Accelerating Federated Edge Learning via Wireless and Heterogeneity Aware Subnetwork Scheduling

Liang Li, *Member, IEEE,* Jiaxiang Geng, *Student Member, IEEE,* Huai-an Su, *Student Member, IEEE,* Xiaoqi Qin, *Senior Member, IEEE,* Yanzhao Hou, *Member, IEEE,* Hao Wang, *Member, IEEE,* Xin Fu, *Senior Member, IEEE,* and Miao Pan, *Senior Member, IEEE.*

***Abstract*—As a popular distributed learning paradigm, federated learning (FL) over mobile devices fosters numerous applications, while their practical deployment is hindered by participating devices' computing and communication heterogeneity. Some pioneering research efforts proposed to extract subnetworks from the global model, and assign as large a subnetwork as possible to the device for local training based on its full computing and communications capacity. Although such fixed size subnetwork assignment enables FL training over heterogeneous mobile devices, it is unaware of (i) the dynamic changes of devices' communication and computing conditions and (ii) FL training progress and its dynamic requirements of local training contributions, both of which may cause very long FL training delay. Motivated by those dynamics, in this paper, we develop a wireless and heterogeneity aware latency efficient FL (WHALE-FL) approach to accelerate FL training through adaptive subnetwork scheduling. Instead of sticking to the fixed size subnetwork, WHALE-FL introduces a novel subnetwork selection utility function to capture device and FL training dynamics, and guides the mobile device to adaptively select the subnetwork size for local training based on (a) its computing and communication capacity, (b) its dynamic computing and/or communication conditions, and (c) FL training status and its corresponding requirements for local training contributions. We provide a theoretical convergence analysis for WHALE-FL with heterogeneous subnetwork assignment, based on which subnetwork structures can be dynamically optimized to reduce the resulting gap to standard full-model FL. Our evaluation shows that, compared with peer designs, WHALE-FL effectively accelerates FL training without sacrificing learning accuracy.**

***Index Terms*—Federated learning, wireless network, subnetwork training, device heterogeneity, training dynamics.**

L. Li is with the Frontier Research Center, Pengcheng Laboratory, Shenzhen, China (e-mails: lil03@pcl.ac.cn).

J. Geng, X. Qin and Y. Hou are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China, (e-mail: {lelegjx, xiaoqiqin, houyanzhao}@bupt.edu.cn).

H. Su, X. Fu and M. Pan are with the Electrical and Computer Engineering Department, University of Houston, Houston, TX 77004 USA (e-mails: hsu4@cougarnet.uh.edu, xfu8@central.uh.edu, mpan2@uh.edu).

H. Wang is with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: hwang9@stevens.edu).

The work of L. Li was supported in part by National Natural Science Foundation of China under grant 62201071; in part by the Basic and Frontier Research Project of PCL under grant 2025QYB041. The work of H. Su and M. Pan was supported in part by the US National Science Foundation under grants CNS-2107057, CNS-2318664, CSR-2403249, and CNS-2431596. The work of X. Qin was supported by Beijing Nova Program 2024102. The work of Y. Hou was supported in part by National Science and Technology Major Project-Mobile Information Networks under Grant No.2025ZD1303200. (*Corresponding author: Miao Pan*)

I. Introduction

Federated Learning (FL) [1] recently experienced a notable evolution, expanding its scope from conventional data center environments to harness the potential of mobile devices [2]–[4]. This shift has been propelled by the continuous advancements in hardware, empowering mobile devices like the NVIDIA AGX, iPhone 16, and MacBook Pro, etc. with increasingly robust on-device computing capabilities tailored for local training. With the collective intelligence of ending edge devices and FL's fundamental principle of preserving data privacy, FL over mobile devices has paved the way for a diverse spectrum of innovative mobile applications, including keyboard predictions [5], smart home hazard detection [6], health event detection [7], and so on.

While FL over mobile device has great potentials, its practical deployment faces significant challenges due to the inherent heterogeneity among real-world mobile devices, varying in computing capability, wireless conditions and local data distribution [8]. Existing FL studies often assume the model-homogeneous setting, where global and local models share identical architectures across all clients. However, as devices are forced to train models within their individual capability, developers have to choose between excluding low-tier devices, introducing training bias [9], or maintaining a low-complexity global model to accommodate all clients, resulting in degraded accuracy [10], [11]. The trend towards large models like Transformers [12] exacerbates the issue, hindering their training on mobile devices. Furthermore, unlike GPU clusters with stable high-speed Internet connections, mobile devices' computing resources are constrained and heterogeneous and their wireless transmissions are relatively slow and dynamic, both of which lead to huge latency in FL training [13], [14] and may severely degrade the performance of associated applications.

To address the limitations of model-homogeneous FL, researchers have recently studied how to train different sized models across heterogeneous mobile clients and corresponding global model aggregation in FL training. Subnetwork training, exemplified by pioneering approaches like width-based subnetwork generation in Federated Dropout [15] and HeteroFL [16], and depth-based generation in DepthFL [17], has proven effective by enabling mobile devices to train smaller subnetworks derived from the large global server model and offer solutions to aggregating diverse devices' subnetworks. By tailoring subnetwork size for the individual

device, subnetwork training can ensure compatibility with mobile devices owning heterogeneous computing and communications capability. However, a prevalent challenge in current subnetwork approaches lies in their static fixed-size subnetwork assignment policy. For example, once a subnetwork size is determined based on a device's capacity, Federated Dropout randomly samples channels in each layer according to this size, while HeteroFL selects the first consecutive set of channels to match the target width - and this size remains unchanged throughout the entire training process. Such a static policy may fail to unleash the full potential of subnetwork based training, mainly due to the unawareness of system dynamics (i.e., computing and communications dynamics) and FL training dynamics.

System dynamics encompass the time-varying computing loads of devices' background applications and the fluctuating wireless communication conditions across FL training rounds, which affects the sizes of subnetworks that a mobile device can support over rounds. Since most modern mobile devices (e.g., smartphones) participating in FL training have the ability to run multiple tasks (e.g., video streaming, image processing, and social media updates [18]) simultaneously, the dynamic orchestration of CPU/GPU resources across these concurrent activities results in the fluctuations in computing power and available memory for FL tasks, consequently impacting the supported subnetwork sizes for on-device computing. Similarly, wireless communications dynamics caused by users' mobility, wireless channel fading, etc. lead to dynamic transmission rates, which directly affect candidate subneworks sizes that a mobile device can support for local model updates.

FL training dynamics represents FL convergence's dynamic requirements for the contributions from local training at different training stages, which implicitly affects participating devices' selections on subnetwork sizes. Recent studies have revealed that critical learning periods (CLP) exist in the training process of deep neural networks [19], [20], which refers to specific phases during training when the neural network undergoes significant changes in how it learns and organizes information. Notably, the information in the weights does not increase monotonically during training. As FL training proceeds into the CLP, more accurate local model updates are needed for the global training model to converge. When FL training is close to convergence (i.e., the late stage), most mobile devices have already made substantial contributions to the global model. Thus, the adjustment of subnetwork sizes in FL is reasonable to align with such training dynamics.

We observe that failing to capture system or FL training dynamics and always using the possible largest-sized subnetworks under devices' full capabilities may significantly prolong FL training process. Different from prior static fixed-size subnetwork assignment methods, in this paper, we propose a wireless and heterogeneity aware latency efficient FL (WHALE-FL) approach to accelerate FL training via adaptive width-wise subnetwork scheduling. WHALE-FL characterizes system dynamics and FL training dynamics and tailors appropriate-sized subnetworks for heterogeneous mobile devices under dynamic computing/wireless environments at different FL training stages. As far as we know, WHALE-FL is the first paper that converts static fixed-size subnetwork allocation, e.g., HeteroFL [16], Federated Dropout [15], etc., into dynamic/adaptive subnetwork scheduling for each device by jointly considering system heterogeneity and FL training dynamics, and conducts system level experiments for validation. Our salient contributions are summarized as follows.

- We design a novel subnetwork selection utility function to capture system and FL training dynamics, guiding mobile devices to adaptively size their subnetworks for local training based on the time-varying computing/communication capacity and FL training status.
- We provide a theoretical convergence analysis for WHALE-FL with heterogeneous subnetwork assignment, based on which subnetwork structures can be dynamically optimized to reduce the resulting gap to standard full-model FL.
- We develop a WHALE-FL prototype and evaluate its performance with extensive experiments. The experimental results validate that WHALE-FL can remarkably reduce the latency for FL training over heterogeneous mobile devices without sacrificing learning accuracy.

The remainder of the paper is organized as follows. Section II reviews the literature and states our design goals. Section III introduces necessary preliminaries. Section IV elaborates on our WHALE-FL design. Section V introduces the setup of our experiments and Section VI presents the evaluation results. Section VII finally concludes the paper.

II. Related Work

Heterogeneous FL often faces the straggler issue, where a few low-end devices may significantly prolong the convergence process. Early approaches, such as FedProx [21] and Oort [22], aimed to mitigate the impact of system heterogeneity by improving federated optimization or incorporating client selection strategies. However, these methods assumed that all local models share the same architecture as the global model, resulting in identical numbers of local training parameters and uniform transmission costs across all clients. In practice, client devices often possess highly diverse computational and communication resources. Enforcing the same model architecture excludes resource-constrained devices from participation, limiting both inclusivity and potential contributions from their local data. With the growing trend toward larger models, model-homogeneous FL becomes increasingly impractical due to the inability of many devices to support such sizes.

Recent studies have empirically demonstrated the feasibility of using heterogeneous client models in FL to alleviate straggler effects. This has drawn substantial attention to a class of FL algorithms that train reduced-size heterogeneous local models - often derived by pruning a shared global model or extracting subnetworks - for global aggregation. For instance, methods incorporating pruning into FL, such as PruneFL [23], proposed adaptive parameter pruning during training, while FedMask [24] introduced FL with personalized, structured sparse masks. Similarly, Hermes [25] applies structured pruning to identify small local models, and FedPE [26] leverages personalized local models with pruning and error compensation to address heterogeneity and reduce communication costs.

While model pruning clearly reduces transmission overheads, its benefit on computational overhead saving is hard to verify in practice, as zeroed parameters, typically generated using binary masks applied to dense parameters in most existing implementations, still contribute to computations [23].

Another line of research in model-heterogeneous FL focuses on subnetwork training. Pioneering studies like HeteroFL [16] and FjORD [27] proposed generating heterogeneous local models by statically extracting sub-models from the global network. Dynamic subnetwork extraction methods have also been explored. For example, FedDropout [15] extracts subnetworks randomly, while FedRolex [28] employs rolling subnetwork extraction within the full model. FIARSE [29] dynamically adjusts subnetworks based on parameter importance for heterogeneous clients, and FedDSE [30] allows clients to extract neurons based on their activation over local datasets to form personalized subnetworks. While these subnetwork selection approaches effectively address device heterogeneity, they remain unaware of challenges posed by dynamic variations in computational resources, communication conditions, and FL training progress. Moreover, most existing works demonstrate convergence through experiments but lack rigorous theoretical convergence analysis. These limitations highlight the need for an adaptive subnetwork scheduling design that captures FL training dynamics, recognizes computing/communication constraints, and selects appropriately sized subnetworks for local training, to accelerate FL training with theoretical convergence guarantees.

III. Preliminary

A. FL over Heterogeneous Mobile Devices

Consider that M mobile devices in a wireless network collaboratively engage in FL to train a deep neural network on locally distributed datasets $\{\mathcal{D}_1, \cdots, \mathcal{D}_i, \cdots, \mathcal{D}_M\}$. The objective of FL is to learn a global model by minimizing the global loss function $F(\theta) \triangleq \frac{1}{M}\sum_{i=1}^{M} F_i(\theta)$ where $F_i(\theta)$ is the local loss function of client i defined over its dataset $\mathcal{D}_i$. Each device maintains a local model W_i, which is updated by applying stochastic gradient descent (SGD) [31] to its parameters θ_i on the data samples through local training. Here, we use $W_{i,r}$ to represent the local model structure of client i in the round r, while $\theta_{i,r}$ denotes the trainable parameters (weights) of its model. The server collects the local model updates and aggregates them into a global model $W_{g,r}$ [1], [32]. This aggregation occurs over multiple communication rounds. In the subsequent training round, the global model $W_{g,r}$, instantiated with the aggregated parameters θ_r, is transmitted to the mobile devices, based on which the devices update their local models. This process repeats until FL converges, while system heterogeneity (communications and computing) among mobile devices incurs huge training latency and slows down FL convergence.

