From Non-IID to IID: Mobility-Aware Hierarchical Federated Learning With Client-Edge Association Control

Haibo Liu, *Graduate Student Member, IEEE*, Zhenzhe Zheng, *Member, IEEE*, Fan Wu, *Member, IEEE*, and Guihai Chen, *Fellow, IEEE*

***Abstract*—Deploying federated learning (FL) in wireless network with hierarchical client-edge-cloud architecture enables large-scale distribution collaboration without long-distance communication latency. However, the ongoing edge dynamics with uncertain client mobility and imbalanced data distributions, poses great challenge for collaboration efficiency of FL. In this work, we first model the client mobility with a Markov chain, and formulate the minimization of performance degradation as a client-edge association control problem. With the analysis of client mobility patterns, we propose ALPHA, a new client-edge association control framework for mobility-aware FL, to reshape the edge-level data distributions close to i.i.d in both offline and online mobility scenarios. In the offline scenario with deterministic client mobility trajectories, we leverage alternating optimization theory to transform the client-edge association control problem into a weighted bipartite b-matching problem, and derive an efficient solution with linear relaxation and dependent rounding techniques. As for the online scenario, where each client arrives at different edge access points (APs) in an online manner, we design a fast and simple online subgradient projection algorithm with a bounded regret to make an online decision on client-edge association. Extensive experiment results on three public datasets and a real-world mobility trajectory dataset show that ALPHA has a superior learning performance with 1.40× – 2.89× convergence speedup compared to state-of-the-art solutions.**

***Index Terms*—Hierarchical federated learning, mobility, non-i.i.d data, client-edge association.**

I. Introduction

NOWDAYS, FL is becoming a popular distributed machine learning paradigm for enabling massive devices to jointly train a model without exposing raw data [1], [2], [3], [4], [5]. However, when deploying FL in wireless networks [6], [7], the communication between clients and the cloud server is unreliable for a long distance transmission, which degrades learning efficiency as well as final model accuracy [8]. Hierarchical FL (HFL) has been proposed as an effective solution to resolve this paradigm by enabling multiple clients to perform model aggregation at nearby edge APs [9], [10], [11]. Although HFL has advantage on communication efficiency, how to make effective collaboration among clients, edge APs and the cloud within a hierarchical architecture is a challenging problem, especially within uncertain edge environments, such as client mobility.

Mobility is a key feature of many real-world mobile computing applications such as path planning for autonomous vehicles [12] and target search with impossible aerial vehicles [13]. Although few works have focused on client mobility [14], [15], [16], these approaches attempt to tame mobility through device sampling with a restricted number of clients or by introducing extra operations to redesign the access mechanism, leading to inefficient collaboration in HFL. The significance of client mobility in HFL extends beyond its impact on communication efficiency in dynamic network topologies to critically influencing the balance of data distributions and model convergence performance. To enhance the collaboration efficiency of HFL, it is crucial to understand the uncertainty of client mobility and its impact on collaboration performance. As shown in Fig. 1, mobile clients in HFL may roam across different edge APs, and the uncertainty of client mobility where the client-edge association will take place and for how long, leading to dynamic clients sets associated with edge APs. In this way, the data distributions within the edge APs can evolve with time, which may intensify the performance degradation due to dynamic non-i.i.d data distributions in hierarchical aggregation manner. Despite that the uncertainty of client mobility may degrade the model training performance without a careful coordination, we identify that it also creates new model aggregation opportunities at different edge APs, which is capable of reshaping the data distributions of edge APs to be i.i.d. To validate the feasibility of this motivation, we conduct pilot experiments with two distinct federated collaboration strategies: edge-dynamic model collaboration and edge-controlled model collaboration. The edge-dynamic model collaboration strategy disregards client mobility patterns, performing model aggregation with completely random client selection at edge APs. In contrast, the edge-controlled model collaboration strategy systematically selects mobile clients for model aggregation based on their mobility trajectories and data distribution. As shown in Fig. 2, it is verified that the data

Received 16 October 2024; revised 26 April 2025; accepted 17 June 2025. Date of publication 18 July 2025; date of current version 3 October 2025. This work was supported in part by National Key R&D Program of China under Grant 2023YFB4502400 and in part by China NSF under Grant 62322206, Grant 62132018, Grant 62025204, Grant U2268204, Grant 62272307, and Grant 62372296. Recommended for acceptance by W. Liao. *(Corresponding author: Zhenzhe Zheng.)*

The authors are with the Shanghai Key Laboratory of Scalable Computing and Systems, School of Computer Science, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: liuhaibo@sjtu.edu.cn; zhengzhenzhe@sjtu.edu.cn; fwu@cs.sjtu.edu.cn; gchen@cs.sjtu.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TMC.2025.3585538, provided by the authors.

Digital Object Identifier 10.1109/TMC.2025.3585538

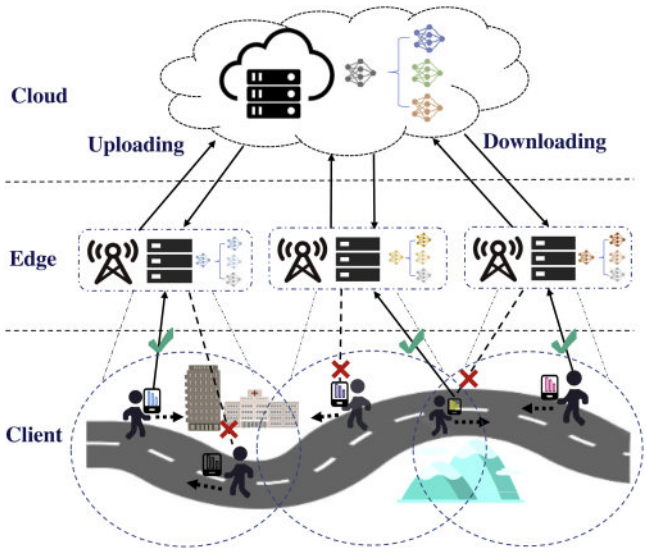

Fig. 1. Client-edge association control in mobility-aware HFL.

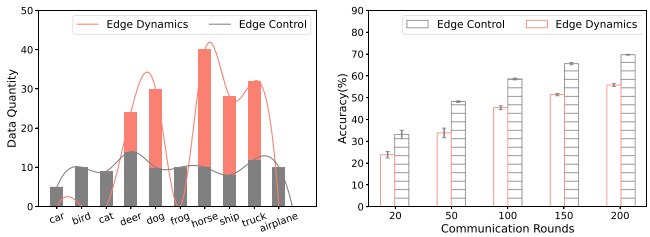

Fig. 2. The performance comparison between edge dynamics and client-edge control validated on the dataset CIFAR10: (a) the class distribution of edge APs; (b) average accuracy after a certain communication rounds.

distributions of edge APs can be reshaped close to i.i.d through efficient client-edge association control, thereby enhancing the learning performance.

How to alleviate the impact of uncertain client mobility to improve collaboration efficiency in HFL is a quiet challenging problem. In particular, to realize the mobility-aware HFL in wireless networks mainly faces the following three challenges: (1) Unknown mobility patterns: mobile clients with different data distributions usually download the latest global model from the nearest edge AP, and then roam across different edge APs with their own mobility behaviors. Thus, it is hard to know which edge AP the clients will encounter in the next time slot, and whether to upload the model parameters for model aggregation at the current edge APs. This uncertain mobility trajectory of clients will significantly impact the data distributions at the edge APs. (2) Dynamic client-edge association: a large number of previous works in HFL only focus on the static network topology without considering the edge dynamics with uncertain client mobility and imbalanced data distributions [17], [18]. Although mobile clients may have new opportunities to exchange model information with multiple edge APs in HFL, such new features may lead to a time-varying network topology, which introduces high computational complexity to identify the optimal client-edge association among large candidates. Moreover, If mobile clients upload their local models at different edge APs with frequent client-edge association, it would result in redundant model parameters across edge APs, and consequently impair the collaborative efficiency of FL [19]. (3) High client heterogeneity: the heterogeneous clients with various computation capacity could differ in the time slot of completing and uploading the updated model parameters, which may not be consistent with the time slot associated with the optimal edge AP due to the client mobility. Furthermore, the mobility patterns could lead to a unbalance clustering phenomenon, where mobile clients are concentrated into one edge AP with dense association and few clients at other edge APs with sparse connectivity [20]. Without a careful control for client-edge association, such phenomenon gives rise to inefficient collaboration, intensifying the non-i.i.d data distribution problem.

In this paper, by jointly considering the above challenges, we propose a new mobility-aware wireless federated learning framework called ALPHA, which conducts effective and efficient client-edge association control in both the offline and online mobility scenarios. First, we use markov chain to model the mobility of clients, and analyze the impact of non-i.i.d data distributions of clients, edge APs and the cloud, on the model convergence. We next formulate the data distribution divergence minimization as the client-edge association control problem. Considering the various client mobility scenarios, we analyze the periodic and random mobility patterns, and propose two specific client-edge association control strategies, respectively. In the offline scenario, we use alternating optimization theory to transform the formulated non-linear binary programming into a bipartite b-matching problem [21], [22], and an efficient solution can be derived with the LP relaxation and dependent rounding technique. As for the online one, we leverage the subgradient projection to design a fast and simple online LP algorithm with a bounded regret to make an online decision on which edge APs to upload the local model parameters.

The main contributions can be summarized as follows:

- In this work, we consider mobility-aware wireless FL with client-edge association control. We investigate the performance degradation of HFL due to the hierarchical divergence of data distributions with uncertain client mobility, and control client-edge association to reshape data distributions at edge APs to minimize the data distribution divergence.
- For mobility-aware FL, we model client's mobility with Markov chain for periodic and random mobility patterns. Moreover, we quantify the performance degradation of model training in HFL, and formulate the minimization of data distribution divergence as a client-edge association control problem.
- In offline mobility scenario, we leverage alternate optimization theory to transform the non-linear binary programming into a bipartite b-matching problem, and address it with linear relaxation and dependent rounding technique. As for the online scenario without knowing the future mobility trajectories, a fast and simple online algorithm

TABLE I
SUMMARY OF MAIN NOTATIONS IN THIS PAPER

Variable	Description
x, y	the feature and label of data sample.
m, n	the number of edge APs and clients.
$\mathcal{M}, \mathcal{N}$	the set of edge APs and clients.
$c_{i,j}, \mathcal{C}$	the bandwidth resource of client i with edge AP j.
$o_i, \mathcal{O}$	the i-th location, the location space.
$b_j, \mathcal{B}$	the limited bandwidth of edge AP j.
$d_i, \bar{d}_j, d$	the data distribution of client i, edge AP j and the cloud.
$H_{i,j}$	the decision variable between the client i and the edge AP j.
ω, f_k	the model parameter and the probability function.
τ	the deadline for uploading local models to edge APs.
L_i, L	the loss function of client i, the global loss function.
a_i	the amount of data in the local dataset of client i.
I, E	the frequency of downward and upward model aggregation.
S_j	the set of clients aggregated with edge AP j.
$\mathcal{S}_i$	the set of edge APs which the client i moves across.
P, p_x^i	transition matrix and the probability of client i at location x.
$\omega_{i,j}$	the weight between the client i and the edge AP j.
α, β	the weight of downward divergence and upward one.
κ	the degree of non-i.i.d data distribution.

based on subgradient projection is proposed to make an irrevocable decision on client-edge association.

