round}}\}_{\forall m}$ in an ascending order, we record the first K indices to form a set $\mathcal{K} \subset \{1, 2, \ldots, M\}$.
- (Line 6: Updating): The newly estimated distribution profile parameterized by $\tilde{\boldsymbol{\mu}}_{j+1}$ and $\tilde{\boldsymbol{\sigma}}_{j+1}$ are respectively updated by

$$\tilde{\boldsymbol{\mu}}_{j+1} = \sum_{m \in \mathcal{K}} \frac{\boldsymbol{\mu}_{j,m}}{K}, \text{and } \tilde{\boldsymbol{\sigma}}_{j+1} = \sum_{m \in \mathcal{K}} \frac{\boldsymbol{\sigma}_{j,m}}{K}. \tag{41}$$

Algorithm 3: Learning-Based Cross Entropy Searching.

1 **Initialization**. Mean value $\boldsymbol{\mu}_0 = (0.5, \ldots, 0.5)$, standard deviation $\boldsymbol{\sigma}_0 = (1, \ldots, 1)$, iteration counter $j = 0$, and a small positive number ε.
2 **while** $j < J$ and $\max\{\sigma | \sigma \in \boldsymbol{\sigma}_j\} > \varepsilon$ **do**
3 Generate M samples from $\mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j^2)$ to form the solutions $\{\boldsymbol{\eta}_{j,m}\}_{\forall m}$, and $m \in \{1, 2, \ldots, M\}$;
4 Invoke Algorithms 1 and 2 to compute $\{E_{j,m}^{\text{round}}\}_{\forall m}$, where $E_{j,m}^{\text{round}} = E^{\text{round}}(\boldsymbol{\eta}_{j,m})$;
5 Select the top K best-performing solutions from $\{\boldsymbol{\eta}_{j,m}\}_{\forall m}$, and record the corresponding indices to form $\mathcal{K} \subset (1, 2, \ldots, M)$;
6 Compute $\tilde{\boldsymbol{\mu}}_{j+1}$ and $\tilde{\boldsymbol{\sigma}}_{j+1}$ according to Eqn. (41);
7 Smooth the distribution profile to obtain $\boldsymbol{\mu}_{j+1}$ and $\boldsymbol{\sigma}_{j+1}$ based on Eqns. (42a) and (42b);
8 Update $j = j + 1$;
9 **end**
Output: The converged solution $\boldsymbol{\eta}^* = \boldsymbol{\mu}_j$.

- (Line 7: Smoothness): To reduce fluctuations during iteration, the two parameters of the distribution profile for the next iteration step are respectively smoothed as

$$\begin{cases} \boldsymbol{\mu}_{j+1} = \varrho \boldsymbol{\mu}_{j+1} + (1-\varrho)\tilde{\boldsymbol{\mu}}_{j+1}, & \text{(42a)} \\ \boldsymbol{\sigma}_{j+1} = \varrho \boldsymbol{\sigma}_{j+1} + (1-\varrho)\tilde{\boldsymbol{\sigma}}_{j+1}, & \text{(42b)} \end{cases}$$

where $\varrho \in (0, 1)$ is a pre-determined smoothing factor.

We now analyze the overall computational complexity for solving Problem (P1). Specifically, there are at most J iteration steps in Algorithm 3. At each iteration in Algorithm 3, it invokes M executions for both Algorithms 1 and 2. We use χ to denote the maximum search range among the three bi-section searches in Algorithms 1 and 2, where χ is computed by

$$\chi = \max\left\{\nu^{\max} - \nu^{\min}, \varpi^{\max} - \varpi^{\min}, \max_i\{B - b_i^{\min}\}\right\}. \tag{43}$$

The overall computational complexity for obtaining the solution to Problem (P1) can be expressed as

$$\mathcal{O}\left(JM\left(\underbrace{\log_2(\chi)}_{\text{Alg. 1}} + \underbrace{\log_2^2(\chi)}_{\text{Alg. 2}}\right)\right) = \mathcal{O}\left(JM\log_2^2(\chi)\right). \tag{44}$$

Note that the proposed search algorithm can be effectively executed by the server in practical scenarios. The empirical runtime can be disregarded when compared to the latency consumed for model training and parameter transmission.

D. Optimal Augmentation Policy

Upon obtaining the optimal quantity of synthetic data for each device $\{D_i^{\text{gen},*}\}_{\forall i}$, each device will determine the category-wise data amount locally. Let C denote the total number of label types (i.e., data categories) for the classification task. Each device aims at optimizing the distribution of synthetic data at a category-wise level. This involves determining $\{d_{i,c}^{\text{gen}}\}_{\forall c}, (c = 1, 2, \ldots, C)$ for each individual device.

According to [44], [45], we adopt the concept of *data entropy* to quantify the distribution of local data. For device i, the data entropy of the local mixed dataset $\mathcal{D}_i^{\text{loc}} \cup \mathcal{D}_i^{\text{gen}}$ is defined as

$$H_i = -\sum_{c=1}^{C} \frac{d_{i,c}^{\text{loc}} + d_{i,c}^{\text{gen}}}{D_i^{\text{loc}} + D_i^{\text{gen}}} \log_2\left(\frac{d_{i,c}^{\text{loc}} + d_{i,c}^{\text{gen}}}{D_i^{\text{loc}} + D_i^{\text{gen}}}\right). \tag{45}$$

where d_i^{loc} is the data amount of the c-th category before the data augmentation at device i, and $\sum_{c=1}^{C} d_i^{\text{loc}} = D_i^{\text{loc}}$. Intuitively, when the data distribution is uniform across all categories (i.e., $\forall c, d_{i,c}^{\text{loc}} + d_{i,c}^{\text{gen}} = \frac{D_i^{\text{loc}} + D_i^{\text{gen}}}{C}$), the data entropy attains its maximum value. Conversely, if data is skewed significantly toward a single category (i.e., $\exists c, d_{i,c}^{\text{loc}} + d_{i,c}^{\text{gen}} = D_i^{\text{loc}} + D_i^{\text{gen}}$), the data entropy reaches its minimum value, indicating a strong non-IID distribution.