B. FL with Subnetwork Extraction

To address system heterogeneity issue in FL training, subnetwork method was introduced in [16], which extracts differently sized subnetworks from a global model.

Let $\mathcal{W} = \{W^1, W^2, \cdots, W^p, \cdots, W^P\}$ be a collection of candidate subnetworks to be selected by mobile devices for local training, where P complexity/size levels are considered. Here, a lower size level p corresponds to a larger-sized subnetwork, and W^P is the smallest subnetwork for selection, i.e., $W^P \subset W^{P-1} \subset \cdots \subset W^1$. We follow the same approach as illustrated in [16] to extract subnetworks from the global model by shrinking the width of hidden channels with specific ratios. Let $s \in (0, 1]$ be the hidden channel shrinkage ratio. For a single hidden layer with output channel size out_g and input channel size in_g, the sizes for the subnetwork at level p are given by $out^p = s^{p-1}out_g$ and $in^p = s^{p-1}in_g$. It then follows that the size of the local model parameters is $|W^p| = s^{2(p-1)}|W_g|$, and the model shrinkage ratio is $|W^p|/|W_g| = s^{2(p-1)}$. For instance, when we set the hidden channel shrinkage ratio to $s = \frac{1}{2}$ and use $P = 5$ subnetwork size levels. The width shrinkage ratios for the five levels are 1, $\frac{1}{4}$, $\frac{1}{16}$, $\frac{1}{64}$, and $\frac{1}{256}$, respectively. This means that each layer of a subnetwork is pruned from that of the original global model by a given shrinkage ratio. This enables the assignment of appropriately sized models to participating mobile devices with varying capabilities. During model aggregation, the server combines these heterogeneous subnetworks by averaging each parameter only over the devices whose assigned subnetwork contains that parameter. This approach supports FL training with variable-sized local models.

Although the subnetwork method in [16] alleviates system heterogeneity issue, it employs a fixed assignment policy. It cannot adapt to the dynamic changes in wireless transmission/on-device computing conditions, or the dynamic requirements of contributions from local training at different FL training stages, either of which may result in a huge training latency.

C. Fisher Information

Fisher information is utilized as a measurement of how much a change in weights can affect the output of neural networks [19]. It is essentially a second-order approximation of the Hessian of the loss function [33], [34], providing information on the curvature of the loss landscape near the current weights. Such characteristics help to indicate how fast the gradient changes during training, which may be used to characterize the FL training dynamics from device side and further help clients decide how to adjust their subnetworks.

To enable distributed subnetwork scheduling, we use Federated Fisher Information Matrix (FedFIM) from [20] instead of the traditional definition of the Fisher Information Matrix for centralized training to avoid requiring access to the entire dataset. Given that training data resides in each client, the gradient of the loss function for a sample (x, y) is calculated by $\bigtriangledown(x, y) = \frac{\partial}{\partial\theta}\ell(x, y; \theta)$, which is parameterized by its local model weights θ. Then the FedFIM for client i in the r-th training round is defined by

$$f_{i,r} = \mathbb{E}_{x_i\sim\mathcal{X}_i}\mathbb{E}_{\hat{y}_i\sim\mathbb{P}_{\theta_{i,r}}(\hat{y}_i|x_i)}[\bigtriangledown(x_i, \hat{y}_i) \bigtriangledown (x_i, \hat{y}_i)^\intercal], \quad (1)$$

where x_i and y_i denote the input data and its corresponding label for client i, $\mathcal{X}_i$ represents the empirical distribution of the

i-th client's local data, and $\hat{y}_i$ is a random variable, rather than a true label, following an approximate posterior distribution $\mathbb{P}_{\theta_{i,r}}(\hat{y}_i|x_i)$.

IV. WHALE-FL Design

Aiming to reduce FL training latency, WHALE-FL entitles mobile devices to distributedly extract subnetworks with appropriate sizes for local training, adapting to their system dynamics and FL training dynamics. To capture those dynamics, WHALE-FL presents a novel adaptive subnetwork selection utility function jointly considering system efficiency and FL training efficiency. Moreover, WHALE-FL provides a normalization procedure to convert the calculated subnetwork selection utility values to discrete size levels of subnetowrks for mobile devices' local scheduling decisions. Building on insights from our theoretical convergence analysis, WHALE-FL also integrates a subnetwork extraction method that optimizes subnetwork structures for each mobile device in every FL training round, aiming for faster and better convergence.

A. Adaptive Subnetwork Selection Utility

WHALE-FL's adaptive subnetwork selection performance hinges on two critical aspects: *system efficiency* and *training efficiency*. System efficiency encompasses the duration of each training round, including local computing and model updates time consumption. Training efficiency gauges the local training's contributions to global convergence. The fluctuating wireless conditions and available computing resources of devices, as well as their training progress with local data, collectively determine the system and training efficiency, forming what we term as adaptive subnetwork selection utility.

To accelerate FL training without sacrificing learning accuracy, it is critical to balance system and training efficiencies when selecting the appropriate subnetwork size for each device's local training in each round. When system efficiency is high, indicating strong computational and transmission capabilities, the focus shifts to optimizing training efficiency for subnetwork size adjustments. At the early stage of FL training, WHALE-FL schedules small-sized subnetworks for devices' local training to conserve resources. As FL training enters the CLP, where more precise local training is required for convergence, WHALE-FL adapts by increasing the subnetwork size for participating mobile devices. When FL is close to convergence, WHALE-FL gradually reduces the size of subnetworks for local training, given the fact that most devices have contributed enough to global model and it is unnecessary to keep large-sized subnetworks for local training. Conversely, when system efficiency is low, indicating that a device experiences poor computation and communication conditions, WHALE-FL jointly considers system and training efficiencies, particularly when increasing subnetwork size, to avoid prolonging computation and transmission delays.

System efficiency utility. We define the system efficiency utility ($SE_{i,r}$) for any given client i in the r-th round based on its wireless transmission rate and available computing resources at that time, which is calculated as follows:

$$SE_{i,r} = \frac{\Delta}{T^{tr}_{i,r} + T^{co}_{i,r}}. \tag{2}$$

Here, $T^{tr}_{i,r}$ and $T^{co}_{i,r}$ are the transmission delay and the computing delay, respectively, for the unit/smallest subnetwork. Δ is the developer-preferred duration of each round, which may vary for different FL tasks. We assume that the wireless transmission rates and available computing resources dynamically change over rounds, but are relatively stable within a FL training round. Thus, given a learning task, transmission and computing workloads for the unit subnetwork are fixed, and $T^{tr}_{i,r}$ and $T^{co}_{i,r}$ can be easily estimated for device i in the r-th round. A higher $SE_{i,r}$ enables devices to opt for larger subnetwork sizes for local training within this round, and vice versa. The formulation in Eq. (2) comprehensively covers the system efficiency for communication delay dominant cases (i.e., slow transmissions & fast computing), computing delay dominant cases (i.e., fast transmissions & slow computing), and communication-computing comparable cases.

Training efficiency utility. By employing FedFIM, we define the training efficiency utility $TE_{i,r}$ for device i in the r-th round as follows:

$$TE_{i,r} = B_i \sqrt{\frac{1}{D}\sum_{d=1}^{D}\sum_{t=1}^{T} f^2_{i,r-q,t}}, \tag{3}$$

where B_i represents the local batch size for device i and $f_{i,r-q,t}$ denotes the FedFIM for device i at the t-th local iteration of round $r-q$. Here, we utilize a window-averaged FedFIM to measure the dynamic utility during training with $\{1, .., d, .., D\}$ as the set of window sizes. The sliding window operation helps to prevent frequent zigzag changes in subnetwork sizes, as Fisher information across different local training iterations may be highly unstable, and directly using the Fisher information of each iteration could result in unstable subnetwork selection strategies [19].

Adaptive subnetwork selection utility function. WHALE-FL trades-off the system and training efficiencies to determine the utility values for subnetwork scheduling over rounds. The adaptive subnetwork selection utility function is given by:

$$Util(i,r) = \underbrace{B_i \sqrt{\frac{1}{D}\sum_{d=1}^{D}\sum_{t=1}^{T} f^2_{i,r-q,t}}}_{\textit{Training efficiency utility}} \times \underbrace{\left(\frac{\Delta}{T_{i,r}}\right)^{\mathbb{1}(T_{i,r}>\Delta)\times\beta}}_{\textit{System efficiency utility}}, \tag{4}$$

where $T_{i,r} = T^{tr}_{i,r} + T^{co}_{i,r}$ is the duration of client i in the r-th round. A large/small value of $Util(i,r)$ suggests that device i should opt for a large/small sized subnetwork in the subsequent r-th round. $Util(i,r)$ associates system and training efficiencies with an on-off trade-off term $\mathbb{1}(T_{i,r} > \Delta) \times \beta$, where $\mathbb{1}(\cdot)$ is an indicator function that takes value 1 if the condition in $\{\cdot\}$ is true and 0 otherwise. This penalizes the utility of devices that might become bottlenecks for the current round's duration by a developer-specified factor $\beta(\beta > 0)$. The longer the transmission/computing delay, the greater the penalty, and the smaller the corresponding utility value. That is, when a device experiences a round duration $T_{i,r}$ longer than the average due to due to unexpected poor communication or computing conditions, its system

efficiency utility is scaled to a value between 0 and 1, thereby reducing the overall utility $Util(i,r)$. A larger β implies a stronger penalty. According to our design, such a device will consequently choose a smaller subnetwork in the subsequent round. By contrast, for non-straggler devices, their utilities are not penalized (system efficiency utility = 1) and depend solely on their training dynamics, as their completion times do not prolong the FL training round. Here, β rescales the system efficiency to align with the scale of training efficiency, facilitating an effective trade-off between the two.

B. Utility Value to Subnetwork Size Conversion

The calculated utility in Eq. (4) cannot directly be used by individual mobile device to decide its subnetwork size selection. To facilitate mobile devices' decisions, it is necessary to convert subnetwork selection utility values into available/candidate subnetwork sizes.

Given definitions above, the next step is to normalize devices' utility values into the range of $[0,1]$, in order to identify the model shrinkage ratio. We propose to use a piecewise linear function to normalize $Util(i,r)$ into $U_n(i,r)$ as follows.

$$U(i,r) = \begin{cases} \frac{Util(i,r)}{U_{th}}, & Util(i,r) \leq U_{th}, \\ 1, & \text{otherwise}, \end{cases} \tag{5}$$

where U_{th} is a configurable threshold that represents the utility level at which the full-sized model should be adopted.

After the utility value normalization, device i selects its subnetwork for the r-th round local training by

$$W_{i,r} = \begin{cases} \hat{W}_{i,r}, & \textbf{if } |W_i^{max}| > |\hat{W}_{i,r}|, \\ W_i^{max}, & \textbf{if } |W_i^{max}| \leq |\hat{W}_{i,r}|, \end{cases} \tag{6}$$

and its subnetwork size level is p satisfying $W_{i,r} = W^p, \forall W^p \in \mathcal{W}$. Here, $|W_i^{max}|$ denotes the maximum subnetwork size that device i can support with its full computing capacity, where $W_i^{max} \in \mathcal{W}$ as defined in Sec. III-B. $\hat{W}_{i,r} \in \mathcal{W}$ is a subnetwork derived from normalized utility value $U(i,r)$, which can be expressed as

$$\hat{W}_{i,r} = \begin{cases} W^1, & \textbf{if } U(i,r) \geq \frac{(P-1)}{P}; \\ W^2, & \textbf{if } U(i,r) \in [\frac{(P-2)}{P}, \frac{(P-1)}{P}); \\ \cdots, & \cdots \\ W^p, & \textbf{if } U(i,r) \in [\frac{(P-p)}{P}, \frac{(P-p+1)}{P}); \\ \cdots, & \cdots \\ W^P, & \textbf{if } U(i,r) < \frac{1}{P}, \end{cases} \tag{7}$$

where $|W^p|/|W_g| = s^{2(p-1)}, \forall W^p \in \mathcal{W}$.