- Extensive evaluations on three public image datasets and a real-world mobility trajectory dataset have been conducted, and the results demonstrate that ALPHA has a superior learning performance with 1.40× – 2.89× convergence speedup compared to the state-of-the-art FL approaches.

The rest of the paper is organized as follows. In Section II, we formally present preliminaries with FL setting, mobility model and problem formulation. The analysis of two classical mobility patterns will be given in Section III. The Section IV provides the specific design of ALPHA. The experimental results and related works are shown in Sections V and VI. The conclusions are made in Section VII.

II. PRELIMINARIES

In this section, we describe FL setting, client mobility model and problem formulation. The notations used in this work are summarized in Table I.

A. Federated Learning

In wireless networks, we consider a set of mobile clients indexed by $\mathcal{N} = \{1, 2, \ldots, n\}$, a set of edge APs indexed by $\mathcal{M} = \{1, 2, \ldots, m\}$ and one cloud server. Each mobile client $i \in \mathcal{N}$ consumes bandwidth resource $c_{i,j} \in \mathcal{C}$ to communicate with the nearest edge AP $j \in \mathcal{M}$, which has a limited bandwidth resource $b_j \in \mathcal{B}$. For the learning task, we focus on a C-categories classification problem with data sample $\{x, y\}$ over a feature space $\mathcal{X}$ and a label space $\mathcal{Y}$. The client $i \in \mathcal{N}$ holds a local dataset $\mathcal{D}_i = \{x_k, y_k\}_{k=1}^{a_i}$ following the distribution d_i, and a_i is the amount of data in local dataset. The data distribution of the edge AP $j \in \mathcal{M}$ depends on the data distribution of mobile clients and its amount of data for model training. Therefore, we denote the virtual data distribution for edge APs as:

$$\bar{d}_j = \frac{\sum_{i=1}^{n} H_{i,j} d_i a_i}{\sum_{i=1}^{n} H_{i,j} a_i}, \quad \forall j \in \mathcal{M}, \tag{1}$$

where $H_{i,j}$ is the decision variable with $H_{i,j} = 1$ indicating that mobile client i uploads model parameters to the edge AP j, otherwise $H_{i,j} = 0$. Similarly, the variable $\hat{d} = \frac{\sum_{i=1}^{n} d_i}{n}$ represents the virtual data distribution for the cloud server.

Next, we define the loss function of mobile client $i \in \mathcal{N}$ with cross-entropy loss as:

$$L_i(\omega) = \sum_{k=1}^{c} d_i(y = k) \mathbb{E}_{x|y=k}[log f_k(x, \omega)], \tag{2}$$

where $d_i(y = k)$ denotes the proportion of the data with class k in the local dataset of client i, and the function f_k denotes the probability of the class k for the input feature x. The variable ω are the model parameters, and the objective of HFL is:

$$\min_{\omega} L(\omega) = \frac{1}{m} \sum_{j=1}^{m} \frac{a_i}{\sum_{i \in S_j} a_i} \sum_{i \in S_j} l_i(\omega), \tag{3}$$

where S_j is the set of mobile clients uploading model parameters to the edge AP j. For the training process in HFL, it involves three steps in a bottom-up manner, i.e., the local model training on mobile clients, downward model aggregation on the edge APs and upward model aggregation on the cloud server. Specifically, in the l-th local iteration, each mobile client $i \in \mathcal{N}$ updates its model by using SGD:

$$\omega_i^{l+1} = \omega_i^l - \eta \sum_{k=1}^{c} d_i(y = k) \nabla_{\omega} \mathbb{E}_{x|y=k}[\log f_k(x, \omega_i^l)]. \tag{4}$$

After every I local iterations, the downward model aggregation will be performed at each edge AP $j \in \mathcal{M}$. Specifically, at local iterations $l \in \{I, 2I, 3I, \cdots\}$, we update the edge model w_j^l by aggregating the local models from the associated client set S_j, i.e.,

$$\omega_j^l = \frac{1}{|S_j|} \sum_{i \in S_j} \omega_i^l, \quad \forall i \in \mathcal{N}, j \in \mathcal{M}. \tag{5}$$

Furthermore, after E rounds of downward model aggregation, models from m edge APs are averaged to obtain the global cloud model w^l and complete the upward model aggregation, i.e.,

$$\omega^l = \frac{1}{m} \sum_{j=1}^{m} \omega_j^l, \quad \forall j \in \mathcal{M}. \tag{6}$$

B. Client Mobility Model

In wireless networks, mobile clients could roam across edge APs at a certain mobility pattern. Within a time slot, mobile client could stay at the same location or move to another edge APs with a certain probability. In order to model the client's mobility, we adopt a stochastic process $\{X^t, t \geq 0\}$ for each mobile client, which is a markov chain with finite location space $\mathcal{O}$ and transition matrix P. Thus, we can obtain the following expression:

$$P(X^{t+1} = o_i | X^t = o_j) = P(o_i, o_j), \tag{7}$$

which is the conditional probability of moving from location o_i at time slot t to location o_j at time slot $t + 1$.

Mobile clients are heterogeneous in terms of their own mobility patterns, and the transition probability of each client could be quite different. We assume that mobile clients are randomly distributed over all edge APs at the beginning, and the initial location of all clients can be denoted as $\{\pi_i^0\}_{i=1}^n$, where $\pi_i^0 \in R^m$ is a probability vector with each element $\pi_i^0(j)$ representing the probability of client i within the edge AP j at time slot 0. We represent the transition probability of clients with the variables $\{P_i^t\}_{i=1}^n$ at the time slot t. In this way, we can get the next location as:

$$\pi_i^{t+1} = \pi_i^t P_i^t = \pi_i^0 \prod_{k=0}^{t} P_i^k, \forall i \in \mathcal{N}. \tag{8}$$

Moreover, mobile clients could also be heterogeneous in terms of their computation capabilities, and the time required for local training is also different. To address the heterogeneity in computational capabilities across mobile clients, we implement a synchronized aggregation signal, i.e., the model aggregation deadline τ. This deadline is intentionally set slightly longer than the time required by the lowest straggler client to complete its I rounds of local model updates, ensuring universal the participation of all mobile clients in model aggregation. As mobile clients may roam across several edge APs before deadline, we can characterize the set of visited edge APs for mobile client i before the deadline τ as:

$$\mathcal{S}_i = \prod_{t=1}^{\tau} \pi_i^{t-1} \bigvee \pi_i^t, \forall i \in \mathcal{N}, \tag{9}$$

where for two vectors π_i^t and π_i^{t-1}, the operator $\pi_i^t \bigvee \pi_i^{t-1} = (\max\{\pi_i^t(1), \pi_i^{t-1}(1)\}, \ldots, \max\{\pi_i^t(m), \pi_i^{t-1}(m)\})^T$ denotes the element-wise maximum operator, and $\mathcal{S}_i^t = \pi_i^t$ is the state information of mobile client i at time slot t. The parameter $\|\mathcal{S}_i\|_1$ denotes the number of edge APs, and is related to the transition probability P and the deadline τ, which we discuss in next.

C. Problem Formulation

Before formulating the client-edge association control problem, we first analyse the mobility-aware FL with some assumptions and notations:

1) The loss functions $E_{x|y=k}[\log f_k(x,\omega)]$ are differentiable and β-smooth;
2) The model gradients $\nabla_\omega E_{x|y=k}[\log f_k(x,\omega)]$ are bound, i.e., $||\nabla_\omega E_{x|y=k}[\log f_k(x,\omega)]|| \leq \delta$;
3) The gradient $\nabla_\omega E_{x|y=k}[\log f_k(x,\omega)]$ is λ_k-Lipschitz for each class $k \in \{c\}$, i.e., $||\nabla_\omega E_{x|y=k}[\log f_k(x,\omega_i)] - \nabla_\omega E_{x|y=k}[\log f_k(x,\omega_j)]|| \leq \lambda_k ||\omega_i - \omega_j||$.

Let $\bar{\omega}^t$ denotes the model weight after the t-th update in the centralized setting. With these notations, we present the following theorem for the model convergence analysis of client mobility in HFL.

Theorem 1: After the g^{th} global model aggregation in HFL, we have the following inequality for model weight divergence:

$$\|\omega^g - \bar{\omega}^g\| \leq \zeta^{EI} \|\omega^{g-1} - \bar{\omega}^{g-1}\|$$

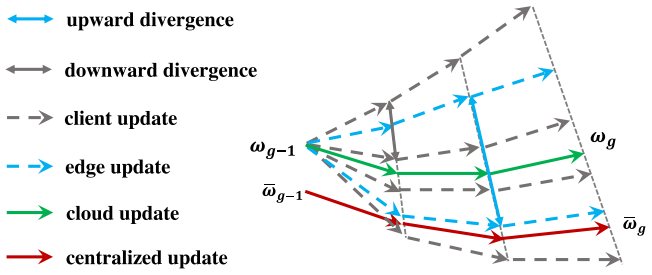

Fig. 3. The model weight divergence in HFL.

$$+ \underbrace{\beta \sum_{j=1}^{m} \|\hat{d} - \bar{d}_j\|_1}_{\mathcal{U}} + \underbrace{\alpha \sum_{i=1}^{n} \sum_{j \in \mathcal{S}_i} H_{i,j} \|d_i - \bar{d}_j\|_1}_{\mathcal{D}}$$
$$+ \eta \sum_{j=1}^{m} \left(\sum_{l=1}^{EI} (\zeta)^l + \sum_{l=1}^{I} (\gamma)^l \right) h_j(I), \tag{10}$$

where $\gamma = 1 + \eta \sum_{k=1}^{c} d_i(y=k)\lambda_k$ and $\zeta = 1 + \eta \sum_{k=1}^{c} \hat{d}(y=l)\lambda_k$. The symbol $\mathcal{U}$ denotes the upward data distribution divergence between edge APs and cloud server, and $\mathcal{D}$ is the downward one between edge APs and clients.