The augmentation policy optimization at device i is to maximize its local data entropy by regulating the category-wise data amount $\{d_{i,c}^{\text{gen}}\}_{\forall c}$. Formally, the objective of device i is to solve the following optimization problem:

$$\text{(P8)} \quad \max_{\{d_{i,c}^{\text{gen}}\}_{\forall c}} H_i \tag{46}$$

$$\text{subject to:} \quad \sum_{c=1}^{C} d_{i,c}^{\text{gen}} = D_i^{\text{gen}}, \tag{46a}$$

$$0 \le d_{i,c}^{\text{gen}} \le D_i^{\text{gen}}, \forall c. \tag{46b}$$

Theorem 3 (Optimal local data augmentation): For a given device i, characterized by its category-wise local dataset distribution $\{d_{i,c}^{\text{loc}}\}_{\forall c}$ and a predetermined data amount of synthetic data D_i^{gen}, the optimal local data augmentation strategy to maximize the data entropy can be expressed as

$$d_{i,c}^{\text{gen}} = \left[\pi_i - d_{i,c}^{\text{loc}}\right]_0^{D_i^{\text{gen}}}. \tag{47}$$

Here, π_i represents the variable that needs to be sought in order to satisfy $\sum_{c=1}^{C} d_{i,c}^{\text{gen}} = D_i^{\text{gen}}$.

Proof: We first introduce a set of auxiliary variables $\{p_{i,c}\}_{\forall c}$ to denote the category-wise data proportion at device i, where $p_{i,c} = \frac{d_{i,c}^{\text{loc}} + d_{i,c}^{\text{gen}}}{D_i^{\text{loc}} + D_i^{\text{gen}}}$. We can further transform Problem (P8) into the following problem:

$$\text{(P9)} \quad \min_{\{p_{i,c}\}_{\forall c}} \sum_{c=1}^{C} p_{i,c} \log_2(p_{i,c}) \tag{48}$$

$$\text{subject to:} \quad \sum_{c=1}^{C} p_{i,c} = 1, \tag{48a}$$

$$p_{i,c}^{\min} \le p_{i,c} \le p_{i,c}^{\max}, \forall c, \tag{48b}$$

where $p_{i,c}^{\min} = \frac{d_{i,c}^{\text{loc}}}{D_i^{\text{loc}} + D_i^{\text{gen}}}$ and $p_{i,c}^{\max} = \frac{d_{i,c}^{\text{loc}} + D_i^{\text{gen}}}{D_i^{\text{loc}} + D_i^{\text{gen}}}$. It can be verified that Problem (P9) is a convex optimization problem. By introducing $\{\vartheta_{i,c}\}_{\forall c}$ and $\{\theta_{i,c}\}_{\forall c}$ as the Lagrange multipliers for inequality constraints, and ς_i as the multiplier for the equality constraint.

$$\begin{cases} \vartheta_{i,c} \geq 0,\ \vartheta_{i,c}(p_{i,c}^{\min} - p_{i,c}) = 0, \forall c, & (49a) \\ \theta_{i,c} \geq 0,\ \theta_{i,c}(p_{i,c} - p_{i,c}^{\max}) = 0, \forall c, & (49b) \\ G(p_{i,c}) - \vartheta_{i,c} + \theta_{i,c} + \varsigma_i = 0, \forall c, & (49c) \\ \text{Constraints (48a) and (48b)}, & (49d) \end{cases}$$

where $G(p_{i,c}) = \log_2(p_{i,c}) + \frac{1}{\ln(2)}$ is an intermediate function. Furthermore, we have

$$\begin{cases} \vartheta_{i,c} \geq 0,\ \theta_{i,c} \geq 0, \forall c, & (50a) \\ \left(\underbrace{G(p_{i,c}) + \theta_{i,c} + \varsigma_i}_{B_1}\right)(p_{i,c}^{\min} - p_{i,c}) = 0, \forall c, & (50b) \\ \left(\underbrace{-G(p_{i,c}) + \vartheta_{i,c} - \varsigma_i}_{B_2}\right)(p_{i,c} - p_{i,c}^{\max}) = 0, \forall c, & (50c) \end{cases}$$

Next, the solution to Problem (P9) can be divided into the following three cases according to the value of Lagrange multiplier ς_i.

- When $\varsigma_i > -G(p_{i,c}^{\min})$, we have $B_1 > 0$. Based on (50a), we obtain $p_{i,c} = p_{i,c}^{\min}$.
- When $\varsigma_i < -G(p_{i,c}^{\max})$, we have $B_2 > 0$. Based on (50b), we obtain $p_{i,c} = p_{i,c}^{\max}$.
- When $-G(p_{i,c}^{\max}) \leq \varsigma_i \leq -G(p_{i,c}^{\min})$, we have $\vartheta_{i,c} = 0$ and $\theta_{i,c} = 0$. Thus $\varsigma_i = -\log_2(p_{i,c}) - \frac{1}{\ln(2)}$.