Then, mobile devices conduct local computing according to their selected subnetworks, respectively, followed by transmitting local model updates to FL server. FL server aggregates updated local models with heterogeneous subnetworks and updates the global model according the rule specified in Sec. III.B,

In summary, during FL training, mobile devices collect their local information at runtime, including uplink channel quality, background computational loads, memory usage, training loss, etc. Based on the collected information, at the beginning of the r-th training round, each device leverages Eq. (4) to trade-off system efficiency and training efficiency, and calculates its adaptive subnetwork selection utility value $Util(i,r)$. The utility value is then normalized into $U(i,r)$. Device i uses $U(i,r)$ to determine the subnetwork size and select an appropriate subnetwork for its local training according to Eq. (6) and Eq. (7). Each device send the subnetwork index (an integer from 1 to P) to indicate its choice along with its model updates to the FL server in each communication round. After that, the server aggregates locally trained subnetworks with different sizes and updates the global model for the next round training. Note that, in our WHALE-FL, each device selects a subnetwork from a set of predefined levels by width-wise contraction, informing the server of its choice via a single scalar index that incurs negligible communication overhead. The pseudocode of WHALE-FL procedure is provided in Algorithm 1.

Algorithm 1 WHALE-FL Procedure

Input: Datasets $\{\mathcal{D}_1, \ldots, \mathcal{D}_M\}$ distributed over M mobile devices; number of local iterations T; learning rate η; channel shrinkage ratio s; number of subnetwork size levels P; local mini-batch size $B_i, \forall i$; maximum subnetwork size $W_i^{\max}, \forall i$; target per-round duration Δ; window size D; utility threshold U_{th}; hyperparameter β.
Initialize global model $W_{g,0}$
for communication round $r = 0, 1, 2, \cdots$ **do**
  for each client $i \in [M]$ **in parallel do**
    $p_{i,r} \leftarrow \text{SUBNETSELECT}(\Delta, \beta, D, B_i, W_i^{max}, P)$
    $in^i \leftarrow s^{p_{i,r}-1} in_g, out^i \leftarrow s^{p_{i,r}-1} out_g$
    $W_{i,r} \leftarrow W_{g,r}[: in^i, : out^i]$ /* Width-shrinking-based subnetwork extraction */
    $W_{i,r+1} \leftarrow \text{CLIENTUPDATE}(p_{i,r}, W_{i,r})$
  for each size level $p \in [P]$ **do**
    Aggregate local parameters of subnetwork W_{r+1}^p from all client with $p_{i,r} \leq p$
  Update global model $W_{g,r+1}$ by composing $\{W_{r+1}^p\}_{p=1}^P$
function SUBNETSELECT($\Delta, \beta, D, B_i, W_i^{\max}, P$)
  if $r = 0$ **then**
    $p_{i,r} \leftarrow 1$
  else
    Compute $Util(i,r)$ via Eq. (4)
    Compute normalized utility $U(i,r)$ via Eq. (5)
    Quantize model size $\hat{W}_{i,r}$ via Eq. (7)
    Determine subnetwork size level $p_{i,r}$ via Eq. (6)
  Return $p_{i,r}$
function CLIENTUPDATE($p_i, W_{i,r}$)
  for local ieteration t from 1 to T **do**
    Sampling mini-batch of size B_i from $\mathcal{D}_i$
    Update the local model $W_{i,r}$ via SGD
  $W_{i,r+1} \leftarrow W_{i,r}$
  Return $W_{i,r+1}$

C. Convergence Analysis

In this subsection, we present a thorough convergence analysis of the proposed WHALE-FL under an arbitrary subnetwork scheduling policy. Specially, our analysis is established under the scenarios of heterogeneous FL with non-convex objectives, where local subnetwork sizes may vary across mobile devices in any given training round and may also change across rounds for individual devices.

To facilitate the theoretical analysis, we introduce a notion of neural region that is defined by

$$S^P = W_g^P, \tag{8a}$$

$$S^p = W_g^{p-1} \backslash W_g^p, \forall p \in [2, P], \tag{8b}$$

$$\mathcal{S} = \{S^1, S^2, ..., S^p, ..., S^P\}. \tag{8c}$$

Here, S^p for $p = 1, 2, ..., P$ represents non-overlapping neural regions such that a subnetwork at any width level p can be constituted by a set of neural regions $\{S^P, S^{P-1}, ..., S^1\}$. Let $\mathcal{S}_r$ be the set of the neural regions trained in round r, where $\mathcal{S}_r \subset \mathcal{S}$. We further denote by $\mathcal{M}_r^p$ the set of clients whose subnetworks train parameters in the neural region $S^p \in \mathcal{S}_r$ in round r and $|\mathcal{M}_r^p|$ be the number of clients in $\mathcal{M}_r^p$. For the ease of theoretical analysis, we further define $|\mathcal{M}^*| = min_{r,p}|\mathcal{M}_r^p|,\ p \in \mathcal{S}_r,\ \forall r$, which measures the minimum occurrence of any neural region $S^p, \forall p$ across all rounds r. Intuitively, a larger $|\mathcal{M}^*|$ implies more sufficient training for all neural regions, as it reflects a higher minimum level of client contribution for any neural region. In any training round r, clients' subnetworks are extracted based on the initial global model θ_r by using masks, where $m_{i,r}$ represents the subnetwork mask for client i in the r-th round.

With the above notations, we begin with making several widely-adopted assumptions as follows.

Assumption 1 (Mask-induced noises)**.** *There exists $\omega \in [0, 1)$ such that the mask-induced noise is bounded by:*

$$||\theta_r - \theta_r \odot m_{r,i}||^2 \leq \omega^2 ||\theta_r||^2, \quad \forall i, r. \tag{9}$$

Assumption 2 (Smoothness Condition)**.** *Loss function $F(\cdot)$ is with L-smoothness:*

$$\begin{aligned}&\mathbb{E}[F(\theta_{r+1})] - \mathbb{E}[F(\theta_r)] \\ \leq &\mathbb{E}[\langle \nabla F(\theta_r), \theta_{r+1} - \theta_r \rangle] + \frac{L}{2}\mathbb{E}[||\theta_{r+1} - \theta_r||^2], \quad \forall r.\end{aligned} \tag{10}$$

Assumption 3 (Bounded variance)**.** *There exists $\sigma > 0$ satisfying:*

$$\mathbb{E}_{\xi_{i,t} \sim \mathcal{D}_i} ||\nabla F_i(\theta_{i,r,t}; \xi_{i,t}) - \nabla F_i(\theta_{i,r,t})||^2 \leq \sigma^2, \quad \forall i, r, t, \tag{11}$$

where $\xi_{i,t}$ is the data batch sampled from $\mathcal{D}_i$ in iteration t.

Assumption 4 (Bounded data heterogeneity level)**.** *There exists $\delta > 0$ satisfying:*

$$||\nabla F_i(\theta_r) - \nabla F(\theta_r)||^2 \leq \delta^2, \quad \forall r. \tag{12}$$

In particular, Assumption 1 is a standard and common setting assuming a smooth loss function. Assumption 2 follows from [35], which implies the noise introduced by subnetwork is bounded and quantified. Assumptions 3 is standard for FL convergence analysis and assume the stochastic gradients to be bounded and unbiased. Assumptions 4 assumes the differences between local gradients and global gradients are bounded, which is required for heterogeneous FL to converge to a stationary point of standard FL.

Theorem 1. *Let all assumptions hold. Suppose that the step size γ satisfies $0 \leq \gamma \leq \min\left\{\frac{1}{12TL}, \frac{|\mathcal{M}^*|}{16TL\sqrt{M}}, \left(\frac{\sqrt{|\mathcal{M}^*|}}{768T^3L^3M}\right)^{1/3}\right\}$. Then, for all $R \geq 1$, we have:*

$$\begin{aligned}&\frac{1}{R}\sum_{r=1}^{R}\sum_{S^p \in \mathcal{S}_r} \mathbb{E}\left[||\nabla F^p(\theta_r)||^2\right] \\ \leq &\frac{8}{RT\gamma}\left(\mathbb{E}[F(\theta_1)] - \mathbb{E}[F(\theta_{R+1})]\right) \\ &+ \frac{32\omega^2L^2M + 48L^3\gamma TM}{|\mathcal{M}^*|R}\sum_{r=1}^{R}\mathbb{E}\left[||\theta_r||^2\right] \\ &+ \frac{8\delta^2 M}{|\mathcal{M}^*|}\left(32\gamma^2T^2L^2 + 1 + 96L^3\gamma^3T^3 + 3L\gamma T\right) \\ &+ \frac{8\gamma L\sigma^2 M}{|\mathcal{M}^*|}(4\gamma TL + \frac{3}{2} + 12L^2\gamma^2T^2).\end{aligned} \tag{13}$$

Proof: See Appendix A for the proof. ■

Theorem 1 demonstrates the convergence rate of the WHALE-FL algorithm by providing an upper bound on the average gradient of all clients across all trained parameters. WHALE-FL relaxes the constraint that all model parameters must be trained in every round. The results shows that WHALE-FL can converge under arbitrary adaptive subnetwork size scheduling. Specifically, larger $|S_r|$ values for each training round r lead to more bounded gradients in the trained neuron regions, which help improve the convergence rate. Besides, except for the non-trained parameters from the global model, others can be trained by at least $|\mathcal{M}^*|$ subnetworks in each round. As $|\mathcal{M}^*|$ increases, the model parameters are trained more frequently and sufficiently, allowing WHALE-FL to converge to a stationary point more quickly.

Discussion on WHALE-FL's scalability. In terms of implementation complexity, the adaptive subnetwork scheduling method has a theoretical complexity of $O(B_i \times D)$, ensuring low overhead. Besides, our design allows clients to adjust their subnetworks locally based on computation and communication time measurements, which doesn't require developers to understand the global dynamic resource allocation strategies, thereby facilitating its implementation at scale. From the perspective of theoretical convergence, the effect of increasing the number of clients on convergence speed in our WHALE-FL is broadly consistent with that in vanilla FL algorithms. By mitigating the straggler effect through adaptive subnetwork scheduling, WHALE-FL has the potential to enhance scalability in heterogeneous device environments. To support extension to larger settings, our approach remains compatible with systemic optimizations such as participant selection, resource allocation, batch size control, and more, and can benefit from these complementary improvements.

D. Subnetwork Extraction Optimization

Inspired by the theoretical findings above, we introduce a more flexible subnetwork extraction method, given the subnetwork size level $p_{i,r}, \forall i, r$ obtained in Sec. IV-B. Note that this method is compatible the overall workflow of WHALE-FL as outlined in Algorithm 1 and can serve as an alternative to the fixed width-shrinking-based extraction rule, in which each size level corresponds to a predefined subnetwork structure in the set $\mathcal{W}$.