Proof: See Appendix A, available online. □

As shown in Fig. 3, the gray color lines associated with client update denote the local model training performed by mobile clients, and the light blue ones (edge update) represents the aggregation of these local models at edge APs. As for the green ones and red ones, these two denote the further aggregation at the cloud server and the centralized update where all client data are pooled together for model training. The model weight divergence of the g^{th} global aggregation mainly comes from two parts: the model weight divergence of the $(g-1)^{th}$ global aggregation, and the data distributions divergence, i.e., the upward data distribution divergence $\mathcal{U}$ and the downward one $\mathcal{D}$. According to the above convergence analysis, the goal of this work is to leverage client mobility to reshape the data distributions of edge APs, aiming to minimize both upward and downward data distribution divergence in HFL. Therefore, we formulate the following $\mathcal{P}1$ problem:

$$\mathcal{P}1: \min_{H} \quad \alpha \sum_{i=1}^{n} \sum_{j \in \mathcal{S}_i} H_{i,j} \|d_i - \bar{d}_j\|_1 + \beta \sum_{j=1}^{m} \|\hat{d} - \bar{d}_j\|_1 \tag{11}$$

$$\text{s.t.} \sum_{j \in \mathcal{S}_i} H_{i,j} = 1, \quad \forall i \in \mathcal{N}, \tag{11a}$$

$$\sum_{i=1}^{n} H_{i,j} c_{i,j} \leq b_j, \quad \forall j \in \mathcal{M}, \tag{11b}$$

$$H_{i,j} \in \{0,1\}, \quad \forall i \in \mathcal{N}, j \in \mathcal{M}. \tag{11c}$$

The first constraint (11a) ensures that each client can only be associated to one edge AP during her mobility trajectory. The second one (11b) is to make the connection of each edge AP not so dense due to limited bandwidth resources, and the last constraint (11c) indicates that $H_{i,j}$ is a binary decision variable. We

recall that the data distributions of edge APs $\{\bar{d}_j\}_{j=1}^m$ depends on these of clients $\{d_i\}_{i=1}^n$ in a non-linear way, because the decision variables H appearing in the numerator and denominator of $\{\bar{d}_j\}_{j=1}^m$. Therefore, the above $\mathcal{P}1$ problem is a non-linear 0-1 programming problem [23], [24], which is known to be NP-hard problem.

III. Mobility-Aware FL Framework

In this section, we first analyse key parameters in client mobility model, and discuss two mobility patterns with different settings. According to two mobility patterns, we present the mobility-aware FL framework.

A. Mobility Trajectory Analysis

As discussed in $\mathcal{P}1$ problem, to minimize the data distribution divergence is related to the mobility trajectories of clients, which depends on the transition probability P and the deadline τ. According to the different mobility patterns of clients [25], we can classify their mobility trajectories into two categories: deterministic mobility trajectory and the non-deterministic mobility trajectories. Next, we first discuss the characterize of these two mobility trajectory by exploring the transition probability P and the deadline τ.

A deterministic mobility trajectory is characterized by high regularity and predictability, where the future positions of clients can be predicted with a high degree of certainty based on past behaviors and known factors [26]. Periodic mobility trajectory is the classical deterministic mobility trajectory, which represents that clients commute among a small set of fixed locations, such as home, school and restaurant. For simplicity, we denote $C_i \subseteq \mathcal{M}$ as the regularly visited locations of client $i \in \mathcal{N}$. The transition probability over the location set C_i of the client i is simply the sequence of successive locations as follows:

$$P_i(x,y) = \begin{cases} p_x^i, & \text{if } y = x, \forall x \in C_i \ , \\ 1 - p_x^i, & \text{if } y = (x+1) \mod |C_i| \ , \\ 0, & \text{otherwise} \ , \end{cases} \tag{12}$$

where p_x^i is the stay probability of client i at location $x \in C_i$, and $1 - p_x^i$ is the transition probability to next location. A lower p_x^i implies that the client is more likely to roam across more edge APs, leading to more client-edge association with more optimization space. As shown in Fig. 4, the trajectory of mobile client with periodic mobility pattern is a cycle movement trajectory.

A non-deterministic mobility trajectory (also referred to as a stochastic mobility trajectory) is characterized by uncertainty and randomness [27]. As for random mobility pattern, we assume that mobile clients are uniformly and randomly distributed over all edge APs at the beginning, and the probability of moving to different edge APs is stochastic and uncertain. Therefore, the element of transition probability matrix can be written as:

$$P_i(x,y) = \begin{cases} p_x^i, & \text{if } y = x, \forall x \in \mathcal{M} \ , \\ \frac{1-p_x^i}{|\mathcal{M}|-1}, & \text{if } y \neq x, \forall y \in \mathcal{M} \ . \end{cases} \tag{13}$$

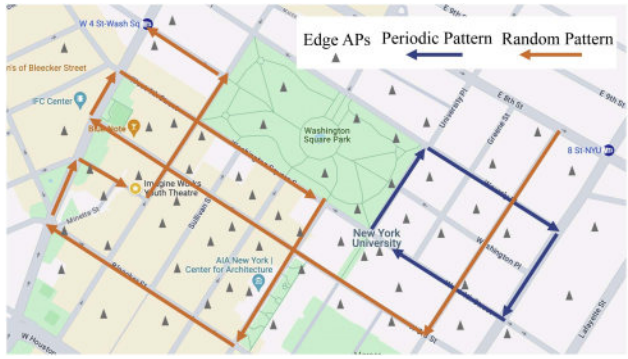

Fig. 4. Illustrations for two classical mobility patterns.

The mobility trajectory of random mobility pattern could be quiet mussy in the Fig. 4, and the next location of each client is hard to predict in advance. As for these two mobility patterns, the deadline τ determines the content of the visited edge APs set $\mathcal{S}_i$ of clients. According to the equation in (9), each client $i \in \mathcal{N}$ will move across more different edge APs with bigger τ.

B. FL Framework Design

According to the mobility patterns we discussed above, the targeted $\mathcal{P}1$ problem can be considered as either an offline optimization problem or the online one. In the offline setting, the mobility trajectory of clients with periodic mobility pattern (that is the set of edge APs $\mathcal{S}_i$) are already known, we aim to solve the above $\mathcal{P}1$ problem by finding the optimal client-edge association to minimize both the upward and downward data distribution divergence, simultaneously. If the number of edge APs in the set $\mathcal{S}_i$ is large, each mobile client $i \in \mathcal{N}$ would have more optimization space to realize more efficient client-edge association in HFL. In the online setting, the mobility trajectory of clients with random mobility pattern (that is the set of edge APs $\mathcal{S}_i$) is hard to predict, and we need to determine whether to upload the local model at each time slot.

To this end, we develop a novel mobility-aware FL framework named ALPHA which is defined as follows: ALPHA is a mobility-aware FL framework which contains the offline binary programming and the online binary programming respectively.

1) In the offline setting, the mobility trajectory of clients with periodic mobility pattern (that is the set of edge APs $\mathcal{S}_i$) are already known, ALPHA is to solve the above $\mathcal{P}1$ problem by finding the optimal client-edge association to minimize upward and downward data distribution divergence, simultaneously.
2) In the online setting, the mobility trajectory of clients with random mobility pattern (that is the set of edge APs $\mathcal{S}_i$) is hard to predict, and ALPHA is to tackle the online problem for the decision about whether to upload model parameters at the current time slot t.

The problem in these two settings are quiet different which mainly rest with mobility patterns. Thus, we need to work out the problem in these settings seperately.

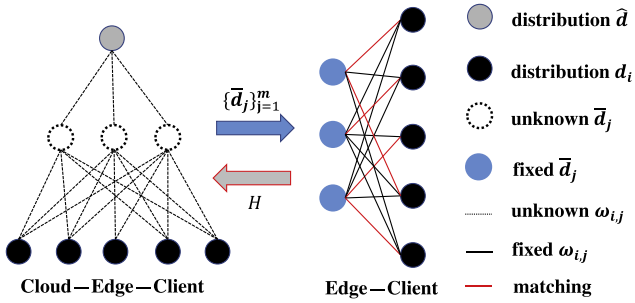

Fig. 5. Illustrations for alternating optimization methods.

IV. Design of Alpha

In this section, we first provide the design of offline mobility scenario, and then consider the online one.

A. Offline Scenario

For the offline scenario, we need to solve the optimization problem $\mathcal{P}1$ with a fixed set of edge APs $\mathcal{S}_i$. We denote the objective function in problem $\mathcal{P}1$ as the variable $\mathcal{F}$. As shown in Fig. 5, we leverage the alternating optimization theory [28] to solve the challenging problem $\mathcal{P}1$ by the following three steps: (1) we first fix the data distributions of edge APs with a certain probabilities $\bar{\mathbf{d}}$, which denotes the set of $\{\bar{d}_j\}_{j=1}^m$. In this way, the objective function in $\mathcal{P}1$ will be transformed as:

$$\mathcal{F}(\bar{\mathbf{d}}) = \alpha \sum_{i=1}^{n} \sum_{j \in \mathcal{S}_i} H_{i,j} w_{i,j} + \Delta, \tag{14}$$

where $w_{i,j} = \|d_i - \bar{d}_j\|_1$, and $\Delta = \beta \sum_{j=1}^m \|\bar{d}_j - \hat{d}\|_1$ is the upward data distribution divergence, which is a constant when $\bar{\mathbf{d}}$ is fixed. (2) Then, we can solve the $\mathcal{P}1$ problem with the new objective function $\mathcal{F}$ by developing a LP-based bipartite b-matching algorithm to determine the decision variable; (3) After that, we evaluate whether the initialized probabilities $\bar{\mathbf{d}}$ are the solution with the satisfied performance. If not, we further search for the initialized probabilities $\bar{\mathbf{d}}$ for an efficient solution H with a smaller value of the objective $\mathcal{F}$. As for Step (1), there exist many possible probabilities for data distributions of edge APs $\bar{\mathbf{d}}$. We first need to narrow the space of $\bar{\mathbf{d}}$ to avoid invalid initial values and reduce the search cost. According to the definition of data distributions of edge APs, i.e., $\bar{d}_j = \frac{\sum_{i=1}^n H_{i,j} a_i d_i}{\sum_{i=1}^n H_{i,j} a_i}$, we can further obtain that: the data distribution of edge AP $\bar{d}_j$ is always less than the maximum data distribution of clients in edge APs set S_j, and greater than the minimum data distribution of clients in set S_j. In particular, we can have $\bar{d}_j^l \le \bar{d}_j \le \bar{d}_j^u$ where $\bar{d}_j^l = \min\{d_i, \forall i \in S_j\}$ and $\bar{d}_j^u = \max\{d_i, \forall i \in S_j\}$. Therefore, we can first fix the values of $\bar{\mathbf{d}}$ with its lower boundary $\bar{\mathbf{d}}^l$, and then conduct step search without beyond its upper boundary $\bar{\mathbf{d}}^u$.