As a result, the optimal solution to the Problem (P9) is summarized as

$$p_{i,c}^* = \left[2^{-(\frac{1}{\ln(2)} + \varsigma_i^*)}\right]_{p_{i,c}^{\min}}^{p_{i,c}^{\max}}, \ \forall c. \tag{51}$$

By putting $p_{i,c}$ into $d_{i,c}^{\text{gen}} = p_{i,c}(D_i^{\text{loc}} + D_i^{\text{gen}}) - d_{i,c}^{\text{loc}}$, the value of $d_{i,c}^{\text{gen}}$ can be expressed as

- When $p_{i,c}^* = p_{i,c}^{\min}$, we have $d_{i,c}^{\text{gen},*} = 0$.
- When $p_{i,c}^* = p_{i,c}^{\max}$, we have $d_{i,c}^{\text{gen},*} = D_i^{\text{gen}}$.
- When $p_{i,c}^* = 2^{-(\frac{1}{\ln(2)} + \varsigma_i^*)}$, we have $d_{i,c}^{\text{gen},*} = -d_{i,c}^{\text{loc}} + 2^{-(\frac{1}{\ln(2)} + \varsigma_i^*)} \cdot (D_i^{\text{loc}} + D_i^{\text{gen}})$.

It can be verified that the above solution is equivalent to (47). Thus, we complete the proof. □

Note that the numerical value of π_i^* can be efficiently obtained through linear search or bi-section search. Thus, we complete the device-side augmentation strategy optimization.

V. Experiment Evaluations

A. Experiment Settings

1) Settings for Edge Devices: We consider $I = 20$ edge devices for FL training within a 400-meter radius cell, with the base station positioned at the center of the cell. We utilize the path loss model from [31] to model the communication link, which is represented as $128.1 + 37.6\log_{10}(R_i)$ with R_i denoting the distance between the server and device i in kilometers. The power spectral density of additive Gaussian noise is $N_0 = -174$ dBm/MHz and the total bandwidth is $B = 20$ MHz. The maximum transmitting power for each device follows the uniform distribution $P_i^{\max} \sim U(20, 23)$ dBm. Regarding the model training, the training workload for a single sample is measured as 5×10^6 cycles. The maximum computing frequency and the hardware energy coefficient for each device follow the uniform distributions of $f_i^{\max} \sim U(1, 2)$ GHz, and $\epsilon_i \sim U(4 \times 10^{-27}, 6 \times 10^{-27})$, respectively.

2) Settings for FL Training: Our study is targeted for the implementation of image classification using the CIFAR10 and GTSRB datasets [11], [46]. We employ the VGG-9 model with the parameter size of 111.7Mb during uplink transmission [10]. In terms of data distribution, we evaluate the proposed framework using two Dirichlet distribution levels: Dir(0.4) for skewed non-IID and Dir(0.6) for slight non-IID scenarios [12]. For the local dataset, each device is assigned 1250 training samples, i.e., $d_i^{\text{loc}} = 1250, \forall i$. For the other hyper-parameters, we set the learning rate as 0.02, the batch size as $B = 64$, the local epoch as $\tau = 1$, the maximum allowable global error as $\Delta^{\max} = 0.2$, the maximum latency per round $T^{\max} = 60$ seconds, and the number of the global rounds as $N = 200$.

3) Settings for Generative Model: The pre-trained generative model is trained on the public dataset (e.g., CINIC10) with the classifier-free guidance technique [47], [48]. The number of denoising steps is set as 300 and the output image resolution is 32×32. Specifically, the total amount of synthetic data is 6659 under the default setting, and it takes about 7.17 minutes for a workstation equipped with eight RTX A5000 GPUs to generate the synthetic data. During the synthetic data distribution, each device receives about 333 samples on average. This yields about 8.1 Mb of downlink traffic per device on average, which is much smaller than 111.7 Mb incurred by the one-time model upload. Therefore, the latency associated with data synthesis and distribution is negligible compared to the numerous rounds of FL training.

B. Performance Comparisons With Existing Methods

We conduct performance comparisons between our proposed FIMI approach and several existing methods, including traditional approaches and data augmentation-based FL methods. Their details are as follows.

- *Traditional FL (TFL):* TFL adheres to the conventional FL paradigm, wherein each device solely uses its local dataset for model training [5].
- *Semi-supervised FL (SEMI):* SEMI incorporates unlabelled samples to enhance model updates [21].
- *Heuristic data compensation (HDC):* Each device adopts a heuristic scheme, namely, generating synthetic data solely for the category with the least available local data.
- *Server-side training (SST):* SST preserves the synthetic data at the server, and conducts model training to generate a complementary update every round. This complementary