In each FL round, we prefer subnetwork extraction strategies with a large minimum covering index $|\mathcal{S}_r|$ to reduce the optimality gap to standard full-model FL, while those important parameters with high importance are better trained to promote convergence. Let ϕ^p be the parameter size of the neural region S^p. We introduce a binary variable $x_{i,r}^p$ where $x_{i,r}^p = 1$ indicates if client i includes neural region S^p into its subnetwork for the r-th round local training; otherwise, $x_{i,r}^p = 0$. Notice that, for any client i in round r, all the selected neural regions then form its subnetwork. Let f_r^p be the FIM for neural region S^p of the global model at round r, and let $f_r^{p,max}/f_r^{p,min}$ represent the maximum/minimum FIM of S^p among all clients. For every round r, we formulate a subnetwork extraction optimization problem in a global view as follows:

$$\max_{x_{i,r}^p} \quad \min_{i,p}\Big(\sum_{r'\le r} x_{i,r'}^p\Big)\cdot\frac{f_r^p - f_r^{p,min}+\epsilon}{f_r^{p,max}-f_r^{p,min}} \tag{14a}$$

$$s.t. \quad \left|\sum_p x_{i,r}^p \phi^p\right| \le |W^{p_{i,r}}|, \forall i, \tag{14b}$$

$$x_{i,r}^p \in \{0,1\}, \forall i, p. \tag{14c}$$

Here, the term $\min_{i,p}(\sum_{r'\le r} x_{i,r'}^p)$ in (14a) represents the minimum number of times that any neural region S^p has been locally trained by any mobile device over the previous training rounds, and maximizing it essentially increases the minimum covering index $|\mathcal{S}_r|$. The relative Fisher information term $\frac{f_r^p - f_r^{p,min}+\epsilon}{f_r^{p,max}-f_r^{p,min}}$ encourages the inclusion of important parameters with high Fisher information, i.e., large values of f_r^p, into the subnetworks, where ϵ is a small positive constant that guarantees the numerator remains positive. Constraint (14b) ensures that subnetwork size of any mobile device is restricted to the size level determined by evaluating its utility value, as specified in Sec. IV-B.

To address the above problem, we first construct an N-by-P matrix Λ^r for the r-th round whose elements are $\lambda_{i,r}^p \cdot \frac{f_r^p - f_r^{p,min}+\epsilon}{f_r^{p,max}-f_r^{p,min}}$, where $\lambda_{i,r}^p$ is calculated by $\lambda_{i,r}^p = \sum_{r'<r} x_{i,r'}^p$. We then employ a greedy search to determine subnetworks for the M clients in a round-by-round manner. The basic idea is to optimize the strategy by considering the overall subnetwork covering region across devices and their training contributions in the previous FL rounds, under each device's subnetwork size constraints. The procedure is outlined in Algorithm 2.

Algorithm 2 Subnetwork Extraction Algorithm

1: **Input:** The initialized subnetworks of N mobile devices, the global FIM f_r^p in terms of every neural region S^p in each round i.
2: **for** each communication round $r = 0, 1, 2, \cdots$ **do**
3: Update the matrix Λ^r.
4: Sort the elements in each row of matrix Λ^r in descending order to generate M sequences.
5: **for** each sequence $i = 1, 2, \cdots, M$ **do**
6: Pick large value elements and set the corresponding $x_{i,r}^p \leftarrow 1$ until violating $\left|\sum_p x_{i,r}^p \phi^p\right| \le |W^{p_{i,r}}|$.
7: Return $x_{i,r}^p, \forall i, r, p$

V. Experimental Setup

A. WHALE-FL Testbed

The testbed consists of an FL aggregator and a set of heterogeneous mobile devices as FL clients. A NVIDIA RTX 3090 serves as the FL server, whose memory capacity is 24 GB. For heterogeneous FL clients, we have incorporated 5 types of mobile devices, i.e., MacBookPro2018, NVIDIA Jetson Xavier, NVIDIA Jetson TX2, NVDIA Jetson Nano, and Raspberry Pi 4, representing a range of on-device computing capabilities from high to low. The WHALE-FL system involves a total of 20 mobile devices, 4 devices per type. Communication between FL clients and the FL server is facilitated through Wi-Fi 5, LTE, and BlueTooth transmission environments, with the averaged transmission rates 80 Mbps (Wi-Fi 5), 20 Mbps (LTE), and 10 Mbps (BlueTooth 3.0), respectively. We leverage the inherent fluctuation in real transmission rates across different training rounds to emulate system dynamics. This dynamic setting was applied to evaluate the performance of both our method and all baselines. All results are averaged over 10 independent runs to ensure statistical reliability. We set hidden channel shrinkage ratio $s = \frac{1}{2}$ and adopt 5 subnetwork size levels. Accordingly, the model shrinkage ratios for the 5 size levels (i.e., $p = 1, 2, \cdots, 5$) are 1, $\frac{1}{4}$, $\frac{1}{16}$, $\frac{1}{64}$, and $\frac{1}{256}$, respectively.

B. Datasets, Models, Parameters and Baselines

We conduct our experiments with three different FL tasks: image classification, human activity recognition and language modeling. As for the image classification task, we train a CNN on MNIST dataset [36] and a ResNet18 on CIFAR10 dataset [37]. Human activity recognition involves training a CNN on the HAR dataset [38], and a Transformer is trained on the WikiText2 dataset [39] for the language modeling task. We use the balanced non-IID data partition [40]. Take the MNIST dataset as example, the total number of classes is 10. Our default setup is that each device has $\sigma = 2$ classes. We apply similar non-IID setup to other tasks. The Fisher information's window size $|\mathbf{D}|$ = 10. We employ the following peer designs for performance evaluation: (i) FedAvg [1], where all the clients train with full-sized models; (ii) FedProx [21], where all the clients train with full-sized models but add a proximal term to their local objective functions to enhance

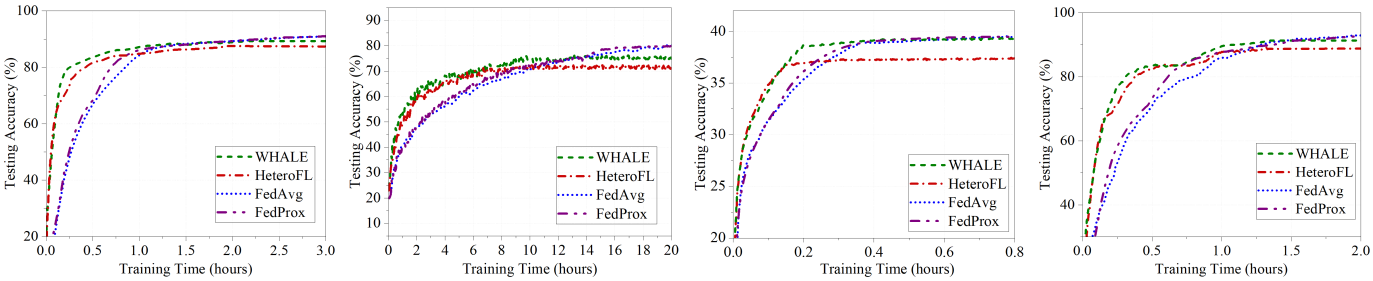


Fig. 1: Performance comparison of different FL training approaches under various learning tasks. Figures from left to right are CNN@MNIST, ResNet18@CIFAR10, Transformer@WikiText2, and CNN@HAR with non-IID datasets.

TABLE I: Performance comparison under different subnetwork methods (Speedup).

Task	CV		NLP	HAR
	CNN@MNIST	Resnet@CIFAR10	Transformer@Wikitext2	CNN@HAR
Target Accuracy	85%	70%	37%	88%
Method	Speedup			
WHALE-FL vs HeteroFL	1.79x	1.35x	1.33x	1.12x
WHALE-FL vs FedRolex	1.76x	1.33x	1.31x	1.11x
WHALE-FL vs FedDropout	1.84x	1.38x	1.35x	1.14x

TABLE II: Performance comparison under different subnetwork methods (Final Accuracy Improvement).

Task	CV		NLP	HAR
	CNN@MNIST	Resnet@CIFAR10	Transformer@Wikitext2	CNN@HAR
Method	Final Accuracy Improvement			
FedAvg	92.71%	80.61%	40.54%	92.94%
HeteroFL	87.42%	71.65%	37.40%	88.86%
FedRolex	87.82%	72.52%	38.02%	89.11%
FedDropout	86.16%	70.08%	37.19%	88.25%
WHALE-FL	**89.98%**	**79.81%**	**39.86%**	**92.42%**

the training efficiency; (iii) HeteroFL [16], where subnetwork assignments are fixed and align with clients' full computation and communication capabilities; (iv) FedDropout [15], which generates subnetworks by choosing the neurons at random; and (v) FedRolex [28], which extracts subnetwork in a rolling way across FL training rounds.

VI. Evaluation and Analysis

A. Time Efficiency and Learning Performance

As the results shown in Fig. 1, the proposed WHALE-FL consistently achieves remarkable training speedup across various FL tasks without sacrificing learning accuracy. Compared with FedAvg, WHALE-FL accelerates the FL training to the target testing accuracy by approximately 1.5x, 1.9x, 1.3x and 2.1x for FL tasks including CNN@MNIST, ResNet18@CIFAR10, Transformer@WikiText2, and CNN@HAR, respectively. As detailed in Sec. II, HeteroFL's static fixed-size subnetwork assignment policy is not aware of system and training dynamics, which may slow down FL convergence. In contrast, considering both system efficiency and training efficiency, WHALE-FL appropriately assesses the subnetwork selection utility for individual device and adaptively adjusts the local subnetwork size to suit for time-varying communication and computational conditions and dynamic changing requirements of FL training at different FL training stages, in order to prompts faster accuracy increments. Consequently, compared with HeteroFL, WHALE-FL achieves a notable speedup of 1.74x, 1.25x, 1.21x and 1.06x for the tested 4 learning tasks, respectively. Results in Tables 1 and 2 further demonstrate that WHALE-FL achieve faster convergence and better testing accuracy than the peer designs across different FL tasks. During the early and late stages of FL training, despite using the reduced subnetwork sizes, WHALE-FL demonstrates accuracy performance that is nearly indistinguishable from the baselines. This adaptability in our subnetwork scheduling to the time-varying communication and computational conditions allows improvements in system efficiency to compensate for, or even surpass, the slight training performance degradation introduced by smaller-sized local models.

B. Subnetwork Size and Training/System Efficiency Changes

As shown in Fig. 2, the subnetwork sizes adapt to the D-averaged variations in local Fisher information across three heterogeneous devices: MacBookPro 2018 (high-end), NVIDIA Jetson TX2 (medium-performance), and Raspberry Pi 4 (low-end). The results align with our expectations: when Fisher information is high, the subnetwork size increases to enhance the global model's accuracy; as training proceeds and Fisher information decreases, indicating that its impacts learning decreases, the subnetwork size is becoming smaller to improve training latency efficiency. On the server side, the averaged size of the aggregated local subnetworks changes along with global model's Fisher information, which exhibits a similar trend to the local Fisher information. Figure 2 demonstrates that WHALE-FL effectively captures training

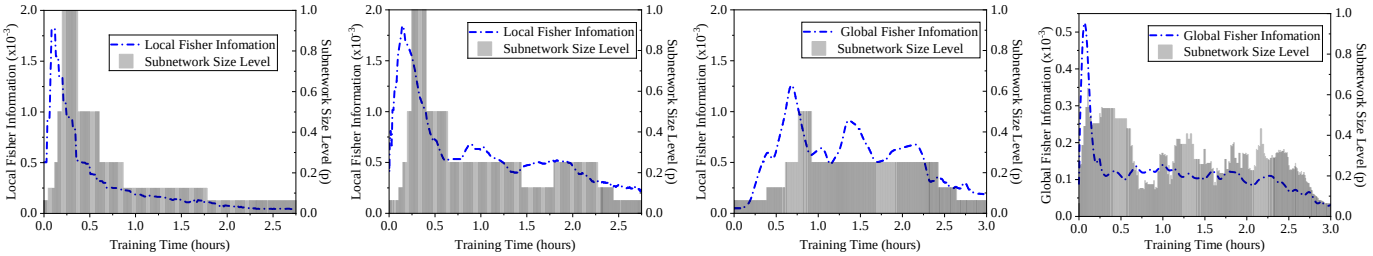


Fig. 2: Fisher information and subnetwork size level changes over training time (CNN@MNIST). From left to right, the performance of the user-side models on MacBookPro 2018, NVIDIA Jetson TX2, and Raspberry Pi 4, as well as the global model's performance, are shown.

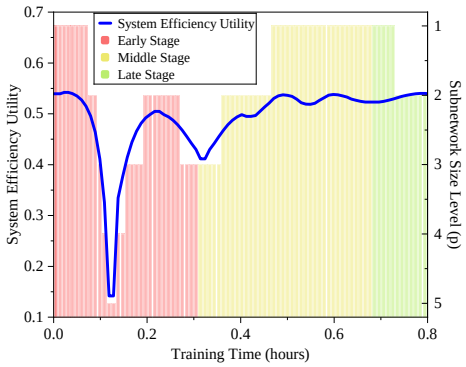


Fig. 3: System efficiency utility and subnetwork size level changes over training time (Macbookpro2018, CNN@HAR).