The optimization problem in Step (2) can be considered as a weighted bipartite b-matching problem, which is also a NP-hard problem. A bipartite b-matching in graph $G = (\{\mathcal{M}, \mathcal{N}\}, W)$ is to find a set of weight edges in W where one vertex in $\mathcal{M}$ has at most b edges (i.e., a degree constraint) while the other in $\mathcal{N}$ are unique. This implies that the edge AP can handle multiple clients, and each client can only be collected to one edge AP. The bipartite b-matching problem is an extension of classical bipartite matching problem with more complex constraints, and we solve this problem by using the LP relaxation with dependent rounding technique. We first relax the integer decision variable $H_{i,j}, i.e., 0 \le H_{i,j} \le 1$ to obtain the optimal fractional solution. Second, we introduce a dependent rounding method to get the integer one. Specifically, let $\overline{W}$ be the set of edges with fractional values. Repeat the following steps if the set $\overline{W}$ is non-empty: a) find a simple cycle or maximal path P in the subgraph $(\{\mathcal{M}, \mathcal{N}\}, \overline{W})$ where the simple cycle or maximal path is to ensure that the set of P can be divided into two matchings M_1 and M_2 for two rounding directions (0 or 1) [29]; b) In these two matchings, we have four options for fractional $\bar{H}_{i,j}$ to get the integer one $H_{i,j}$ with different probabilities as follows:

$$H_{i,j} = \begin{cases} \bar{H}_{i,j} + \omega_1, & w.p. \ \frac{\omega_2}{\omega_1+\omega_2}, \ \text{if} \ \bar{H}_{i,j} \in M_1, \\ \bar{H}_{i,j} + \omega_2, & w.p. \ \frac{\omega_1}{\omega_1+\omega_2}, \ \text{if} \ \bar{H}_{i,j} \in M_1, \\ \bar{H}_{i,j} - \omega_1, & w.p. \ \frac{\omega_2}{\omega_1+\omega_2}, \ \text{if} \ \bar{H}_{i,j} \in M_2, \\ \bar{H}_{i,j} - \omega_2, & w.p. \ \frac{\omega_1}{\omega_1+\omega_2}, \ \text{if} \ \bar{H}_{i,j} \in M_2, \end{cases} \tag{15}$$

where ω_1 is the minimizer of two kinds of distances: the minimum distance between the nodes in M_1 to 1 and the minimum distance between the nodes in M_2 to 0, and the ω_2 is similar to ω_1 which can be described as:

$$\begin{cases} \omega_1 = \min\{\min\{1 - \bar{H}^1_{i,j}, \forall \bar{H}^1_{i,j} \in M_1\}, \\ \qquad\qquad \min\{\bar{H}^2_{i,j}, \forall \bar{H}^2_{i,j} \in M_2\}\}, \\ \omega_2 = \min\{\min\{\bar{H}^1_{i,j}, \forall \bar{H}^1_{i,j} \in M_1\}, \\ \qquad\qquad \min\{1 - \bar{H}^2_{i,j}, \forall \bar{H}^2_{i,j} \in M_2\}\}. \end{cases} \tag{16}$$

We next conduct Step (3) to search for the optimal values of $\bar{\mathbf{d}}$. There exist many possible $\bar{\mathbf{d}}$ with large search space due to the coupling of edge APs. Suppose there are k possible values for $\bar{d}_j$ for each edge AP, then we need to evaluate m^k possible probability profile if there are m edge APs. To solve this intractable problem, we design a minimum overlapped search method with mobility pattern information to reduce the search space. The key idea of this method is to identify an loose coupling search order among edge APs, reducing the exponential complexity of enumerating all the possible probability profile. From the above discussion, the number of clients connected with corresponding edge APs under periodic mobility pattern is limited ($|\mathcal{S}_j| << |\mathcal{N}|, \forall j \in \mathcal{M}$), and with low coupling among edge APs. Therefore, we can regard the number of shared clients between edge AP j and the other edge APs, i.e., $o_j = \sum_{i=1}^N |\mathcal{S}_i \cap \{j\}|$ as its overlapped degree. Furthermore, we can obtain the search order among edge APs based on its overlapped degree to find the efficient values of $\bar{\mathbf{d}}$ through only one loop.

From the above discussion, we can combine these three steps to form an efficient solution to solve the problem $\mathcal{P}1$. As shown in Algorithm 1, we first narrow the space of $\{\bar{d}_j\}_{j=1}^m$ by finding the lower and upper boundary, and compute the overlapped degree to find the search order (line 1 to line 3). After that, we initialize the data distributions of edge APs $\bar{\mathbf{d}}$ and reorder

Algorithm 1: Offline Control.

Input: number of clients m, number of edge APs n, big constant Z, search step ς, data distribution of clients $\{d_i\}_{i=1}^n$, mobility trajectory of clients $\{\mathcal{S}_i\}_{i=1}^n$, the set of clients within edge APs $\{\mathcal{S}_j\}_{j=1}^m$
Output: the decision variable H

1 **for** $j \in \mathcal{M}$ **do**
2 $\bar{d}_j^l = \min\{d_i\}_{i=1}^{|S_j|}$, $\bar{d}_j^u = \max\{d_i\}_{i=1}^{|S_j|}$;
3 $o_j \leftarrow \sum_{i=1}^N |\mathcal{S}_i \cap \{j\}|$;
4 reorder edge APs $\mathcal{M}$ based on $\{o_j\}_{j=1}^m$;
5 initialize $\mathcal{F}^* \leftarrow Z$ and $\bar{\mathbf{d}} \leftarrow \bar{\mathbf{d}}^l$;
6 **for** $j \in \mathcal{M}$ **do**
7 **while** $\bar{d}_j \le \bar{d}_j^u$ **do**
8 get $\bar{H}$ with LP under fixed $m, n, \bar{\mathbf{d}}, \{d_i\}_{i=1}^n$;
9 find the edge set $\overline{W}$ with fractional value;
10 **while** $|\overline{W}| \neq \emptyset$ **do**
11 **for** $\bar{H}_{i,j} \in \bar{H}$ **do**
12 compute $H_{i,j}$ according to Eq.(15);
13 compute $\mathcal{F}$ with H according to Eq.(14);
14 **if** $\mathcal{F} \le \mathcal{F}^*$ **then**
15 $\bar{d}_j^* \leftarrow \bar{d}_j$, $\mathcal{F}^* \leftarrow \mathcal{F}$, $H \leftarrow \bar{H}$;
16 update $\bar{d}_j \leftarrow \bar{d}_j + \varsigma$;
17 **return** H.

edge APs based on the overlapped degree (line 4 to line 5). In order to determine the decision variable H, we leverage the LP with dependent rounding technique to solve bipartite b-matching with fixed $\bar{\mathbf{d}}$ (line 8 to line 12). After that, we evaluate whether the initialized probabilities $\bar{\mathbf{d}}$ are the solution with the satisfied performance. If not, we further search for the initialized probabilities $\bar{\mathbf{d}}$ for a good solution H with a smaller value $\mathcal{F}$ (line 13 to line 16).

This linear relaxation with dependent rounding approach ensures that the degree of each node closely approximates its initial fractional degree, effectively balancing multiple constraints. Moreover, this approach guarantees negative correlation between edges in bipartite graph, which helps maintain better fairness and stability during the allocation process. Using ζ to denote the computation complexity of LP with dependent rounding technique, and ς as the step of search, the computational complexity of Algorithm 1 is $O(m\zeta\frac{\max\{\bar{d}_j^u\}_{j=1}^m - \min\{\bar{d}_j^l\}_{j=1}^m}{\varsigma})$, which mainly depends on the number of edge APs, the cost of search for $\{\bar{d}_j\}_{j=1}^m$ and the complexity of LP with dependent rounding.

Theorem 2: We denote $\mathcal{F}(\{\bar{d}_j^*\}_{j=1}^m, H^*)$ as the optimal value of problem $\mathcal{P}1$, and there exist constant χ and ξ satisfying:

$$E[\mathcal{F}(\{\bar{d}_j^*\}_{j=1}^m, H^*)] \le \chi E[\mathcal{F}(\{\bar{d}_j\}_{j=1}^m, H\})] + \xi \tag{17}$$

where $\chi = \frac{c_{\max}\cdot d_{\max}}{c_{\min}\cdot d_{\min}}$ and $\xi = (n\cdot\alpha\frac{c_{\max}}{c_{\min}} + m\cdot\beta)(\frac{d_{\max}}{d_{\min}} - 1)$.

Proof: See Appendix B, available online. □

Algorithm 2: Online Control.

Input: number of clients m, number of edge APs n, learning rate γ, data distribution $\{d_i\}_{i=1}^n$
Output: $\{H^t\}_{t=1}^\tau$

1 initialize Φ and H with 0;
2 **for** $t = 1 : \tau$ **do**
3 **for** $i = 1 : n$ **do**
4 the current edge AP $j \leftarrow \mathcal{S}_i^t$;
5 **if** $c_{i,j}^t < b_j^t$ *and* $c_{i,j}^t \Phi_j^t < w_{i,j}^t$ **then**
6 $H_{i,j}^t \leftarrow 1$;
7 $b_j^{t+1} \leftarrow b_j^t - c_{i,j}^t H_{i,j}^t$;
8 $\Phi_j^{t+1} \leftarrow \Phi_j^t + \gamma(b_j - c_{i,j}^t H_{i,j}^t)$;
9 **return** $\{H^t\}_{t=1}^\tau$.

B. Online Scenario

For the online scenario, clients only know the current location in each time slot t without the future information, and should make an irrevocable immediate decision on whether to upload model parameters for aggregation at the current edge AP. Specifically, at each time slot t, we should make an online decision $H_{i,j}^t$ on each client i with the the bandwidth needed to connect to the current edge AP j, i.e., $c_{i,j}^t$, the remaining bandwidth resource on the current edge AP b_j^t and the divergence of data distribution between associated client-edge, i.e., $w_{i,j}^t = \|d_i^t - \bar{d}_j^t\|_1$, where the data distribution of edge AP j is accumulated over time slots, i.e.,

$$\bar{d}_j^t = \sum_{k=1}^{t-1} \frac{\sum_{i=1}^n a_i H_{i,j}^k d_i^k}{\sum_{l=1}^{t-1}\sum_{i=1}^n a_i H_{i,j}^l}, \forall j \in [1, m]. \tag{18}$$

However, it is hard to establish the relation among these three variables in the original problem. Inspired by the dual theory [30], we determine the relation among these three variables in the dual problem. Similar to the offline setting, we first relax 0-1 decision variables $\{H^t\}_{t=1}^\tau$ into continuous variables $0 \le H_{i,j}^t \le 1$, and the dual problem of online LP problem is:

$$\mathcal{P}2: \max_{\{\Phi^t\}_{t=1}^\tau, \{\Psi\}_{t=1}^\tau} \sum_{t=1}^\tau \sum_{i=1}^n \sum_{j\in\mathcal{S}_i^t} (\Phi_j^t b_j^t + \Psi_i^t) \tag{19}$$

$$\text{s.t.} \quad \Phi_j^t c_{i,j}^t + \Psi_i^t \ge w_{i,j}^t, \tag{19a}$$

$$\Phi_j^t \ge 0, \Psi_i^t \ge 0, \ \forall i, \forall j, \forall t, \tag{19b}$$

where Φ and Ψ are the dual decision variables. To determine the decision variable H, we need to build the relation between the optimal solution of the prime problem and the dual problem, which are denoted as H^*, Φ^* and Ψ^*. From the complementary slackness condition, we can derive H^* without a rounding procedure:

$$H_{i,j}^* = \begin{cases} 1, & \text{if } c_{i,j}\Phi_j^* < w_{i,j}, \\ 0, & \text{if } c_{i,j}\Phi_j^* > w_{i,j}, \end{cases} \tag{20}$$

where $H_{i,j}$ may be the non-integer value when $c_{i,j}\Phi_j^* = w_{i,j}$.