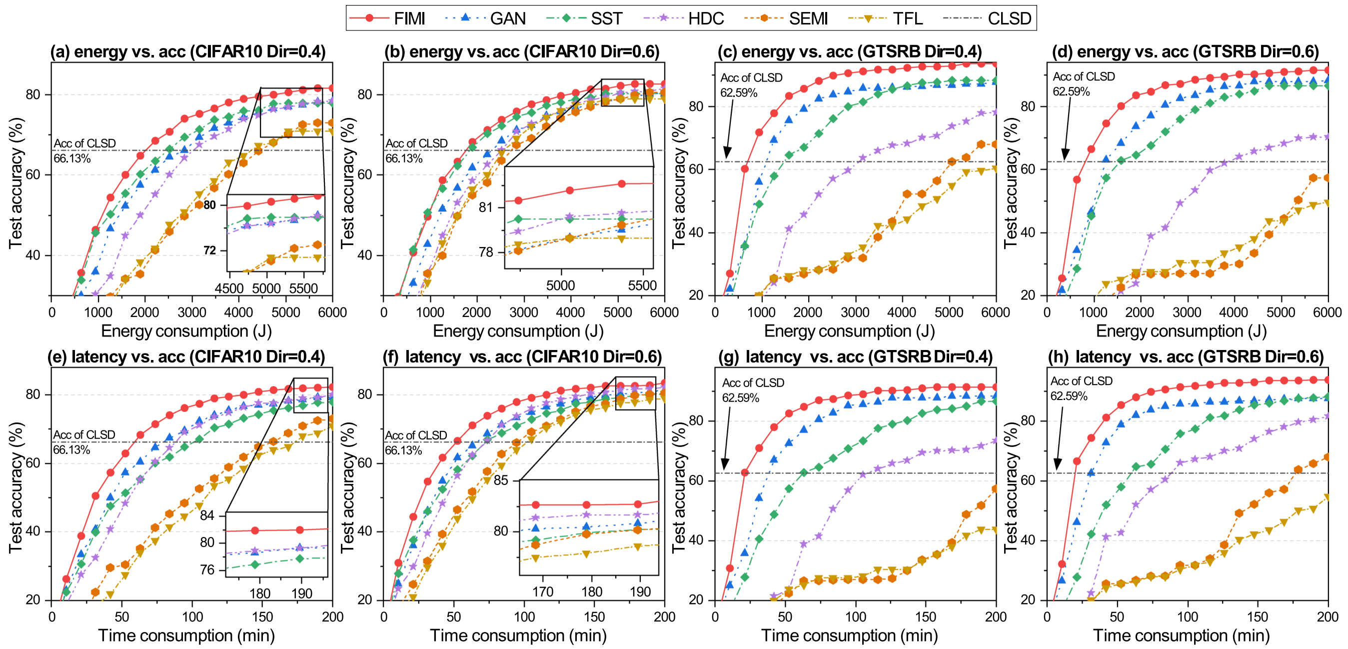


Fig. 4. Performance comparison of FIMI and baseline methods on CIFAR10 and GTSRB datasets with different levels of non-IID distribution. (The upper row: energy consumption versus test accuracy; the lower row: time consumption versus test accuracy.).

TABLE I
PERFORMANCE COMPARISON BETWEEN FIMI AND OTHER BASELINE METHODS

		Dir=0.4				Dir=0.6			
Dataset	Method	Energy@70 (J)*	Latency@70 (h)*	Uplink@70 (GB)*	Best Acc.(%)	Energy@78 (J)*	Latency@78 (h)*	Uplink@78 (GB)*	Best Acc.(%)
CIFAR10	TFL	4858.64 (1.0×)	3.22 (1.0×)	56.45 (1.0×)	70.86	4506.20 (1.0×)	2.98 (1.0×)	52.36 (1.0×)	78.95
	SEMI	4913.18 (1.0×)	2.85 (0.9×)	50.02 (0.9×)	73.32	4625.86 (1.0×)	2.68 (0.9×)	47.09 (0.9×)	80.54
	HDC	3507.94 (0.7×)	1.68 (0.5×)	29.54 (0.5×)	79.92	4098.38 (0.9×)	1.97 (0.7×)	34.52 (0.7×)	82.05
	SST	2844.70 (0.6×)	1.88 (0.6×)	33.05 (0.6×)	77.92	3725.80 (0.8×)	2.47 (0.8×)	43.29 (0.8×)	80.23
	GAN	3403.74 (0.7×)	1.63 (0.5×)	28.66 (0.5×)	79.43	4723.56 (1.0×)	2.27 (0.8×)	39.78 (0.8×)	81.34
	FIMI	**2361.78** (0.5×)	**1.13** (0.4×)	**19.89** (0.4×)	**82.21**	**3264.81** (0.7×)	**1.57** (0.5×)	**27.50** (0.5×)	**83.41**
Dataset	Method	Energy@72 (J)*	Latency@72 (h)*	Uplink@72 (GB)*	Best Acc.(%)	Energy@80 (J)*	Latency@80 (h)*	Uplink(GB)*	Best Acc.(%)
GTSRB	TFL	N/A†	N/A†	N/A†	43.81	N/A†	N/A†	N/A†	54.84
	SEMI	N/A†	N/A†	N/A†	57.33	N/A†	N/A†	N/A†	67.93
	HDC	6807.48 (1.0×)	3.27 (1.0×)	57.33 (1.0×)	73.40	6494.89 (1.0×)	3.12 (1.0×)	54.70 (1.0×)	81.59
	SST	2467.08 (0.4×)	1.63 (0.5×)	28.66 (0.5×)	86.51	2819.52 (0.4×)	1.87 (0.6×)	32.76 (0.6×)	89.04
	GAN	1806.07 (0.3×)	0.87 (0.3×)	15.21 (0.3×)	88.46	2049.19 (0.3×)	0.98 (0.3×)	17.26 (0.3×)	87.69
	FIMI	**1146.16** (0.2×)	**0.55** (0.2×)	**9.65** (0.2×)	**91.48**	**1319.82** (0.2×)	**0.63** (0.2×)	**11.12** (0.2×)	**93.80**

*"Energy@x", "Latency@x" and "Uplink@x" denote the required energy, latency and uplink traffic size to reach x% accuracy, respectively.
†"N/A" means that the corresponding method cannot achieve the target global accuracy within the predefined maximum global iteration.

update is aggregated with the local model updates to renew the global model.

- *GAN-based data synthesis (GAN):* The server uses a pre-trained conditional GAN model to generate additional training samples [23].
- *Centralized learning with synthetic data (CLSD):* This method only utilizes the generated synthetic data provided by the server to perform centralized training.