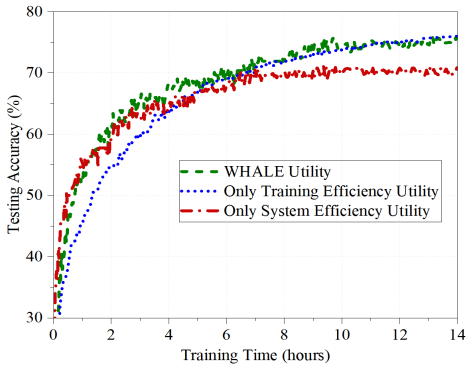


Fig. 4: Performance comparison of WHALE-FL, system-efficiency-only and training-efficiency-only designs (ResNet18@CIFAR10).

dynamics while selecting appropriate subnetwork sizes for heterogeneous devices.

To further illustrate this relationship, we also include a visualization showing how system dynamics influence our subnetwork scheduling strategy. Using the MacBook Pro 2018 (CNN@HAR task) as an example, we track the changes in subnetwork size under dynamic variations in system efficiency utility over the training period during 0-0.8h and present the corresponding results in Fig. 3. To clarify how the influence of system and training dynamics shifts over time and adaptively guides the subnetwork scheduling decisions, we roughly divide this training period into three stages: early stage, middle stage, and late stage. As shown in the figure, during the early training stage, the subnetwork size adapts primarily according to the system efficiency utility. Since FL training starts from scratch, contributions from any device - even with smaller subnetworks - are beneficial. Adapting subnetwork sizes to system conditions helps expedite local updates and improves communication efficiency. In the middle stage, the subnetwork size gradually increases, influenced by both system conditions and training efficiency. Notably, between 0.4h-0.6h, the subnetwork size expands rapidly even though system utility remains relatively stable, indicating the increased influence of training dynamics as the process enters the CLP. During the late stage, as convergence stabilizes, the subnetwork size is reduced to improve system resource efficiency by aligning with the system efficiency utility. Here, we take Macbookpro2018 for example, and the analysis applies to all participating mobile devices.

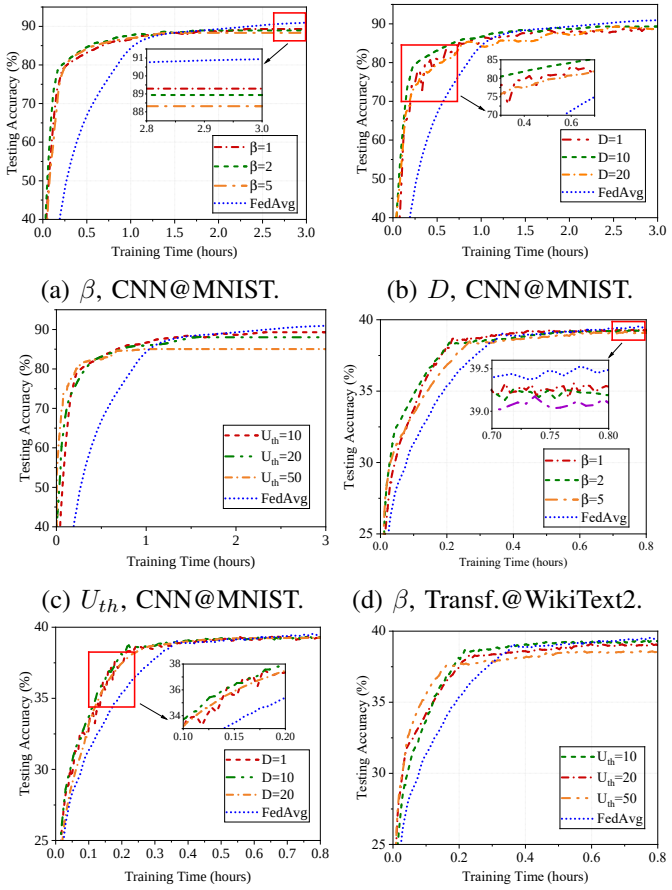
C. System Efficiency vs Training Efficiency

To differentiate system efficiency's contributions from training efficiency's ones, we compare WHALE-FL with system efficiency utility only and training efficiency utility only schedulings. As the results shown in Fig. 4, WHALE-FL converges faster than training efficiency only subnetwork scheduling when achieving the target accuracy, since training efficiency only design has no consideration of system dynamics and its impacts on subnetwork size selection; WHALE-FL has better testing accuracy but proceeds slower than system efficiency only subnetwork scheduling at the early training stage. The reason behind is that system efficiency only design prioritizes system dynamics while ignoring dynamic model accuracy requirements for local training at different FL training stages. WHALE-FL trades-off system and training efficiencies and jointly considers their benefits for FL training.

D. Sensitivity Analysis

Taking CNN@MNIST as an example, we conduct sensitivity analysis of its performance under different β, D, and U_{th} values, and present the results in Fig. 5(a)-(c). For generalization purpose, we have also conducted the sensitivity study for NLP task in Fig. 5(d)-(f).

The hyperparameter β trades-off system efficiency and training efficiency utilities. The large/small β value means that the device prioritizes system/training efficiency. As the results shown in Fig. 5(a) and Fig. 5(d), we find that the FL training



(a) β, CNN@MNIST. (b) D, CNN@MNIST.

(c) U_{th}, CNN@MNIST. (d) β, Transf.@WikiText2.

(e) D, Transf.@WikiText2. (f) U_{th}, Transf.@WikiText2.

Fig. 5: Sensitivity analysis under different β, D, and U_{th} values (a-c: CNN@MNIST; d-e: Transformer@WikiText2).

converges slower but achieves higher testing accuracy when β is small, e.g., $\beta = 1$, while FL training is faster at early stage but achieves lower testing accuracy when β is larger, e.g., $\beta = 5$. System efficiency and training efficiency are somehow balanced when $\beta = 2$. Thus, although β is a developer-specified factor, a proper selection of β helps FL training converge fast while achieving good learning performance.

The hyperparameter D represents the window size for calculating the averaged Fisher information. A small window size, such as $D = 1$ in Fig. 5(b) and Fig. 5(e), makes the subnetwork size updates sensitive to changes in Fisher information, leading to fluctuations in model accuracy during training. Accordingly, the subnetwork size grows rapidly in the early stages, causing the client to prematurely utilize a full-size model, which can decelerate convergence due to extended computation and communication delays. Conversely, employing a larger window size, like $D = 20$, results in slower subnetwork-size changes. This may cause a situation where a small-sized subnetwork is well-trained while the clients have no chances to switch to the larger sized subnetworks, thus impairing the training performance during the critical learning periods. A window size of $D = 10$ strikes a good balance, achieving faster convergence.

Similarly, a higher U_{th}, e.g., U_{th} = 50 shown in Fig. 5(c) and 5(f), leads clients to choose smaller subnetworks, which speeds up FL convergence in the early stages by reducing transmission and computation delays but results in lower final accuracy. Conversely, with U_{th} = 10, clients select larger subnetworks, which slows down convergence but yields higher accuracy. A proper U_{th} selection helps to balance learning performance and delay efficiency.

TABLE III: Performance comparison under different data heterogeneity (CNN@MNIST), where "SP" is the speedup.

Local Model	CNN@MNIST		
non-IID Level	$\sigma = 2$	$\sigma = 5$	$\sigma = 10$
Target Acc.	85%	90%	95%
Metric	Hours (SP)		
FedAvg	1.12 (1.00x)	0.33 (1.00x)	0.30 (1.00x)
HeteroFL	1.06 (1.06x)	0.26 (1.27x)	0.10 (3.00x)
FedDropout	1.09 (1.03x)	0.28 (1.18x)	0.11 (2.73x)
FedRolex	1.03 (1.09x)	0.25 (1.32x)	0.10 (3.00x)
WHALE	**0.58 (1.93x)**	**0.17 (1.94x)**	**0.07 (4.29x)**

E. Impacts of Data Heterogeneity

We further evaluate the impacts of data heterogeneity on WHALE-FL's performance. Here, we take CNN@MNIST as an example and use the balanced non-IID data partition [40]. The total number of classes in the MNIST dataset is 10. We study the cases that each device has $\sigma = 2, 5$ or 10 classes, where the data distribution is IID if $\sigma = 10$, i.e., every device has all classes. Following this setting, we generate the local dataset for each user by drawing the data from the whole dataset with specific labels. The results are shown in Table III, where we find that (i) FL training with non-IID data takes longer time to converge, and (ii) embracing both system and training efficiency utilities, WHALE-FL can remarkably improve FL training delay efficiency when applied to existing subnetwork methods under various data heterogeneity scenarios.

VII. Conclusion

In this paper, we have proposed WHALE-FL, a wireless and heterogeneity aware latency efficient federated learning approach, to accelerate FL training over mobile devices via adaptive subnetwork scheduling. Unlike existing static fixed-size subnetwork assignments, WHALE-FL has incorporated an adaptive subnetwork scheduling policy, enabling mobile devices to flexibly select subnetworks for local training, with a keen awareness of mobile devices' system dynamics and FL training dynamics. At its core, WHALE-FL has employed a well-designed subnetwork sizing function combined with a convergence-guided subnetwork extraction method to assign appropriate subnetworks for mobile devices in each FL training round, which captures changes in the device's system conditions (including available computing and communication capacities) and evolving FL training requirements for local training. Experimental results have demonstrated that WHALE-FL surpasses peer designs, significantly accelerating FL training over heterogeneous mobile devices without sacrificing learning accuracy.

APPENDIX

A. Proof of Theorem 1

From the smoothness condition of the objective function $F(\cdot)$, we have:

$$\mathbb{E}[F(\theta_{r+1})] - \mathbb{E}[F(\theta_r)] \leq \underbrace{\mathbb{E}[\langle \nabla F(\theta_r), \theta_{r+1} - \theta_r \rangle]}_{U_1} + \underbrace{\frac{L}{2}\mathbb{E}[||\theta_{r+1} - \theta_r||^2]}_{U_2}. \tag{15}$$

To bound U_1, we decompose the global and local contributions, and expand the local model updates for each participating client with different selections of the subnetworks. Specifically, we have:

$$\begin{aligned}
&\mathbb{E}[\langle \nabla F(\theta_r), \theta_{r+1} - \theta_r \rangle] \\
&= \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[\langle \nabla F^p(\theta_r), \theta_{r+1}^p - \theta_r^p \rangle] + \sum_{S^p \in \mathcal{S} \setminus \mathcal{S}_r} \mathbb{E}\langle \nabla F^p(\theta_r), \theta_{r+1}^p - \theta_r^p \rangle \\
&= \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[\langle \nabla F^p(\theta_r), \theta_{r+1}^p - \theta_r^p \rangle] + \sum_{S^p \in \mathcal{S} \setminus \mathcal{S}_r} \mathbb{E}\langle \nabla F^p(\theta_r), 0 \rangle \\
&= \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[\langle \nabla F^p(\theta_r), -\frac{1}{|\mathcal{M}_r^p|} \sum_{i \in \mathcal{M}_r^p} (\theta_{i,r,0} - \theta_{i,r,T}) \rangle] \\
&= \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[\langle \nabla F^p(\theta_r), -\frac{1}{|\mathcal{M}_r^p|} \sum_{i \in \mathcal{M}_r^p} (\theta_{i,r,0} - (\theta_{i,r,0} - \gamma \sum_{t=1}^{T} \nabla F_i(\theta_{i,r,t-1}, \xi_{i,t-1}) \odot m_{i,r})) \rangle] \\
&= \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[\langle \nabla F^p(\theta_r), -\frac{1}{|\mathcal{M}_r^p|} \sum_{i \in \mathcal{M}_r^p} (\gamma \sum_{t=1}^{T} \nabla F_i^p(\theta_{i,r,t-1})) \rangle] = \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[\langle \nabla F^i(\theta_r), -\frac{1}{|\mathcal{M}_r^p|} \sum_{i \in \mathcal{M}_r^p} (\gamma \sum_{t=1}^{T} \nabla F_i^p(\theta_{i,r,t-1})) \rangle] \\
&= \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[\langle \nabla F^p(\theta_r), -\frac{1}{|\mathcal{M}_r^p|} \sum_{i \in \mathcal{M}_r^p} \sum_{t=1}^{T} \gamma (\nabla F_i^p(\theta_{i,r,t-1}) - \nabla F^p(\theta_r) + \nabla F^p(\theta_r)) \rangle] \\
&= -\sum_{S^p \in \mathcal{S}_r} \gamma T \mathbb{E}[||\nabla F^p(\theta_r)||^2] + \underbrace{\sum_{S^p \in \mathcal{S}_r} \mathbb{E}[\langle \nabla F^p(\theta_r), -\frac{1}{|\mathcal{M}_r^p|} \sum_{i \in \mathcal{M}_r^p} \sum_{t=1}^{T} \gamma (\nabla F_i^p(\theta_{i,r,t-1}) - \nabla F^p(\theta_r)) \rangle]}_{U_3}.
\end{aligned} \tag{16}$$