In order to identify the relation among the three variables, we need to rewrite an equivalent form of the problem $\mathcal{P}2$, which

only includes decision variables Φ by removing the dual decision variables Ψ with plugging the constraints $\Psi_i^t \geq w_{i,j}^t - \Phi_j^t c_{i,j}^t$ and $\Psi_i^t \geq 0$ into the objective function:

$$\max_{\Phi} f(\Phi) = \sum_{t=1}^{\tau} \sum_{i=1}^{n} \sum_{j \in \mathcal{S}_i^t} (\Phi_j^t b_j^t + (w_{i,j}^t - c_{i,j}^t \Phi_j^t)^+), \quad (21)$$

where $(\cdot)^+$ denotes the positive part function. In this way, we can establish the relation among these three variables by solving $f(\Phi)$. To address this problem, we first compute the subgradient evaluated with variable Φ_j^t at time slot t as:

$$\partial_{\Phi_j^t}(\Phi_j^t b_j^t + (w_{i,j}^t - c_{i,j}^t \Phi_j^t)^+) = b_j - c_{i,j}^t H_{i,j}^t. \quad (22)$$

Now, we can state our proposed approach in Algorithm 2: sub-gradient based online LP algorithm, which makes an online decision of the variable H^t immediately after observing the value of $(w_{i,j}^t; c_{i,j}^t)$. In particular, the update from Φ_j^t to Φ_j^{t+1} can be viewed as a projected stochastic subgradient ascent method in line 8. At the beginning of the algorithm, the edge APs prefer to admit the clients when the dual variable Φ_j is small, and sufficient bandwidth resource b_j is sufficient. After several iterations, the value of Φ_j becomes large, and the bandwidth resource b_j is small. Thus, few clients will be selected in the latter time slots according to the complementary slack condition.

Next, we will analyze the performance of Algorithm 2. Before giving the related result, we first introduce some assumptions on the input of LP [31]. We assume:

1) The coefficient pair $(w_{i,j}, c_{i,j})$, are i.i.d. sampled from an unknown distribution;
2) There exist constant $\overline{w}$ and $\overline{c}$ such that $w_{i,j} \leq \overline{w}$ and $c_{i,j} \leq \overline{c}, \forall i \in [1, n], \forall j \in [1, m]$;
3) There exist $\underline{b}$ and $\overline{b}$ such that $\underline{b} \leq b_j \leq \overline{b}, \forall j \in [1, m]$.

We formally define the regret as the difference between the optimal objective values of the original problem R_τ^*, and the objective value derived by the Algorithm 2 as R_τ.

Theorem 3: Under the above assumptions with the deadline τ, the regret of Algorithm 2 satisfies:

$$\mathbb{E}[R_\tau - R_\tau^*] \leq 3\gamma\tau n(\overline{c} + \overline{b})^2. \quad (23)$$

where γ is the learning rate and n denotes the client number.

Proof: See Appendix C, available online. □

V. Performance Evaluation

In this section, we first describe the experiment setting, and then evaluate the performance of our proposed ALPHA.

A. Experiment Setting

Datasets: we perform extensive experiments on three public image datasets and a real-world trajectory dataset, which are described as follows:

- *SVHN*, which is captured from Google Street View images for a significant real-world problem, and widely used in field of object detection and pattern recognition [32].
- *GTSRB*, which is the german traffic sign recognition dataset including 43 categories images with varying light conditions, rich backgrounds and unbalanced category distribution [33].
- *CIFAR10*, which is labeled subsets of the 80 million tiny colorful images dataset, has 10 mutually exclusive classes, and is a classical dataset for image classification [34].
- *TELECOM*, which is provided by Shanghai Telecom, includes more than 7.2 million records of mobility trajectory by accessing the interent through 3233 base stations from 9481 mobile phones for six months [35].

Parameter Setting: We use CNN with three convolutional layers and three fully connected layers for SVHN, GTSRB and CIFAR10. The learning rate and batch size of model training is 0.02 and 32. As for TELECOM, we utilize MLP with three fully connected layers. The learning rate and batch size of model training is 0.01 and 32. For all models, we set the decay factor and momentum to their default values of 0. To realize the non-IID data distribution, we employ Dirichlet distribution to characterize the identicalness among clients as shown in previous studies [36], [37]. Specifically, we sample a Dirichlet distribution $Dir(\kappa b)$ for different non-IID data distributions, where b characterizes the prior class distribution and κ is a concentration parameter. With $\kappa \to \infty$, all clients have identical distribution to the prior, and $\kappa \to 0$ is the other extreme. The communication cost of clients and the bandwidth resource of edge APs are subject to a random distribution, i.e., $c_{i,j} \in [0, 1], b_j \in [5, 10], \forall i \in \mathcal{N}, j \in \mathcal{M}$. As for the number of mobile clients and edge APs, we take $m \in \{5, 10\}$ and $n \in \{20, 50, 100\}$ into consideration. The LP method is written in python with PuLP.

Baselines: we mainly make a comparison among the following mobility optimization frameworks:

- *ORACLE*, which makes a precise decision for all clients on where and when to upload model parameters for edge aggregation with brute-force search [38].
- *MACFL*, which aims to realize mobility-aware cluster federated learning by utilizing clustering algorithm to calculate the similarity of mobile clients for model aggregation [14].
- *FedMobile*, which focuses on the mobility of clients without considering the edge APs and cloud server, and aggregates local models of mobile clients in random manner with the D2D communication [39].

B. Overall Performance

To make the results on these two real-world datasets more persuasive, all experiments in Figs. 6, 7 and 8 are performed three times with $m = 10, n = 100$, and the solid line means the averaged value of three experiments, while the size of the shaded area represents the magnitude of fluctuations. Fig. 6 plots the average accuracy of the four mobility optimization methods on three public image datasets, i.e., GTSRB, SVHN and CIFAR10. As the number of communication rounds increases, the average accuracy increases rapidly at first, then slows down until convergence. In Fig. 6(a) and (c), it is obvious that the central

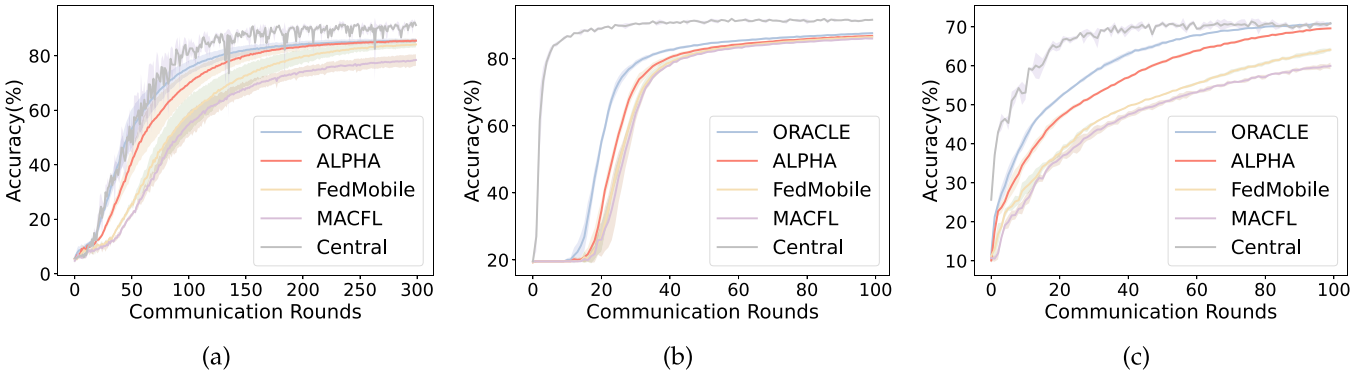

Fig. 6. The performance comparison of average accuracy among four mobility frameworks with central training in HFL validated on three public image datasets: (a) GTSRB; (b) SVHN; (c) CIFAR10.

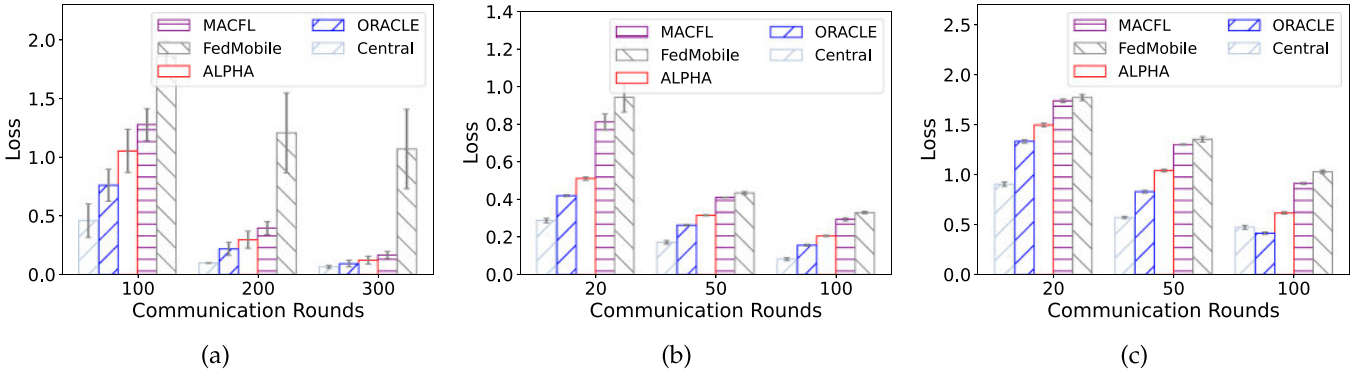

Fig. 7. The performance comparison of training loss among four mobility frameworks with central traning in HFL validated on three public image datasets: (a) GTSRB; (b) SVHN; (c) CIFAR10.

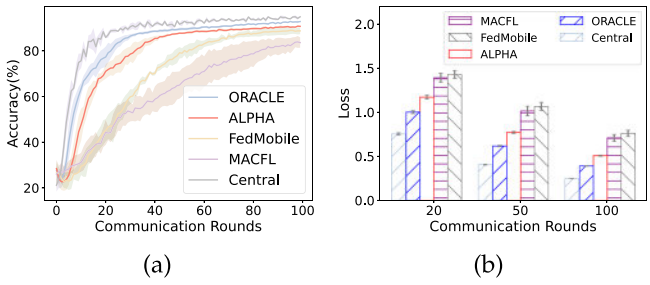

Fig. 8. The performance comparison of four mobility frameworks in HFL validated with the real-world dataset TELECOM on : (a) Average Accuracy; (b) Training Loss.

training has the highest average accuracy without data shift, and ORACLE and ALPHA are close to the central training while having faster convergence speed and better model performance than both MACFL and FedMobile. In Fig. 6(b), the advantage of ALPHA comparing with FedMobile and MACFL is not obvious due to the dataset SVHN which is relatively simpler compared to the other two datasets, resulting in less distinction between these four mobility frameworks. Fig. 7 plots the training loss of the four mobility optimization frameworks with the central training on three public image datasets, i.e., GTSRB, SVHN and CIFAR10. As shown in Fig. 7(a), the training loss of these four methods decreases with the communication rounds, and it is obvious that the training loss of the central training is the minimum one. Moreover, we can find that ORACLE has the minimization of training loss and ALPHA is close to ORACLE while better than MACFL and FedMobile. In Fig. 7(b) and (c), we further validate the efficiency of ALPHA on SVHN and CIFAR10.