To provide a reasonable comparison, we adopt the identical optimization algorithm as FIMI for SEMI, HDC, and GAN. In the cases of TFL and SST, the synthetic data of each device is set to zero, and the resource utilization policy is optimized.

Fig. 4 shows the performance comparison between FIMI and baseline methods under different levels of non-IID data distribution. As shown in Fig. 4(a)–(d), FIMI outperforms the baseline methods in terms of the test accuracy to reduce energy consumption. Furthermore, Fig. 4(e)–(h) show that the proposed FIMI can achieve a faster convergence rate to reduce the training latency. To evaluate the learning performance, we measure the system cost in terms of energy consumption, training latency, and uplink transmission traffic required to achieve the desired test accuracy for each method. The results are provided in Table I. The advantages of our proposed FIMI can be summarized into the following three key aspects, i.e., reducing the energy

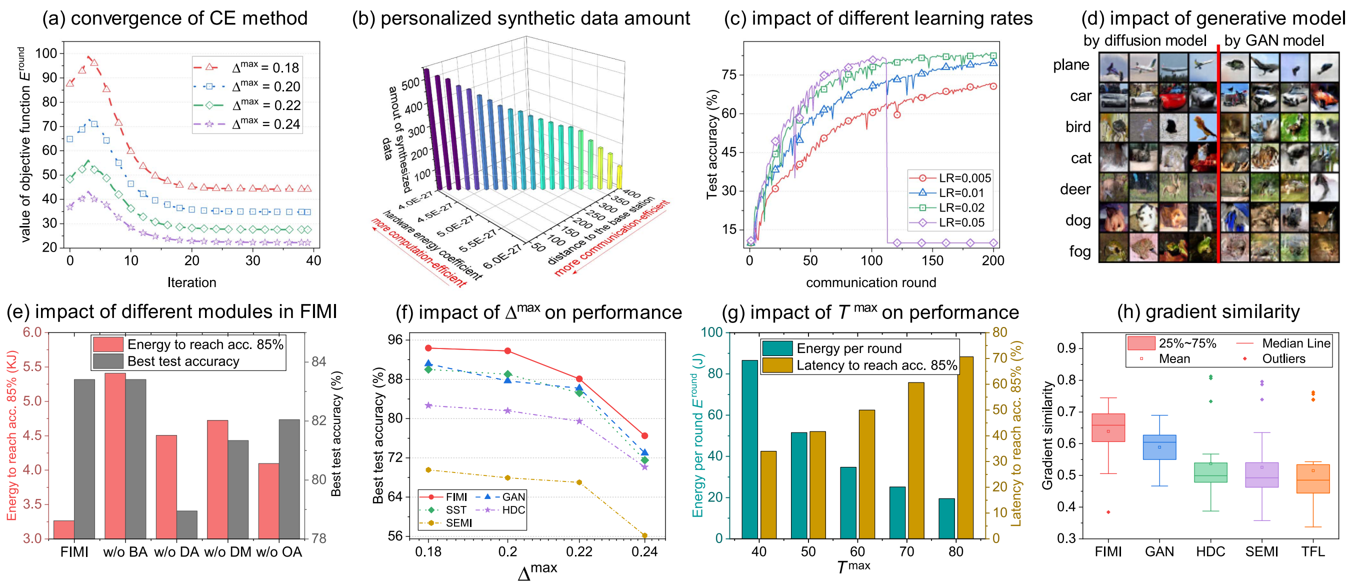


Fig. 5. Impact of key mechanisms and hyper-parameters on the generative AI-based FL training system.

consumption, reducing the time consumption, and improving the converged accuracy.

1) Reducing Energy Consumption: As shown in Table I, our FIMI consumes the energy consumption smaller than the other four methods. Specifically, given the test accuracy of 70% on the CIFAR10 dataset under Dir(0.4), our FIMI achieves 50% reduction in the energy consumption compared to TFL. As shown in the results, although TFL requires the lowest energy consumption per global iteration, it consumes more iterations than FIMI to achieve the desired accuracy, which consequently degrades the performance of TFL in energy consumption. In contrast, FIMI can efficiently achieve the target accuracy with less iterations than TFL. Meanwhile, FIMI reduces the required data size of uplink traffic by approximately 60% and 50% compared to TFL under the data distributions of Dir(0.4) and Dir(0.6), respectively. Therefore, our FIMI can effectively reduce the energy consumption than TFL.

2) Reducing Time Consumption: As shown in Table I, our FIMI can effectively reduce the latency for achieving the desired accuracy compared to the other baseline methods. Specifically, when evaluated on the GTSRB dataset with a target accuracy of 80% under Dir(0.6), FIMI can reduce the latency by 80% and 70% compared to HDC and GAN, respectively. It is noticed that the advantage of reduced latency is mainly attributed to the faster convergence rate of FIMI. This is shown in Fig. 4(e)–(h). Specifically, the convergence rates of the methods ranked according to the descending order are as follows: FIMI, GAN, SST, HDC, SEMI, and TFL.

3) Improving Converged Accuracy: As shown in Table I, the proposed FIMI can effectively improve the accuracy of the converged global model, outperforming TFL by 11.35% and 4.46% on the CIFAR10 dataset under the Dirichlet data distributions of 0.4 and 0.6, respectively. Meanwhile, we also investigate the performance of model training by solely using AI-synthetic data (i.e., the CLSD method) and show the corresponding results in Fig. 4. The results show that relying solely on the synthetic data for model training will result in unsatisfactory convergence accuracy, since the AI-generated dataset still shows a different distribution from the local dataset. This result highlights the importance of using the generative AI model as a data augmentation method instead of solely using synthetic data for model training.