Here, the term U_3 measures the deviation between the local gradient and the global gradient. Expanding U_3 yields:

$$\begin{aligned}
&\sum_{S^p \in \mathcal{S}_r} \mathbb{E}[\langle \nabla F^p(\theta_r), -\frac{1}{|\mathcal{M}_r^p|} \sum_{i \in \mathcal{M}_r^p} \sum_{t=1}^{T} \gamma (\nabla F_i^p(\theta_{i,r,t-1}) - \nabla F^p(\theta_r)) \rangle] \\
\overset{(a)}{\leq} &\frac{\gamma T}{2} \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\nabla F^p(\theta_r)||^2] + \frac{\gamma T}{2} \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\frac{1}{T|\mathcal{M}_r^p|} \sum_{i \in \mathcal{M}_r^p} \sum_{t=1}^{T} [\nabla F_i^p(\theta_{i,r,t-1}) - \nabla F^p(\theta_r)]||^2] \\
\overset{(b)}{\leq} &\frac{\gamma T}{2} \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\nabla F^p(\theta_r)||^2] + \gamma T \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\frac{1}{T|\mathcal{M}_r^p|} \sum_{i \in \mathcal{M}_r^p} \sum_{t=1}^{T} [\nabla F_i^p(\theta_{i,r,t-1}) - \nabla F_i^p(\theta_r)]||^2] \\
&+ \gamma T \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\frac{1}{T|\mathcal{M}_r^p|} \sum_{i \in \mathcal{M}_r^p} \sum_{t=1}^{T} [\nabla F_i^p(\theta_r) - \nabla F^p(\theta_r)]||^2] \\
\leq &\frac{\gamma T}{2} \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\nabla F^p(\theta_r)||^2] + \gamma T \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\frac{1}{T|\mathcal{M}_r^p|} \sum_{i \in \mathcal{M}_r^p} \sum_{t=1}^{T} [\nabla F_i^p(\theta_{i,r,t-1}) - \nabla F_i^p(\theta_r)]||^2] \\
&+ \frac{\gamma}{|\mathcal{M}^*|} \sum_{i=1}^{M} \sum_{t=1}^{T} \mathbb{E}[||[\nabla F_i(\theta_r) - \nabla F(\theta_r)]||^2] \\
\overset{(c)}{\leq} &\frac{\gamma T}{2} \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\nabla F^p(\theta_r)||^2] + \underbrace{\gamma T \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\frac{1}{T|\mathcal{M}_r^p|} \sum_{i \in \mathcal{M}_r^p} \sum_{t=1}^{T} [\nabla F_i^p(\theta_{i,r,t-1}) - \nabla F_i^p(\theta_r)]||^2]}_{U_4} + \frac{TM\gamma}{|\mathcal{M}^*|} \delta^2,
\end{aligned} \tag{17}$$

where $(a-b)$ follow by applying the harmonic inequalities; (c) follows from the upper bound of data heterogeneity level.

Applying Jensen's inequality, we derive the upper bound of U_4 as:

$$\gamma T \sum_{S^p\in\mathcal{S}_r}\mathbb{E}[||\frac{1}{T|\mathcal{M}_r^p|}\sum_{i\in\mathcal{M}_r^p}\sum_{t=1}^{T}[\nabla F_i^p(\theta_{i,r,t-1})-\nabla F_i^p(\theta_r)]||^2] \le \frac{\gamma}{|\mathcal{M}^*|}\sum_{i=1}^{M}\sum_{t=1}^{T}\sum_{S^p\in\mathcal{S}_r}\mathbb{E}[||\nabla F_i^p(\theta_{i,r,t-1})-\nabla F_i^p(\theta_r)||^2]$$
$$\le \frac{\gamma}{|\mathcal{M}^*|}\sum_{i=1}^{M}\sum_{t=1}^{T}\mathbb{E}[||[\nabla F_i(\theta_{i,r,t-1})-\nabla F_i(\theta_r)]||^2] \overset{(a)}{\le} \frac{T\gamma}{|\mathcal{M}^*|}\sum_{i=1}^{M}L^2\underbrace{\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[||\theta_{i,r,t-1}-\theta_r||^2]}_{U_5}, \tag{18}$$

where (a) follows from the lipschitz continuity of the gradient of the loss function.

Now we give an important lemma.

Lemma 1. *Suppose a constant learning rate γ satisfying $8\gamma^2L^2T^2\le\frac{1}{2}$, the following inequality holds:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[||\theta_{i,r,t-1}-\theta_r||^2\right]$$
$$\le 4\gamma^2T\sigma^2+32\gamma^2T^2\delta^2+32\gamma^2T^2\sum_{S^p\in\mathcal{S}_r}\mathbb{E}[||\nabla F^p(\theta_r)||^2]+2\omega^2\mathbb{E}[||\theta_r||^2]. \tag{19}$$

Proof: Applying Assumptions 1-4, we derive the upper bound of U_5 as follows:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[||\theta_{i,r,t-1}-\theta_r||^2\right]$$
$$\le \frac{2}{T}\sum_{t=1}^{T}\mathbb{E}[||\theta_{i,r,t-1}-\theta_{i,r,0}||^2]+\frac{2}{T}\sum_{t=1}^{T}\mathbb{E}[||\theta_{i,r,0}-\theta_r||^2]$$
$$= \frac{2}{T}\sum_{t=1}^{T}\mathbb{E}[||\sum_{j=0}^{t-2}-\gamma\nabla F_i(\theta_{i,r,j},\xi_{i,j})\odot m_{r,i}||^2]+\frac{2}{T}\sum_{t=1}^{T}\mathbb{E}[||\theta_r\odot m_{n,r}-\theta_r||^2]$$
$$= \frac{2\gamma^2}{T}\sum_{t=1}^{T}\mathbb{E}[||\sum_{j=0}^{t-2}(\nabla F_i(\theta_{i,r,j},\xi_{i,j})-\nabla F_i(\theta_{i,r,j})+\nabla F_i(\theta_{i,r,j}))\odot m_{r,i}||^2]+\frac{2}{T}\sum_{t=1}^{T}\omega^2\mathbb{E}[||\theta_r||^2]$$
$$\le \frac{4\gamma^2}{T}\sum_{t=1}^{T}\mathbb{E}[||\sum_{j=0}^{t-2}(\nabla F_i(\theta_{i,r,j},\xi_{i,j})-\nabla F_i(\theta_{i,r,j}))\odot m_{r,i}||^2]+\frac{4\gamma^2}{T}\sum_{t=1}^{T}\mathbb{E}[||\sum_{j=0}^{t-2}\nabla F_i(\theta_{i,r,j})\odot m_{r,i}||^2]+\frac{2}{T}\sum_{t=1}^{T}\omega^2\mathbb{E}[||\theta_r||^2]$$
$$\le \frac{4\gamma^2}{T}\sum_{t=1}^{T}(t-1)\sigma^2+\frac{4\gamma^2}{T}\sum_{t=1}^{T}\mathbb{E}[||\sum_{j=0}^{t-2}[\nabla F_i(\theta_{i,r,j})-\nabla F_i(\theta_r)+\nabla F_i(\theta_r)\odot m_{r,i}||^2]+\frac{2}{T}\sum_{t=1}^{T}\omega^2\mathbb{E}[||\theta_r||^2]$$
$$\le 2\gamma^2T\sigma^2+\frac{8\gamma^2}{T}\sum_{t=1}^{T}(t-1)\sum_{j=0}^{t-2}\mathbb{E}[||(\nabla F_i(\theta_{i,r,j})-\nabla F_i(\theta_r))\odot m_{r,i}||^2]$$
$$+\frac{8\gamma^2}{T}\sum_{t=1}^{T}(t-1)\sum_{j=0}^{t-2}\mathbb{E}[||\nabla F_i(\theta_r)\odot m_{q,n}||^2]+\frac{2}{T}\sum_{t=1}^{T}\omega^2\mathbb{E}[||\theta_r||^2]$$
$$\le 2\gamma^2T\sigma^2+\frac{8\gamma^2L^2}{T}\sum_{t=1}^{T}(t-1)\sum_{j=0}^{t-2}\mathbb{E}[||\theta_{i,r,j}-\theta_r||^2]$$
$$+8\gamma^2T^2\mathbb{E}[||(\nabla F_i(\theta_r)-\nabla F(\theta_r)+\nabla F(\theta_r))\odot m_{r,i}||^2]+\frac{2}{T}\sum_{t=1}^{T}\omega^2\mathbb{E}[||\theta_r||^2]$$
$$\le 2\gamma^2T\sigma^2+8\gamma^2L^2T^2(T-1)\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[||\theta_{i,r,t-1}-\theta_r||^2]+16\gamma^2T^2\mathbb{E}[||(\nabla F_i(\theta_r)-\nabla F(\theta_r))\odot m_{r,i}||^2]$$
$$+16\gamma^2T^2\mathbb{E}[||\nabla F(\theta_r)\odot m_{r,i}||^2]+\frac{2}{T}\sum_{t=1}^{T}\omega^2\mathbb{E}[||\theta_r||^2]$$
$$\le 2\gamma^2T\sigma^2+8\gamma^2L^2T\sum_{t=1}^{T}\mathbb{E}[||\theta_r-\theta_{i,r,t-1}||^2]+16\gamma^2T^2\delta^2+16\gamma^2T^2\mathbb{E}[||\nabla F(\theta_r)\odot m_{r,i}||^2]+2\omega^2\mathbb{E}[||\theta_r||^2]. \tag{20}$$

Because $\gamma \leq \frac{1}{4LT} \Rightarrow 8\gamma^2 L^2 T^2 \leq \frac{1}{2}$, we have:

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[||\theta_{i,r,t-1}-\theta_r||^2] &\leq 4\gamma^2 T\sigma^2 + 32\gamma^2T^2\delta^2 + 32\gamma^2T^2\mathbb{E}[||\nabla F(\theta_r)\odot m_{r,i}||^2] + 2\omega^2\mathbb{E}[||\theta_r||^2] \\ &\leq 4\gamma^2 T\sigma^2 + 32\gamma^2T^2\delta^2 + 32\gamma^2T^2\sum_{S^p\in\mathcal{S}_r}\mathbb{E}[||\nabla F^p(\theta_r)||^2] + 2\omega^2\mathbb{E}[||\theta_r||^2]. \end{aligned} \tag{21}$$

Finally, Lemma 1 holds. ■

Essentially, Lemma 1 upper-bounds the difference between local subnetwork and global model, which quantifies of the effects of local subnetwork training $\theta_{i,r,t-1}-\theta_{i,r,0}$ and subnetwork mask error $\theta_{i,r,0}-\theta_r$.