Fig. 8 presents the performance comparison of four mobility frameworks with the central training on the real-world mobility trajectory dataset TELECOM. In Fig. 8(a), ORACLE and ALPHA are close to the central training, and have a superior learning performance compared to the other two approaches, i.e., FedMobile and MACFL. The main reason behind this phenomenon is that ORACLE and ALPHA pay much attention to minimize the impact of uncertain client mobility, and aim to reshape data distributions of edge APs to be i.i.d, which is much related with the learning performance in HFL. FedMobile and MACFL only focus on model aggregation by given data distribution, and are incapable of addressing uncertain client mobility. In Fig. 8(b), we can find that ORACLE has the minimization of training loss and ALPHA is close to ORACLE while better than MACFL and FedMobile, which further validate the efficiency of our proposed ALPHA.

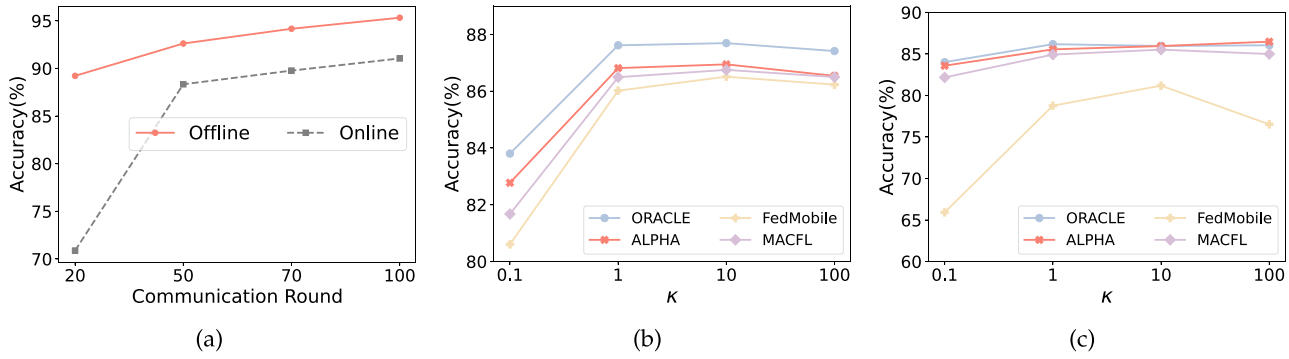

Fig. 9. The performance comparison in HFL with: (a) Accuracy of two mobility scenarios; (b) Impact of κ on the public dataset SVHN; (c) Impact of κ on the real-world dataset TELECOM.

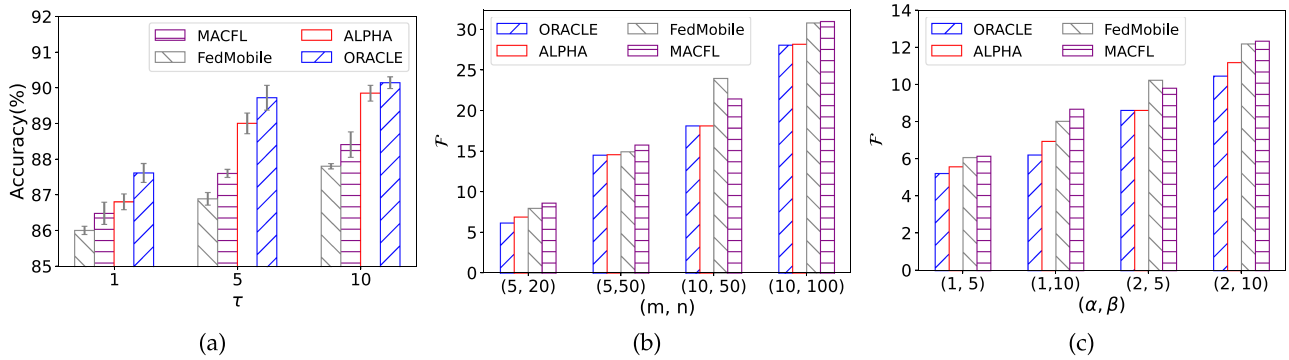

Fig. 10. The performance comparison in HFL with: (a) Impact of τ; (b) the number of edge APs and clients, i.e., (m, n); (c) the weight of downward data distribution divergence and upward one, i.e., (α, β).

C. Key Parameter Analysis

In Fig. 9, we perform the performance comparison in HFL on the impact of κ and two mobility scenarios. Fig. 9(a) presents the prediction accuracy of ALPHA in the offline mobility scenario and the online one. As can be seen from Fig. 9(a), the prediction accuracy of ALPHA both in offline scenario and online one increases with communication rounds. Comparing with the prediction accuracy in offline scenario, the model performance of ALPHA in online scenario is less satisfying due to the uncertainty of mobility trajectory. As stated in the part of parameter setting, we utilize κ to character the non-i.i.d degree. In Fig. 9(b) and (c), we explore the impact of κ on the public image dataset SVHN and the real-world mobility trajectory dataset TELECOM. When $\kappa = 0.1$ with large non-i.i.d degree, the average accuracy of all four mobility frameworks are comparative low, and the performance gap between these four mobility optimization frameworks is large. With the value of κ growing with little non-i.i.d degree, the average accuracy of all four mobility frameworks are increasing, but the performance gap decreases.

In Fig. 10, we perform the performance comparison in HFL on the impact of τ, the number of edge APs and clients, i.e., (m, n), and the weight of downward and upward divergence, i.e., (α, β). In Fig. 10(a), we further study the impact of τ on performance of mobility optimization, and use the variable τ to character the time deadline of mobility. Each client $i \in \mathcal{N}$ will have more time to move across different edge APs with bigger τ. As shown in Fig. 10(a), the mobility trajectory with $\tau = 10$ has the highest averaged accuracy comparing with the mobility scenario with $\tau = 5$ or $\tau = 1$, where the gray error bar is the variance of three experiments. The reason of this phenomenon is that clients have more edge APs to choose for model aggregation with bigger τ, which is useful for data distribution divergence minimization. Further, we make a performance comparison among ALPHA with the other three methods by considering different parameter settings. The parameter $\mathcal{F}$ in vertical coordinate is the total divergence of data distributions which is the optimization objective. As can be seen from Fig. 10(b), the total divergence of data distributions increases with the number of edge APs and clients (m, n), and ALPHA is better than FedMobile and MACFL. Moreover in Fig. 10(c), we also use the weight parameters (α, β) to perform evaluation, and find similar results that the total divergence of data distributions among these four mobility optimization scheme increases with the value of (α, β).

Fig. 11 presents the performance comparisons with four different mobility frameworks on all datasets CIFAR10, SVHN, GSTRB and TELECOM by taking different mobility frequency into account. We use the variable p to character the mobility frequency of mobile clients, and each client $i \in \mathcal{N}$ will prefer moving another locations to staying at same one with little p, which means that $p = 0.1$ is the high mobility scenario. As shown in Fig. 11, we consider three different mobility frequency scenarios including high mobility scenario with $p = 0.1$, normal mobility scenario with $p = 0.5$ and nearly static scenario with

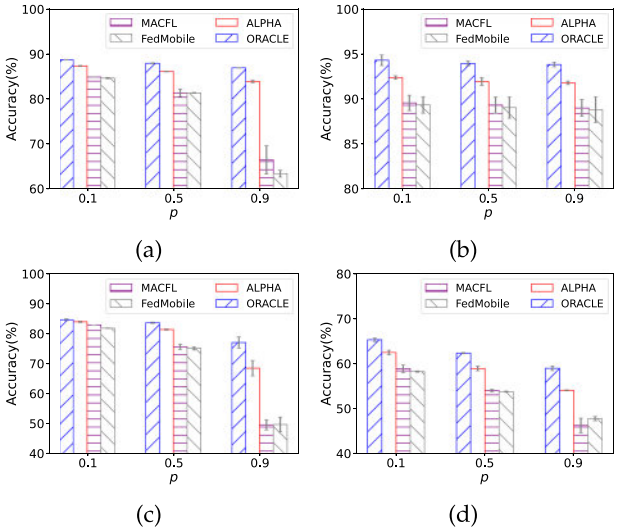

Fig. 11. The performance of ALPHA with different mobility frequency, i.e., p on datasets: (a) Accuracy on SVHN; (b) Accuracy on TELECOM; (c) Accuracy on GTSRB; (d) Accuracy on CIFAR10.

$p = 0.9$. Moreover, the experiment of each mobility scenario with different p has been performed with several times to validate the reliability and stability of the experimental results. As shown in Fig. 11, it is obvious that ORACLE and ALPHA has better model performance than both MACFL and FedMobile on all datasets. In high mobility scenario with $p = 0.1$, ALPHA has the highest average accuracy with frequent client mobility. This is because frequent client mobility creates new model aggregation opportunities at different edge APs, which is capable of reshaping the data distributions of edge APs to be i.i.d. As for the nearly static scenario with $p = 0.9$, the average accuracy declines a lot comparing to the one in high mobility scenario. In Fig. 11(a), the difference among high mobility scenario and nearly static scenario is not obvious due to the dataset SVHN which is relatively simpler compared to the other three datasets, resulting in less distinction between these two mobility scenarios. The model performance on these four datasets fluctuates a lot no matter in the high mobility scenarios or slow mobility scenario. This is because the performance of aggregated model not only depends on mobility frequency but also the other factors, i.e., task complexity and network topology, which will be explored furtherly in the future work.

In this work, we focus exclusively on the selection of edge APs for model uploading and aggregation for mobility-aware hierarchical FL, without considering the local model training methods of mobile clients or the model aggregation strategies at edge APs. To validate the robustness of our approach across various classical FL algorithms, we conducted additional experiments incorporating two classical FL algorithms: FedAvg and FedProx. As shown in Fig. 12(a), our proposed ALPHA demonstrates superior performance improvements over both FedMobile and MACFL, regardless of which FL algorithm is employed. However, comparing these two algorithms reveals that FedAvg achieves better performance than FedProx in hierarchical FL. This discrepancy stems from FedProx's introduction of a global

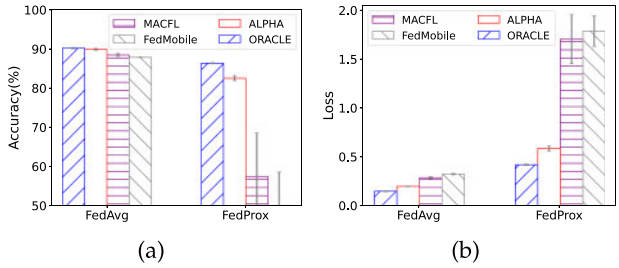

Fig. 12. The performance comparison of four mobility frameworks in HFL with different model aggregation methods (FedAvg and FedProx) on : (a) average accuracy; (b) training loss.