C. Detailed Performances of Different Modules in FIMI

1) Performance of CE Method: Fig. 5(a) shows the convergence of the proposed CE-based searching algorithm under different settings of maximum allowable error $\Delta^{\max}$. The first increasing and then decreasing pattern of the convergence curves can be attributed to the exploration-exploitation trade-off inherent in the CE method. The CE algorithm explores the solution space widely in the initial stages, and then refines the search to converge toward the promising areas in the later stages. Moreover, Fig. 5(b) shows the relationship between the amount of synthetic data $D_i^{\text{gen},*}$ with respect to the computation and communication conditions. We construct two arithmetic sequences, one for hardware energy coefficients $\{\epsilon_i\}_{\forall i}$ and the other for the distances $\{R_i\}_{\forall i}$ between the edge devices and the parameter server, and then optimize the data augmentation policy. The result shows that the devices with smaller ϵ_i and R_i will generate more synthetic data for local training. This result is consistent with the intuition, namely, it will be more beneficial for the devices with favorable channel conditions and energy features to leverage a larger amount of synthetic data for local training.

2) Impact of Generative Models and Hyper-Parameters: Fig. 5(c) shows the convergence curves of FIMI under different choices of learning rate. Specifically, moderate learning rates ranging from 0.01 to 0.03 are recommended for achieving fast convergence while maintaining stability during the training process. Fig. 5(d) shows the synthetic data samples using both the diffusion model and the GAN-based model. The results show that the diffusion-based model can outperform the GAN-based model by generating more realistic images with detailed features. For example, the synthetic images of the fog produced by the diffusion model exhibit a richer texture than the GAN-based model. The images generated by the GAN-based model appear somewhat blurry. Next, we gradually eliminate the essential components of FIMI, including the bandwidth allocation (w/o BA), data augmentation (w/o DA), the diffusion model (w/o DM), and optimal augmentation (w/o OA). Fig. 5(e) shows the ablation studies for our proposed FIMI. We observe that the enhanced resource utilization and improved converged model accuracy primarily stem from the proposed resource allocation and data augmentation, respectively. Fig. 5(f)–(g) show the influence of $\Delta^{\max}$ and $T^{\max}$ on the data augmentation-based FL training. We utilize the GTSRB dataset under Dir(0.6) for the experiments. Specifically, a smaller value for $\Delta^{\max}$ yields an improved test accuracy for the converged global model. Our empirical findings suggest that setting the maximum allowable error $\Delta^{\max}$ within the range of $[0.15, 0.25]$ for practical implementations should be appropriate. Furthermore, the single-round maximum latency $T^{\max}$ regulates the training duration for the FL system. Using a more stringent $T^{\max}$ (i.e., a smaller value) can reduce the training time, while sacrificing an increased energy consumption.

3) Explanation of FIMI: We illustrate the key factors that enable the FIMI to achieve superior performance over existing methods. To shed light on this, we simulate a virtual device holding a locally uniform dataset, resembling IID distribution across all data categories. Let $\boldsymbol{g}_0$ denote the local gradient of the virtual device, and let $\boldsymbol{g}_i$ denote the local gradients of device i resulting from the mixed dataset. Our goal is to quantify the gradient similarity between the edge device and the virtual device. Based on [49], we define the gradient similarity between $\boldsymbol{g}_0$ and $\boldsymbol{g}_i$ for an L-layer shared model as

$$\mathtt{Sim}(\boldsymbol{g}_0, \boldsymbol{g}_i) = \frac{1}{2L}\sum_{l=1}^{L}\left(\frac{< \boldsymbol{g}_0^{[l]}, \boldsymbol{g}_i^{[l]} >}{\|\boldsymbol{g}_0^{[l]}\|\|\boldsymbol{g}_i^{[l]}\|} + 1\right). \tag{52}$$

Here, $\boldsymbol{g}_i^{[l]}$ denotes the gradient vector of $\boldsymbol{g}_i$ in the l-th layer, $< \cdot, \cdot >$ is the dot product operation, and $\|\cdot\|$ calculates the euclidean norm for a given vector. As presented in the box plots of Fig. 5(h), the gradient of our FIMI shows the highest level of similarity to that of the IID local data. This result means that our FIMI can effectively mitigate the issue of non-IID data distribution across different edge devices, which thus accelerates the convergence rate of FL training and improves the accuracy of the converged global model.

VI. Conclusion and Future Work

In this paper, we have proposed a resource-aware data augmentation framework called FIMI to leverage pre-trained generative AI models to synthesize customized local data for FL training. FIMI can improve the convergence accuracy of FL and save the device-side energy consumption by mitigating data and device resource heterogeneity. Our research results have demonstrated the potential of generative AI in FL and provided the insights for developing efficient and effective FL systems. For our future work, we will explore multi-server empowered AI content generation services in mobile edge computing to facilitate the data synthesis procedure. We will also investigate the incentive mechanisms for the collaborative synergy between generative AI and FL.

References

[1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10684–10695.

[2] C. Saharia et al., "Photorealistic text-to-image diffusion models with deep language understanding," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 36479–36494.

[3] M. Xu et al., "Unleashing the power of edge-cloud generative AI in mobile networks: A survey of AIGC services," *IEEE Commun. Surv. Tutor.*, early access, Jan. 12, 2024, doi: 10.1109/COMST.2024.3353265.

[4] S. Azizi, S. Kornblith, C. Saharia, M. Norouzi, and D. J. Fleet, "Synthetic data from diffusion models improves imagenet classification," 2023, *arXiv:2304.08466*.