Applying Lemma 1, we get the upper bound of U_1 as follows:

$$\begin{aligned}&\mathbb{E}[\langle\nabla F(\theta_r), \theta_{r+1}-\theta_r\rangle] \\ &\leq -\sum_{S^p\in\mathcal{S}_r}\gamma T\mathbb{E}[||\nabla F^p(\theta_r)||^2] + \frac{\gamma T}{2}\sum_{S^p\in\mathcal{S}_r}\mathbb{E}[||\nabla F^p(\theta_r)||^2] \\ &\quad + \frac{T\gamma}{|\mathcal{M}^*|}\sum_{i=1}^{M}L^2\left(4\gamma^2T\sigma^2 + 32\gamma^2T^2\delta^2 + 32\gamma^2T^2\sum_{S^p\in\mathcal{S}_r}\mathbb{E}[||\nabla F^p(\theta_r)||^2] + 2\omega^2\mathbb{E}[||\theta_r||^2]\right) + \frac{TM\gamma}{|\mathcal{M}^*|}\delta^2 \\ &\leq -\sum_{S^p\in\mathcal{S}_r}\gamma T\mathbb{E}[||\nabla F^p(\theta_r)||^2] + \frac{\gamma T}{2}\sum_{S^p\in\mathcal{S}_r}\mathbb{E}[||\nabla F^p(\theta_r)||^2] + 4\gamma^3T^2L^2\sigma^2\frac{M}{|\mathcal{M}^*|} \\ &\quad + 32\gamma^3T^3L^2\delta^2\frac{M}{|\mathcal{M}^*|} + 32\gamma^3T^3L^2\frac{M}{|\mathcal{M}^*|}\sum_{S^p\in\mathcal{S}_r}\mathbb{E}[||\nabla F^p(\theta_r)||^2] + 2\gamma TL^2\omega^2\frac{M}{|\mathcal{M}^*|}\mathbb{E}[||\theta_r||^2] + \gamma T\delta^2\frac{M}{|\mathcal{M}^*|}. \end{aligned} \tag{22}$$

Applying Cauchy–Schwarz inequality and Assumption 4, we further derive the upper bound of U_2 in (15):

$$\begin{aligned}&\frac{L}{2}\mathbb{E}[||\theta_{r+1}-\theta_r||^2] = \frac{L}{2}\sum_{S^p\in\mathcal{S}_r}\mathbb{E}[||\theta^p_{r+1}-\theta^p_r||^2] + \frac{L}{2}\sum_{S^p\in\mathcal{S}/\mathcal{S}_r}\mathbb{E}[||\theta^p_{r+1}-\theta^p_r||^2] \\ &= \frac{L}{2}\sum_{S^p\in\mathcal{S}_r}\mathbb{E}[||\theta^p_{r+1}-\theta^p_r||^2] = \frac{L}{2}\sum_{S^p\in\mathcal{S}_r}\mathbb{E}\left|\left|\frac{1}{|\mathcal{M}^p_r|}\sum_{i\in\mathcal{M}^p_r}(\theta^p_{i,r,0}-\theta^p_{i,r,T})\right|\right|^2 \\ &= \frac{L}{2}\sum_{S^p\in\mathcal{S}_r}\mathbb{E}\left|\left|\frac{1}{|\mathcal{M}^p_r|}\sum_{i\in\mathcal{M}^p_r}\left(\theta^p_{i,r,0}-\left(\theta^p_{i,r,0}-\sum_{t=1}^{T}\gamma\nabla F_i(\theta^p_{i,r,t-1},\xi_{i,t-1})\odot m_{i,r}\right)\right)\right|\right|^2 \\ &= \frac{L}{2}\sum_{S^p\in\mathcal{S}_r}\mathbb{E}\left|\left|\frac{1}{|\mathcal{M}^p_r|}\sum_{i\in\mathcal{M}^p_r}\sum_{t=1}^{T}\gamma\nabla F^p_i(\theta_{i,r,t-1},\xi_{i,t-1})\right|\right|^2 \\ &\leq \frac{3L}{2}\sum_{S^p\in\mathcal{S}_r}\mathbb{E}\left|\left|\frac{1}{|\mathcal{M}^p_r|}\sum_{i\in\mathcal{M}^p_r}\sum_{t=1}^{T}\gamma\left(\nabla F^p_i(\theta_{i,r,t-1},\xi_{i,t-1})-\nabla F^p_i(\theta_{i,r,t-1})\right)\right|\right|^2 \\ &\quad + \frac{3L}{2}\sum_{S^p\in\mathcal{S}_r}\mathbb{E}\left|\left|\frac{1}{|\mathcal{M}^p_r|}\sum_{i\in\mathcal{M}^p_r}\sum_{t=1}^{T}\gamma(\nabla F^p_i(\theta_{i,r,t-1})-\nabla F^p(\theta_r))\right|\right|^2 + \frac{3L}{2}\sum_{S^p\in\mathcal{S}_r}\mathbb{E}\left|\left|\frac{1}{|\mathcal{M}^p_r|}\sum_{i\in\mathcal{M}^p_r}\sum_{t=1}^{T}\gamma\nabla F^p(\theta_r)\right|\right|^2 \\ &\overset{(a)}{\leq} \frac{3}{2}LT\gamma^2\sigma^2\frac{M}{|\mathcal{M}^*|} + \frac{3\gamma^2TL^3M}{|\mathcal{M}^p_r|}(4\gamma^2T\sigma^2 + 32\gamma^2T^2\delta^2 + 32\gamma^2T^2\sum_{S^p\in\mathcal{S}_r}\mathbb{E}[||\nabla F^p(\theta_r)||^2] + 2\omega^2\mathbb{E}[||\theta_r||^2]) \\ &\quad + 3L\gamma^2T^2\delta^2\frac{M}{|\mathcal{M}^*|} + \frac{3}{2}L\gamma^2T^2\sum_{S^p\in\mathcal{S}_r}\mathbb{E}[||\nabla F^p(\theta_r)||^2] \\ &= \frac{3}{2}LT\gamma^2\sigma^2\frac{M}{|\mathcal{M}^*|} + 12\gamma^4T^2L^3\sigma^2\frac{M}{|\mathcal{M}^*|} + 96\gamma^4T^4L^3\delta^2\frac{M}{|\mathcal{M}^*|} + 96\gamma^4T^4L^3\frac{M}{|\mathcal{M}^*|}\sum_{S^p\in\mathcal{S}_r}\mathbb{E}[||\nabla F^p(\theta_r)||^2] \\ &\quad + \frac{6\gamma^2TL^3\omega^2M}{|\mathcal{M}^*|}\mathbb{E}[||\theta_r||^2] + 3L\frac{M}{|\mathcal{M}^*|}\gamma^2T^2\delta^2 + \frac{3}{2}L\gamma^2T^2\sum_{S^p\in\mathcal{S}_r}\mathbb{E}[||\nabla F^p(\theta_r)||^2]. \end{aligned} \tag{23}$$

Here, (a) follows from Lemma 1. We substitute U_1, U_2 in (15) with (22) and (23), respectively. Then, iterating from $r=1$ to $r=R$ yields:

$$
\begin{aligned}
&\mathbb{E}[F(\theta_{R+1})] - \mathbb{E}[F(\theta_1)] = \sum_{r=1}^{R} \mathbb{E}[F(\theta_{R+1})] - \sum_{r=1}^{R} \mathbb{E}[F(\theta_r)] \\
\leq & \sum_{r=1}^{R} \mathbb{E}\left[\langle \nabla F(\theta_r), \theta_{R+1} - \theta_r \rangle\right] + \sum_{r=1}^{R} \frac{L}{2} \mathbb{E}[||\theta_{R+1} - \theta_r||^2] \\
\leq & -T\gamma \sum_{r=1}^{R} \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\nabla F^p(\theta_r)||^2] + \frac{T\gamma}{2} \sum_{r=1}^{R} \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\nabla F^p(\theta_r)||^2] + 32\gamma^3 T^3 L^2 \frac{M}{|\mathcal{M}^*|} \sum_{r=1}^{R} \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\nabla F^p(\theta_r)||^2] \\
& + 4T\gamma L^2 \omega^2 \frac{M}{|\mathcal{M}^*|} \sum_{r=1}^{R} \mathbb{E}[||\theta_r||^2] + 4\gamma^3 T^2 L^2 R \sigma^2 \frac{M}{|\mathcal{M}^*|} + 32\gamma^3 T^3 L^2 R \delta^2 \frac{M}{|\mathcal{M}^*|} + T\gamma R \delta^2 \frac{M}{|\mathcal{M}^*|} + \frac{3}{2}\gamma^2 LTR\sigma^2 \\
& + 12L^3\gamma^4 T^3 R\sigma^2 \frac{M}{|\mathcal{M}^*|} + 96\gamma^4 T^4 L^3 R\delta^2 \frac{M}{|\mathcal{M}^*|} + 96\gamma^4 T^4 L^3 \frac{M}{|\mathcal{M}^*|} \sum_{r=1}^{R} \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\nabla F^p(\theta_r)||^2] \\
& + 6L^3\gamma^2 T^2 \frac{M}{|\mathcal{M}^*|} \omega^2 \sum_{r=1}^{R} \mathbb{E}[||\theta_r||^2] + \frac{3}{2} L\gamma^2 T^2 \sum_{r=1}^{R} \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\nabla F^p(\theta_r)||^2] + 3L\gamma^2 T^2 R\delta^2 \frac{M}{|\mathcal{M}^*|}. \qquad (24)
\end{aligned}
$$

Taking $32\gamma^2 T^2 L^2 \frac{M}{|\mathcal{M}^*|} \leq \frac{1}{8} \implies \gamma \leq \frac{\sqrt{|\mathcal{M}^*|}}{16TL\sqrt{M}}$, $96\gamma^3 T^3 L^3 \frac{M}{|\mathcal{M}^*|} \leq \frac{1}{8} \implies \gamma \leq \left(\frac{|\mathcal{M}^*|}{768 L^3 T^3 M}\right)^{\frac{1}{3}}$ and $\frac{3}{2} L\gamma T \leq \frac{1}{8} \implies \gamma \leq \frac{1}{12TL}$ yields:

$$
\begin{aligned}
& \mathbb{E}[F(\theta_{R+1})] - \mathbb{E}[F(\theta_1)] \\
\leq & -\frac{T\gamma}{8} \sum_{r=1}^{R} \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\nabla F^p(\theta_r)||^2] + \left(4L^2 T\gamma\omega^2 \frac{M}{|\mathcal{M}^*|} + 6L^3\gamma^2 T^2 \omega^2 \frac{M}{|\mathcal{M}^*|}\right) \sum_{r=1}^{R} \mathbb{E}[||\theta_r||^2] \\
& + T\gamma R\delta^2 \frac{M}{|\mathcal{M}^*|} \left(32\gamma^2 T^2 L^2 + 1 + 96L^3\gamma^3 T^3 + 3L\gamma T\right) + \gamma^2 TLR\sigma^2 \frac{M}{|\mathcal{M}^*|} (4\gamma TL + \frac{3}{2} + 12L^2\gamma^2 T^2). \qquad (25)
\end{aligned}
$$

Therefore, we have:

$$
\begin{aligned}
& \frac{T\gamma}{8} \sum_{r=1}^{R} \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\nabla F^p(\theta_r)||^2] \\
\leq & \mathbb{E}[F(\theta_1)] - \mathbb{E}[F(\theta_{R+1})] + \left(4L^2 T\gamma\omega^2 \frac{M}{|\mathcal{M}^*|} + 6L^3\gamma^2 T^2 \omega^2 \frac{M}{|\mathcal{M}^*|}\right) \sum_{r=1}^{R} \mathbb{E}[||\theta_r||^2] \\
& + T\gamma R\delta^2 \frac{M}{|\mathcal{M}^*|} \left(32\gamma^2 T^2 L^2 + 1 + 96L^3\gamma^3 T^3 + 3L\gamma T\right) + \gamma^2 TLR\sigma^2 \frac{M}{|\mathcal{M}^*|} (4\gamma TL + \frac{3}{2} + 12L^2\gamma^2 T^2). \qquad (26)
\end{aligned}
$$

Dividing both sides by $\frac{RT\gamma}{8}$,

$$
\begin{aligned}
& \frac{1}{R} \sum_{r=1}^{R} \sum_{S^p \in \mathcal{S}_r} \mathbb{E}[||\nabla F^p(\theta_r)||^2] \\
\leq & \frac{8}{RT\gamma} \left(\mathbb{E}[F(\theta_1)] - \mathbb{E}[F(\theta_{R+1})]\right) + \frac{32\omega^2 L^2 M + 48L^3\gamma TM}{|\mathcal{M}^*|R} \sum_{r=1}^{R} \mathbb{E}\left[||\theta_r||^2\right] \\
& + \frac{8\delta^2 M}{|\mathcal{M}^*|} \left(32\gamma^2 T^2 L^2 + 1 + 96L^3\gamma^3 T^3 + 3L\gamma T\right) + \frac{8\gamma L\sigma^2 M}{|\mathcal{M}^*|} (4\gamma TL + \frac{3}{2} + 12L^2\gamma^2 T^2). \qquad (27)
\end{aligned}
$$

Until now we complete the proof of Theorem 1.