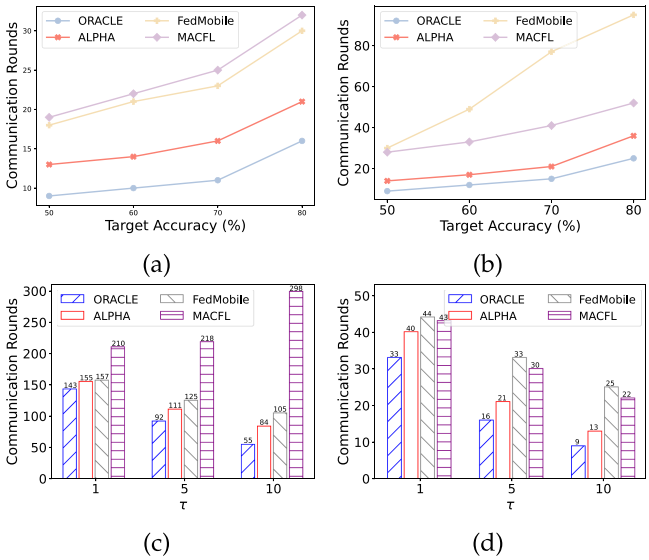

Fig. 13. The performance comparison of four mobility frameworks in HFL on : (a) Round-to-Accuracy on SVHN; (b) Round-to-Accuracy on TELECOM; (c) Accuracy-to-Round on GTSRB; (d) Accuracy-to-Round on TELECOM.

correction term during model training to address data shifts, whereas in hierarchical FL, mobile clients can only access the model of the corresponding edge APs as correction references. The inherent discrepancy between models of edge APs and the global model creates a performance degradation effect: greater discrepancies lead to poorer local model training. Thus, we conclude that in mobility-aware hierarchical FL collaborative control optimization between mobile clients and edge APs holds greater significance than local model training optimization. This conclusion is further corroborated by the training loss presented in Fig. 12(b).

In Fig. 13, we focus on the accuracy-to-round of different mobility optimization frameworks, which emphasizes its superiority by reaching the target accuracy with fewer communication rounds. The results on the dataset SVHN in Fig. 13(a) and the dataset TELECOM in Fig. 13(b), show that comparing with MACFL and FedMobile, ORACLE and ALPHA need fewer communication rounds to reach the corresponding target accuracy, and when the requirement of target accuracy is more strict, the performance gap between four mobility frameworks increases. In Fig. 13(c) and (d), we also find that in long-term mobility scenario, it needs fewer communication rounds to reach

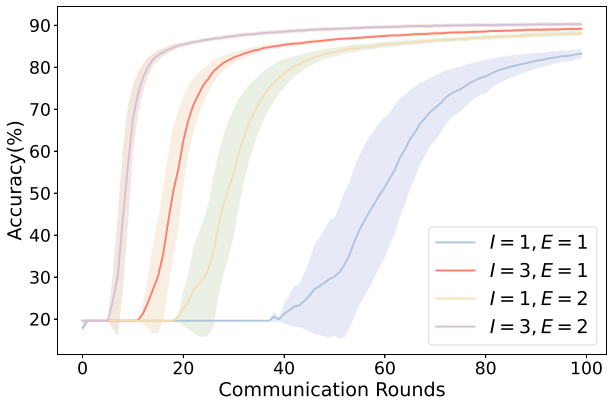

Fig. 14. The performance comparison on the frequency of downward model aggregation and upward model aggregation, i.e., (I, E).

the corresponding target accuracy (80% on TELECOM and 60% on GTRSB) comparing with short-time mobility scenario. In Fig. 14, we explore the model performance on the frequency of downward model aggregation and upward model aggregation, i.e., (I, E). It is easy to find that the model performance of ALPHA will converge faster with the increase of the value of (I, E). Taking the difference between I and E into consideration, we find that the model performance with the setting of $(I = 3, E = 1)$ is better than the one with the setting of $(I = 1, E = 2)$, which means that the downward data distribution divergence in mobility optimization is much vital than the upward one.

VI. Related Works and Discussion

Hierarchical FL in wireless networks have drawn much research attention for the advantage in reducing both communication cost and the number of local iterations [40], [41], [42], [43]. There is an increasing number of HFL-related studies in many areas, e.g., wireless communication networks [44], [45], fault diagnosis [46], industrial IoT [47], intelligent transportation systems [48], human activity recognition [49] and so on. Liu et al. [50] designed a HierFAVG algorithm with partial model aggregation to achieve better communication-computation tradeoff. Similarly, a clustered hierarchical distributed federated learning has been proposed to solve the problem of high consumption of communication resources [51]. Abad et al. [52] proposed a kind of hierarchical FL scheme to reduce the communication latency without sacrificing the accuracy. Although hierarchical FL has been popular with much related research works, the prediction accuracy of hierarchical aggregated model degrades a lot if the data distributions of local clients diverge.

Non-i.i.d data problem have been one of fundamental challenges for model training in HFL where the impact of non-i.i.d data could be exacerbated in HFL [43], [53], [54], [55], [56], [57]. Although a variety of works have been proposed to lessen the problem of non-i.i.d data distribution [58], [59], [60], these methods just deal with a given static data distribution in the single aggregation manner, and is not suitable for mobility scenarios with dynamic non-i.i.d data distributions in hierarchical aggregation manner. Wang et al. [61] provided theoretical results why local aggregation can be beneficial in improving the convergence of Hierarchical SGD. Briggs et al. [62] proposed a hierarchical clustering step to cluster similar clients for specialised models. Similarly, Wu et al. [17] focused on federated user modeling tasks and proposed a novel hierarchical FL framework to work out the problem of inconsistent clients and non-i.i.d data. To cope with imbalanced data distribution in HFL, Liu et al. [18] proposed joint user association and resource allocation algorithm to minimize the learning latency.However, current research works on non-i.i.d data in hierarchical FL are primarily dedicated to the static scenario where clients always stay at the same location and the topology between edge APs and clients are static, which is less practice in real-world applications.

Mobility has been well studied in wireless mobile communication networks for being capable of improving network performance, including but not limited to handover efficiency, resource allocation optimization, and quality-of-service assurance [63], [64], [65], [66]. Some research works have also been focused on the impact of client mobility on collaborative training in hierarchical FL. Yu et al. [15] utilized the mobility patterns and preferences of vehicles to collaboratively learn a global model for high popularity contents prediction. The main limitation of this work is that it ignores the high mobile clients with a decrease number of participants. The mobility attribute of clients have also been utilized with client-to-client communication to design a new asynchronous FL algorithm named FedMobile to improve the learning performance [39]. In D2D communication manner, the model aggregation of FL largely depends on the random of client mobility without performance guarantee. Feng et al. [14] proposed a mobility-aware cluster federated learning (MACFL) algorithm with personalization technique to enhance learning performance by redesigning the access mechanism and model aggregation shceme. Despite considering different non-i.i.d data distribution in mobility-aware HFL, the cluster method has just worked out downward divergence without considering the upward divergence.

Although the most recent work MACFL also focuses on client mobility, it is limited by extra operations to redesign access mechanism and lack of solution for uncertain client mobility. Superior to MACFL, ALPHA controls uncertain client mobility to reshape the data distribution of edge APs from the global perspective, which works out the non-i.i.d data problem in root and also scalable in all FL aggregation algorithm without any operations. Moreover, for the application of ALPHA, we consider that ALPHA in offline scenario is suitable for path planning in fixed trajectory for autonomous vehicles, and ALPHA in online one is advantageous for target search with impossible aerial vehicles.

VII. Conclusion

In this work, we have studied the client mobility in wireless networks, and have proposed a new mobility-aware hierarchical FL framework, namely ALPHA, which considered both the offline and online mobility scenarios. Specifically, we make the convergence analysis of performance degradation induced by the divergence of data distributions, and formulate the mobility-aware client-edge association problem to minimize

the divergence of data distributions. In particular, we first focus on the offline scenarios where all client mobility trajectory are already known, and the efficient solution can be derived with the LP and dependent rounding technique. As for the online one, a fast and simple online algorithm based on the subgradient is proposed to make a quick decision with a bounded regret. Experimental results on three public image datasets and a real-world mobility trajectory dataset show that our proposed ALPHA provides an efficient mobility optimization scheme with superior learning performance with $1.40\times$ – $2.89\times$ convergence speedup comparing to the-state-of-art approaches.

ACKNOWLEDGMENT

The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.

REFERENCES

[1] S. Hu, Q. Li, and B. He, "Communication-efficient generalized neuron matching for federated learning," in *Proc. Int. Conf. Parallel Process.*, 2023, pp. 254–263.

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

[3] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1622–1658, Third Quarter, 2021.

[4] W. Y. B. Lim et al., "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, Third Quarter 2020.

[5] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1/2, pp. 1–210, 2021.

[6] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 46–51, Jun. 2020.

[7] Z. Wang, Y. Zhou, Y. Shi, and W. Zhuang, "Interference management for over-the-air federated learning in multi-cell wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 8, pp. 2361–2377, Aug. 2022.

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

[9] W. Y. B. Lim et al., "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 536–550, Mar. 2022.

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

[11] J. Liu, X. Wei, X. Liu, H. Gao, and Y. Wang, "Group-based hierarchical federated learning: Convergence, group formation, and sampling," in *Proc. Int. Conf. Parallel Process.*, 2023, pp. 264–273.

[12] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1826–1848, May 2020.

[13] Y. Zheng, Y. Du, H. Ling, W. Sheng, and S. Chen, "Evolutionary collaborative human-UAV search for escaped criminals," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 217–231, Apr. 2020.

[14] C. Feng, H. H. Yang, D. Hu, Z. Zhao, T. Q. S. Quek, and G. Min, "Mobility-aware cluster federated learning in hierarchical wireless networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 8441–8458, Oct. 2022.

[15] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, and M. S. Hossain, "Mobility-aware proactive edge caching for connected vehicles using federated learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5341–5351, Aug. 2021.

[16] S. Zhang, Z. Zheng, Q. Li, F. Wu, and G. Chen, "Mobility-aware device sampling for statistical heterogeneity in hierarchical federated learning," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, 2024, pp. 656–667.

[17] J. Wu et al., "Hierarchical personalized federated learning for user modeling," in *Proc. Web Conf.*, 2021, pp. 957–968.

[18] S. Liu, G. Yu, X. Chen, and M. Bennis, "Joint user association and resource allocation for wireless hierarchical federated learning with IID and non-IID data," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 7852–7866, Oct. 2022.

[19] Q. Wu et al., "HiFlash: Communication-efficient hierarchical federated learning with adaptive staleness control and heterogeneity-aware client-edge association," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 5, pp. 1560–1579, May 2023.