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

[6] W. Wu et al., "AI-native network slicing for 6G networks," *IEEE Wirel. Commun.*, vol. 29, no. 1, pp. 96–103, Feb. 2022.

[7] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Wireless communications for collaborative federated learning," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 48–54, Dec. 2020.

[8] R. Yu and P. Li, "Toward resource-efficient federated learning in mobile edge computing," *IEEE Netw.*, vol. 35, no. 1, pp. 148–155, Jan./Feb. 2021.

[9] Y. Guo, X. Tang, and T. Lin, "FedBR: Improving federated learning on heterogeneous data via local learning bias reduction," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2023, pp. 12034–12054.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–14.

[11] A. Krizhevsky, "Learning multiple layers of features from tiny images," Master's thesis, Univ. Toronto, Toronto, ON, Canada, 2009.

[12] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–16.

[13] Z. Zhang, Z. Gao, Y. Guo, and Y. Gong, "Scalable and low-latency federated learning with cooperative mobile edge networking," *IEEE Trans. Mob. Comput.*, vol. 23, no. 1, pp. 812–822, Jan. 2024.

[14] C. Zheng, G. Wu, and C. Li, "Toward understanding generative data augmentation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2024, pp. 1–15.

[15] H. Du et al., "Exploring collaborative distributed diffusion-based AI-generated content (AIGC) in wireless networks," *IEEE Netw.*, early access, Jul. 03, 2023, doi: 10.1109/MNET.006.2300223.

[16] M. Xu et al., "Joint foundation model caching and inference of generative AI services for edge intelligence," 2023, *arXiv:2305.12130*.

[17] J. Wang et al., "Generative AI for integrated sensing and communication: Insights from the physical layer perspective," 2023, *arXiv:2310.01036*.

[18] X. Huang et al., "Federated learning-empowered AI-generated content in wireless networks," *IEEE Netw.*, early access, Jan. 12, 2024, doi: 10.1109/MNET.2024.3353377.

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

[20] S. Itahara, T. Nishio, Y. Koda, M. Morikura, and K. Yamamoto, "Distillation-based semi-supervised federated learning for communication-efficient collaborative training with Non-IID private data," *IEEE Trans. Mob. Comput.*, vol. 22, no. 1, pp. 191–205, Jan. 2023.

[21] S. Wang, Y. Xu, Y. Yuan, and T. Q. S. Quek, "Toward fast personalized semi-supervised federated learning in edge networks: Algorithm design and theoretical guarantee," *IEEE Trans. Wirel. Commun.*, vol. 23, no. 2, pp. 1170–1183, Feb. 2024.

[22] R. Zhao, Y. Wang, Z. Xue, T. Ohtsuki, B. Adebisi, and G. Gui, "Semisupervised federated-learning-based intrusion detection method for Internet of Things," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8645–8657, May 2023.

[23] B. Xin, W. Yang, Y. Geng, S. Chen, S. Wang, and L. Huang, "Private FL-GAN: Differential privacy synthetic data generation based on federated learning," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 2927–2931.

[24] E. Jeong, S. Oh, J. Park, H. Kim, M. Bennis, and S. Kim, "Multi-hop federated private data augmentation with sample compression," in *Proc. Int. Joint Conf. Artif. Intell. Workshop*, 2019, pp. 1–8.

[25] P. Li et al., "Snowball: Energy efficient and accurate federated learning with coarse-to-fine compression over heterogeneous wireless edge devices," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 10, pp. 6778–6792, Oct. 2023.

[26] Y. Xu, Y. Liao, H. Xu, Z. Ma, L. Wang, and J. Liu, "Adaptive control of local updating and model compression for efficient federated learning," *IEEE Trans. Mob. Comput.*, vol. 22, no. 10, pp. 5675–5689, Oct. 2023.

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

[28] Y. Deng et al., "SHARE: Shaping data distribution at edge for communication-efficient hierarchical federated learning," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst.*, 2021, pp. 24–34.

[29] S. Hosseinalipour et al., "Multi-stage hybrid federated learning over large-scale D2D-enabled fog networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 4, pp. 1569–1584, Aug. 2022.

[30] Y. Wu, Y. Song, T. Wang, L. Qian, and T. Q. 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.

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

[32] W. Wu et al., "Split learning over wireless networks: Parallel design and resource management," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1051–1066, Apr. 2023.

[33] H. Ko, J. Lee, S. Seo, S. Pack, and V. C. Leung, "Joint client selection and bandwidth allocation algorithm for federated learning," *IEEE Trans. Mob. Comput.*, vol. 22, no. 6, pp. 3380–3390, Jun. 2023.

[34] I. Chen, F. D. Johansson, and D. Sontag, "Why is my classifier discriminatory?," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 3543–3554.

[35] S. Wang, Y.-C. Wu, M. Xia, R. Wang, and H. V. Poor, "Machine intelligence at the edge with learning centric power allocation," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 11, pp. 7293–7308, Nov. 2020.

[36] Q. Hu, S. Wang, Z. Xiong, and X. Cheng, "Nothing wasted: Full contribution enforcement in federated edge learning," *IEEE Trans. Mob. Comput.*, vol. 22, no. 5, pp. 2850–2861, May 2023.

[37] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey, "CINIC-10 is not ImageNet or CIFAR-10," 2018, *arXiv: 1810.03505*.

[38] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv: 1708.07747*.

[39] C. Ma et al., "Distributed optimization with arbitrary local solvers," *Optim. Methods Softw.*, vol. 32, no. 4, pp. 813–848, 2017.

[40] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1387–1395.