REFERENCES

[1] 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. of AISTATS 2017*, 20–22 Apr. 2017, pp. 1273–1282.

[2] L. Li, D. Shi, R. Hou, H. Li, M. Pan, and Z. Han, "To talk or to work: Flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[3] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[4] R. Chen, Q. Wan, X. Zhang, X. Qin, Y. Hou, D. Wang, X. Fu, and M. Pan, "Eefl: High-speed wireless communications inspired energy efficient federated learning over mobile devices," in *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*, ser. MobiSys '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 544–556. [Online]. Available: https://doi.org/10.1145/3581791.3596865

[5] A. Hard, K. Rao, R. Mathews, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *ArXiv*, vol. abs/1811.03604, 2018.

[6] T. Yu, T. Li, Y. Sun, S. Nanda, V. Smith, V. Sekar, and S. Seshan, "Learning context-aware policies from multiple smart homes via federated multi-task learning," in *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 2020, pp. 104–115.

[7] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," *International Journal of Medical Informatics*, vol. 112, pp. 59–67, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S138650561830008X

[8] J. Geng, B. Li, X. Qin, Y. Li, L. Li, Y. Hou, and M. Pan, "Fedex: Expediting federated learning over heterogeneous mobile devices by overlapping and participant selection," *IEEE Transactions on Mobile Computing*, 2025.

[9] P. J. Bickel, E. A. Hammel, and J. W. O'Connell, "Sex bias in graduate admissions: Data from berkeley," *Science*, vol. 187, no. 4175, pp. 398–404, 1975. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.187.4175.398

[10] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," 2021. [Online]. Available: https://openreview.net/forum?id=PYAFKBc8GL4

[11] D. Ye, R. Yu, M. Pan, and Z. Han, "Federated learning in vehicular edge computing: A selective model aggregation approach," *IEEE Access*, vol. 8, pp. 23 920–23 935, 2020.

[12] C. Liu, X. Qu, J. Wang, and J. Xiao, "Fedet: A communication-efficient federated class-incremental learning framework based on enhanced transformer," in *International Joint Conference on Artificial Intelligence*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259261962

[13] L. Li, C. Huang, D. Shi, H. Wang, X. Zhou, M. Shu, and M. Pan, "Energy and spectrum efficient federated learning via high-precision over-the-air computation," *IEEE Transactions on Wireless Communications*, vol. 23, no. 2, pp. 1228–1242, 2023.

[14] Y. Chen, Z. Chen, P. Wu, and H. Yu, "Fedobd: Opportunistic block dropout for efficiently training large-scale neural networks through federated learning," in *International Joint Conference on Artificial Intelligence*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251468165

[15] D. Wen, K.-J. Jeon, and K. Huang, "Federated dropout—a simple approach for enabling federated learning on resource constrained devices," *IEEE Wireless Communications Letters*, vol. 11, pp. 923–927, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:238226744

[16] E. Diao, J. Ding, and V. Tarokh, "Heterofl: Computation and communication efficient federated learning for heterogeneous clients," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=TNkPBBYFkXg

[17] M. Kim, S. Yu, S. Kim, and S.-M. Moon, "Depthfl : Depthwise federated learning for heterogeneous clients," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=pf8RIZTMU58

[18] S. Banabilah, M. Aloqaily, E. Alsayed, N. Malik, and Y. Jararweh, "Federated learning review: Fundamentals, enabling technologies, and future applications," *Information Processing, Management*, vol. 59, no. 6, p. 103061, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457322001649

[19] A. Achille, M. Rovere, and S. Soatto, "Critical learning periods in deep networks," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:108298098

[20] G. Yan, H. Wang, and J. Li, "Seizing critical learning periods in federated learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, pp. 8788–8796, Jun. 2022. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/20859

[21] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[22] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient federated learning via guided participant selection," in *15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21)*. USENIX Association, Jul. 2021, pp. 19–35. [Online]. Available: https://www.usenix.org/conference/osdi21/presentation/lai

[23] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, "Model pruning enables efficient federated learning on edge devices," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 12, pp. 10 374–10 386, 2022.

[24] A. Li, J. Sun, X. Zeng, M. Zhang, H. Li, and Y. Chen, "Fedmask: Joint computation and communication-efficient personalized federated learning via heterogeneous masking," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 42–55.

[25] A. Li, J. Sun, P. Li, Y. Pu, H. Li, and Y. Chen, "Hermes: an efficient federated learning framework for heterogeneous mobile clients," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 420–437.

[26] L. Yi, X. Shi, N. Wang, J. Zhang, G. Wang, and X. Liu, "Fedpe: Adaptive model pruning-expanding for federated learning on mobile devices," *IEEE Transactions on Mobile Computing*, 2024.

[27] S. Horvath, S. Laskaridis, M. Almeida, I. Leontiadis, S. Venieris, and N. Lane, "Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 876–12 889, 2021.

[28] S. Alam, L. Liu, M. Yan, and M. Zhang, "Fedrolex: model-heterogeneous federated learning with rolling sub-model extraction," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2024.

[29] F. Wu, X. Wang, Y. Wang, T. Liu, L. Su, and J. Gao, "Fiarse: Model-heterogeneous federated learning via importance-aware submodel extraction," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[30] H. Wang, Y. Jia, M. Zhang, Q. Hu, H. Ren, P. Sun, Y. Wen, and T. Zhang, "Feddse: Distribution-aware sub-model extraction for federated learning over resource-constrained devices," in *Proceedings of the ACM Web Conference 2024*, New York, NY, USA, 2024.

[31] S. Ruder, "An overview of gradient descent optimization algorithms," *ArXiv*, vol. abs/1609.04747, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:17485266

[32] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=HJxNAnVtDS

[33] S. Amari and H. Nagaoka, "Methods of information geometry," in *Tanslations of Mathematical Monographs, 191*, 2000. [Online]. Available: https://api.semanticscholar.org/CorpusID:116976027

[34] J. Martens, "New insights and perspectives on the natural gradient method," *J. Mach. Learn. Res.*, vol. 21, pp. 146:1–146:76, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:10284405

[35] Y. Wang, X. Zhang, M. Li, T. Lan, H. Chen, H. Xiong, X. Cheng, and D. Yu, "Theoretical convergence guaranteed resource-adaptive federated learning with mixed heterogeneity," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 2444–2455.

[36] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:206594692

[38] N. Gupta, S. Gupta, R. Pathak, V. Jain, P. Rashidi, and J. Suri, "Human activity recognition in artificial intelligence framework: a narrative review," *Artificial Intelligence Review*, vol. 55, 08 2022.
[39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *North American Chapter of the Association for Computational Linguistics*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:52967399
[40] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," 2021.

Liang Li is an assistant researcher at the Frontier Research Center, Pengcheng Laboratory, Shenzhen, China. She received her Ph.D. degree in Information and Communication Engineering from Xidian University, China, in 2021. She was a visiting Ph.D. student with the Department of Electrical and Computer Engineering, University of Houston, TX, USA, from 2018 to 2020. She was a postdoctoral faculty member with the School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing, China, from 2021 to 2023. Her research interests include 6G networks, mobile LLM, distributed machine learning.

Jiaxiang Geng received his B.S. degree in Information Engineering from Beijing University of Posts and Telecommunications (BUPT) in 2022 and his M.S. degree in Information and Communication Engineering from the School of Information and Communication Engineering, BUPT, in 2025. He is currently pursuing his Ph.D. degree in Electrical and Electronic Engineering at the University of Hong Kong. His research interests include federated learning, mobile edge computing, and deep learning.

Huai-an Su received the B.S. degree in Electrical Engineering from National Tsing Hua University, Taiwan, in 2020, and the Ph.D. degree in Electrical and Computer Engineering from the University of Houston, Houston, TX, USA, in 2025. He is currently engaged in research on federated learning, model compression, and edge AI systems.

Xiaoqi Qin received her B.S., M.S., and Ph.D. degrees from Electrical and Computer Engineering with Virginia Tech. She is currently an Associate Professor of School of Information and Communication Engineering with Beijing University of Posts and Telecommunication (BUPT). Her research mainly focuses on task-oriented machine-type communications and networked intelligence. She has published more than 80 journal and conference papers, one book, and holds 21 patents on these areas. She was a Distinguished Young Investigator of China Frontiers of Engineering. She has received the Best Paper Awards at IEEE GLOBECOM'23 and WCSP'23. She was a recipient of first Prize of Science and Tech. Progress Award by Chongqing Municipal People's Government, and first Prize of Tech. Invention Award by China Institute of Communications.

Yanzhao Hou received his Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2014. He is currently an Associate Professor with the National Engineering Research Center for Mobile Network Technologies, BUPT. His current research interests include federated learning, software defined radio, terahertz communications and trial systems. He received the Best Demo Award in IEEE APCC2018.

Hao Wang is an Assistant Professor in the Department Electrical and Computer Engineering at Stevens Institute of Technology, Hoboken, NJ, USA. He received both his B.E. degree in Information Security and M.E. degree in Software Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2012 and 2015 respectively, and the Ph.D. degree in the Department of Electrical and Computer Engineering at the University of Toronto, Canada in 2020. His research interests include distributed ML systems, AI security and forensics, privacy-preserving data analytics, serverless computing, and high-performance computing. He is a recipient of the NSF CRII Award.

Xin Fu received the Ph.D. in Computer Engineering from University of Florida in 2009. She was an NSF Computing Innovation Fellow with the CS Department, University of Illinois at Urbana-Champaign from 2009 to 2010. From 2010 to 2014, she was an Assistant Professor at the EECS Department, University of Kansas. Currently, she is a Professor at the ECE Department, University of Houston. Her research interests include energy-efficient computing, machine learning, edge/mobile computing, high-performance computing. Dr. Fu is a recipient of 2014 NSF Faculty Early CAREER Award, 2012 Kansas NSF EPSCoR First Award, and 2009 NSF Computing Innovation Fellow.

Miao Pan is a Full Professor in the Department of Electrical and Computer Engineering at University of Houston. He was a recipient of NSF CAREER Award in 2014. Dr. Pan received his Ph.D. degree in Electrical and Computer Engineering from University of Florida in August 2012. Dr. Pan's research interests include wireless for AI, mobile AI systems, LLM security & privacy, deep learning privacy, new biometric based authentication, quantum computing privacy, wireless networks, and underwater IoT networks. He has published 3 books and book chapters, 5 patents, more than 150 papers in prestigious journals/magazines and more than 160 papers in top conferences. He has also been serving as a TPC Co-Chair for Mobiquitous 2019, ACM WUWNet 2019, Local Chair for MobiHoc 2025, etc. Dr. Pan is an Associate Editor for ACM Computing Surveys, IEEE Journal of Oceanic Engineering, IEEE Open Journal of Vehicular Technology and IEEE Internet of Things (IoT) Journal (Area 5: Artificial Intelligence for IoT), and used to be an Associate Editor for IEEE Internet of Things (IoT) Journal (Area 4: Services, Applications, and Other Topics for IoT) from 2015 to 2018. Dr. Pan is a member of AAAI, a senior member of ACM, and a senior member of IEEE and IEEE Communications Society.