[20] M. Piórkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "On clustering phenomenon in mobile partitioned networks," in *Proc. SIGMOBILE Workshop Mobility Models*, 2008, pp. 1–8.

[21] F. Ahmed, J. P. Dickerson, and M. D. Fuge, "Diverse weighted bipartite b-matching," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 35–41.

[22] J. Könemann, J. Toth, and F. Zhou, "On the complexity of nucleolus computation for bipartite b-matching games," *Theor. Comput. Sci.*, vol. 998, 2024, Art. no. 114476.

[23] R. Luus, "Optimization of system reliability by a new nonlinear integer programming procedure," *IEEE Trans. Rel.*, vol. R-24, no. 1, pp. 14–16, Apr. 1975.

[24] M. Balletti, V. Piccialli, and A. M. Sudoso, "Mixed-integer nonlinear programming for state-based non-intrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 13, no. 4, pp. 3301–3314, Jul. 2022.

[25] J. Lee and J. C. Hou, "Modeling steady-state and transient behaviors of user mobility: Formulation, analysis, and application," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2006, pp. 85–96.

[26] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[27] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[28] J. C. Bezdek and R. J. Hathaway, "Convergence of alternating optimization," *Neural Parallel Sci. Computations*, vol. 11, no. 4, pp. 351–368, 2003.

[29] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan, "Dependent rounding in bipartite graphs," in *Proc. 43rd Annu. IEEE Symp. Found. Comput. Sci.*, 2002, pp. 323–332.

[30] A. Bachem, W. Kern, A. Bachem, and W. Kern, *Linear Programming Duality*. Berlin, Germany: Springer, 1992.

[31] X. Li, C. Sun, and Y. Ye, "Simple and fast algorithm for binary integer and online linear programming," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 9412–9421.

[32] Y. Netzer, T. Wang, and A. Coates, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2011, Art. no. 7.

[33] 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.

[34] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. Toronto, Tech. Rep., 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[35] S. Wang, Y. Guo, N. Zhang, P. Yang, A. Zhou, and X. Shen, "Delay-aware microservice coordination in mobile edge computing: A reinforcement learning approach," *IEEE Trans. Mobile Comput.*, vol. 20, no. 3, pp. 939–951, Mar. 2021.

[36] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng, "No fear of heterogeneity: Classifier calibration for federated learning with non-IID data," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 5972–5984.

[37] T. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," 2019, *arXiv: 1909.06335*.

[38] Z. Qu, R. Duan, L. Chen, J. Xu, Z. Lu, and Y. Liu, "Context-aware online client selection for hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 12, pp. 4353–4367, Dec. 2022.

[39] J. Bian and J. Xu, "Mobility improves the convergence of asynchronous federated learning," 2022, *arXiv.2206.04742*.

[40] B. Gong, T. Xing, Z. Liu, W. Xi, and X. Chen, "Towards hierarchical clustered federated learning with model stability on mobile devices," *IEEE Trans. Mobile Comput.*, vol. 23, no. 6, pp. 7148–7164, Jun. 2024.

[41] T. Qi, Y. Zhan, P. Li, and Y. Xia, "Tomtit: Hierarchical federated fine-tuning of giant models based on autonomous synchronization," in *Proc. IEEE Conf. Comput. Commun.*, 2024, pp. 1910–1919.

[42] Y. Guo, F. Liu, T. Zhou, Z. Cai, and N. Xiao, "Privacy vs. efficiency: Achieving both through adaptive hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 4, pp. 1331–1342, Apr. 2023.
[43] H. Liu, J. Lu, X. Wang, C. Wang, R. Jia, and M. Li, "FedUP: Bridging fairness and efficiency in cross-silo federated learning," *IEEE Trans. Serv. Comput.*, vol. 17, no. 6, pp. 3672–3684, Nov./Dec. 2024.
[44] J. Feng, L. Liu, Q. Pei, and K. Li, "Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 11, pp. 2687–2700, Nov. 2022.
[45] Y. Zhuang, Z. Zheng, F. Wu, and G. Chen, "LiteMoE: Customizing on-device LLM serving via proxy submodel tuning," in *Proc. 22nd ACM Conf. Embedded Netw. Sensor Syst.*, 2024, pp. 521–534.
[46] J. Lin, J. Ma, and J. Zhu, "Hierarchical federated learning for power transformer fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 3520611.
[47] T. Zhao, F. Li, and L. He, "DRL-based joint resource allocation and device orchestration for hierarchical federated learning in NOMA-enabled industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 19, no. 6, pp. 7468–7479, Jun. 2023.
[48] X. Wang, W. Liu, H. Lin, J. Hu, K. Kaur, and M. S. Hossain, "AI-empowered trajectory anomaly detection for intelligent transportation systems: A hierarchical federated learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 4, pp. 4631–4640, Apr. 2023.
[49] H. Yu et al., "FedHAR: Semi-supervised online learning for personalized federated human activity recognition," *IEEE Trans. Mobile Comput.*, vol. 22, no. 6, pp. 3318–3332, Jun. 2023.
[50] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–6.
[51] Y. Gou, R. Wang, Z. Li, M. A. Imran, and L. Zhang, "Clustered hierarchical distributed federated learning," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 177–182.
[52] M. S. H. Abad, E. Ozfatura, D. Gündüz, and Ö. Erçetin, "Hierarchical federated learning ACROSS heterogeneous cellular networks," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 8866–8870.
[53] J. Kwon and H. Park, "Efficient low-rank federated learning based on singular value decomposition," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2022, pp. 285–286.
[54] S. Yue, J. Ren, J. Xin, S. Lin, and J. Zhang, "Inexact-ADMM based federated meta-learning for fast and continual edge learning," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2021, pp. 91–100.
[55] X. Zhang, M. Fang, Z. Liu, H. Yang, J. Liu, and Z. Zhu, "NET-FLEET: Achieving linear convergence speedup for fully decentralized federated learning with heterogeneous data," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2022, pp. 71–80.
[56] C. Gong, Z. Zheng, F. Wu, Y. Shao, B. Li, and G. Chen, "To store or not? Online data selection for federated learning with limited storage," in *Proc. ACM Web Conf.*, 2023, pp. 3044–3055.
[57] C. Gong, Z. Zheng, F. Wu, X. Jia, and G. Chen, "Delta: A cloud-assisted data enrichment framework for on-device continual learning," in *Proc. Annu. Int. Conf. Mobile Comput. Netw.*, 2024, pp. 1408–1423.
[58] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143.
[59] J. Kim, G. Kim, and B. Han, "Multi-level branched regularization for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 11058–11073.
[60] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "FedBN: Federated learning on non-IID features via local batch normalization," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–27.
[61] J. Wang, S. Wang, R. Chen, and M. Ji, "Demystifying why local aggregation helps: Convergence analysis of hierarchical SGD," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 8548–8556.
[62] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-IID data," in *Proc. Int. Joint Conf. Neural Netw.*, 2020, pp. 1–9.
[63] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 477–486, Aug. 2002.
[64] Y. Yuan et al., ": Mobility-driven integration of heterogeneous urban cyber-physical systems under disruptive events," *IEEE Trans. Mobile Comput.*, vol. 22, no. 2, pp. 906–922, Feb. 2023.
[65] R. Karmakar, G. Kaddoum, and S. Chattopadhyay, "Mobility management in 5G and beyond: A novel smart handover with adaptive time-to-trigger and hysteresis margin," *IEEE Trans. Mobile Comput.*, vol. 22, no. 10, pp. 5995–6010, Oct. 2023.
[66] F. Zhang, J. Xiong, Z. Chang, J. Ma, and D. Zhang, "Mobi ^{2} Sense: Empowering wireless sensing with mobility," in *Proc. 28th Annu. Int. Conf. Mobile Comput. Netw.*, 2022, pp. 268–281.

Haibo Liu (Graduate Student Member, IEEE) received the BE degree in computer science and technology from Zhejiang Normal University, Jinhua, China. He is currently working toward the PhD degree with the School of Computer Science, Shanghai Jiao Tong University, Shanghai, China. His research interests include federated learning, continual learning, and game theory.

Zhenzhe Zheng (Member, IEEE) received the BE degree in software engineering from Xidian University, in 2012, and the MS and PhD degrees in computer science and engineering from Shanghai Jiao Tong University, in 2015 and 2018, respectively. He is a professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. He has visited the University of Illinois at Urbana-Champaign (UIUC) as a post doc research associate from 2018 to 2019. His research interests include game theory and mechanism design, networking and mobile computing, and online marketplaces. He is a recipient of the China Computer Federation (CCF) Excellent Doctoral Dissertation Award 2018, Google PhD Fellowship 2015 and Microsoft Research Asia PhD Fellowship 2015. He has served as the member of technical program committees of several academic conferences, such as MobiHoc, AAAI, MSN, IoTDI and etc. He is a member of the ACM, and CCF. For more information, please visit https://zhengzhenzhe220.github.io/.

Fan Wu (Member, IEEE) received the BS degree in computer science from Nanjing University, in 2004, and the PhD degree in computer science and engineering from the State University of New York at Buffalo, in 2009. He is a professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. He has visited the University of Illinois at Urbana-Champaign (UIUC) as a post doc research associate. His research interests include wireless networking and mobile computing, data management, algorithmic network economics, and privacy preservation. He has published more than 200 peer-reviewed papers in technical journals and conference proceedings. He is a recipient of the first class prize for Natural Science Award of China Ministry of Education, China National Fund for Distinguished Young Scientists, ACM China Rising Star Award, CCFTencent "Rhinoceros bird" Outstanding Award, and CCF-Intel Young Faculty Researcher Program Award. He has served as an associate editor of *IEEE Transactions on Mobile Computing* and *ACM Transactions on Sensor Networks*, an area editor of *Elsevier Computer Networks*, and as the member of technical program committees of more than 100 academic conferences. For more information, please visit http://www.cs.sjtu.edu.cn/\;fwu/.

Guihai Chen (Fellow, IEEE) received the BS degree from Nanjing University, in 1984, the ME degree from Southeast University, in 1987, and the PhD degree from the University of Hong Kong, in 1997. He is a distinguished professor with Shanghai Jiao Tong University, China. He had been invited as a visiting professor by many universities including the Kyushu Institute of Technology, Japan in 1998, the University of Queensland, Australia in 2000, and Wayne State University, USA during September 2001 to August 2003. He has a wide range of research interests with focus on sensor networks, peer-topeer computing, high-performance computer architecture, and combinatorics. He has published more than 200 peer-reviewed papers, and more than 120 of them are in well-archived international journals such as *IEEE Transactions on Parallel and Distributed Systems*, *Journal of Parallel and Distributed Computing*, *Wireless Networks*, *Computer Journal*, *International Journal of Foundations of Computer Science*, and *Performance Evaluation*, and also in well-known conference proceedings such as HPCA, MOBIHOC, INFOCOM, ICNP, ICPP, IPDPS, and ICDCS.