[41] P. Li et al., "AnycostFL: Efficient on-demand federated learning over heterogeneous edge devices," in *Proc. IEEE Conf. Comput. Commun.*, 2023, pp. 1–10.

[42] 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 Commun. Lett.*, vol. 9, no. 3, pp. 402–405, Mar. 2020.

[43] Z. Sun, Y. Xiao, L. You, L. Yin, P. Yang, and S. Li, "Cross-entropy-based antenna selection for spatial modulation," *IEEE Commun. Lett.*, vol. 20, no. 3, pp. 622–625, Mar. 2016.

[44] G. Xu, Z. Zhou, J. Dong, L. Zhang, and X. Song, "A blockchain-based federated learning scheme for data sharing in Industrial Internet of Things," *IEEE Internet Things J.*, vol. 10, no. 24, pp. 21467–21478, Dec. 2023.

[45] W. Jiang, M. Chen, and J. Tao, "Federated learning with blockchain for privacy-preserving data sharing in Internet of Vehicles," *China Commun.*, vol. 20, no. 3, pp. 69–85, 2023.

[46] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The German traffic sign detection benchmark," in *Proc. Int. Joint Conf. Neural Netw.*, 2013, pp. 1–8.

[47] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 6840–6851.

[48] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *Proc. Int. Conf. Neural Inf. Process. Syst. Workshop*, 2021, pp. 1–8.

[49] B. Zhao, K. R. Mopuri, and H. Bilen, "Dataset condensation with gradient matching," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–12.

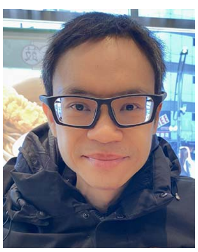
Peichun Li received the MSc degree from the Guangdong University of Technology, China, in 2021. He is currently a research assistant with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Macao, China. His research interests include edge computing and deep learning, particularly in efficient algorithms for artificial intelligence (AI) applications.

Hanwen Zhang received the BSc degree from the University of Electronic Science and Technology of China, China, in 2023. He is currently working toward the PhD degree with the Department of Computer and Information Science, University of Macau, Macao, China. His research interests mainly focus on edge computing and deep learning.

Yuan Wu (Senior Member, IEEE) received the PhD degree in electronic and computer engineering from the Hong Kong University of Science and Technology in 2010. He is currently an associate professor with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Macao, China, and also with the Department of Computer and Information Science, University of Macau. His research interests include resource management for wireless networks, green communications and computing, edge computing and edge intelligence, and energy informatics. He received the Best Paper Award from the IEEE ICC'2016, WCSP'2016, IEEE TCGCC'2017, IWCMC'2021, and WCNC'2023. 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*.

Liping Qian (Senior Member, IEEE) received the PhD degree in information engineering from the Chinese University of Hong Hong in 2010. She worked as a postdoctoral research associate with the Chinese University of Hong Kong, China, during 2010-2011. Since 2011, she has been with College of Information Engineering, Zhejiang University of Technology, Hangzhou, China, where she is currently a full professor. From 2016 to 2017, she was a visiting scholar with the Broadband Communications Research Group, ECE Department, University of Waterloo. Her research interests include wireless communication and networking, resource management in wireless networks, massive IoTs, mobile edge computing, emerging multiple access techniques, and machine learning oriented towards wireless communications. She was a co-recipient of the IEEE Marconi Prize Paper Award in Wireless Communications in 2011, the Best Paper Award from IEEE ICC 2016, the Best Paper Award from IEEE Communication Society GCCTC 2017, the Best Paper Award from the Digital Communications and Networking in 2021, and the Best Paper Award from IEEE WCNC 2023. She was an associate editor of the IET Communications from 2016 to 2022. She is currently on the Editorial Board of *IEEE Transactions on Cognitive Communications and Networking*.

Rong Yu (Member, IEEE) received the BS degree in communication engineering from the Beijing University of Posts and Telecommunications, China, in 2002, and the PhD degree in electronic engineering from Tsinghua University, China, in 2007. After that, he was with the School of Electronic and Information Engineering, South China University of Technology. In 2010, he joined the School of Automation, Guangdong University of Technology, where he is currently a professor. He was a member of the Home Networking Standard Committee, China, where he led the standardization work of three standards. He received the Best Paper Awards from International Conferences including IEEE Blockchain 2022 and IEEE ICCC 2016. His research interests mainly focus on wireless networking and mobile computing, such as edge computing, federated learning, blockchain, digital twin, connected vehicles, and smart grid.



Dusit Niyato (Fellow, IEEE) received the BEng from the King Mongkuts Institute of Technology Ladkrabang (KMITL), Thailand in 1999 and the PhD degree in electrical and computer engineering from the University of Manitoba, Canada in 2008. He is a professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests are in the areas of sustainability, edge intelligence, decentralized machine learning, and incentive mechanism design.

Xuemin (Sherman) Shen (Fellow, IEEE) received the PhD 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, 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. He received "West Lake Friendship Award" from Zhejiang Province in 2023, President's Excellence in Research from the University of Waterloo in 2022, 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 has also received 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 serves/served as the General Chair for the 6G Global Conference'23, and ACM Mobihoc'15, Technical Program Committee Chair/Co-Chair for IEEE Globecom'24, 16 and 07, IEEE Infocom'14, IEEE VTC'10 Fall, and the Chair for the IEEE ComSoc Technical Committee on Wireless Communications. Dr. Shen is the President of the IEEE ComSoc. He was the vice president for Technical & 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 served as the editor-in-chief of the *IEEE IoT JOURNAL*, *IEEE Network*, and *Peer-to-Peer Networking and Applications*.