Mobility-aware Device Sampling for Statistical Heterogeneity in Hierarchical Federated Learning

Songli Zhang, Zhenzhe Zheng, Qinya Li, Fan Wu, and Guihai Chen
Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China 200240
Emails: {zhang_sl, zhengzhenzhe, qinyali}@sjtu.edu.cn, {fwu, gchen}@cs.sjtu.edu.cn

2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS) | 979-8-3503-8605-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDCS60910.2024.00067

***Abstract*—Hierarchical Federated Learning (HFL) is a practical implementation of federated learning in mobile edge computing, employing edge servers as intermediaries between mobile devices and the cloud server for device coordination and cloud communication. However, the devices are usually mobile users with unpredictable mobile trajectories and statistical heterogeneity, leading to the edge models optimized along dynamic edge data distribution directions and further resulting in instability and slow convergence of the global model. In this work, we propose a Mobility-Aware deviCe sampling algorithm in HFL, namely MACH, which can dynamically maintain the device sampling strategy at each edge to accelerate the convergence of the global model. First, we analyze the convergence bound of HFL with mobile devices under arbitrary device sampling probabilities. Based on this convergence bound, we formalize the sampling optimization problem for mobility-aware device sampling, aiming to minimize the convergence error under time-averaged cost constraints, while taking the limited device-edge wireless channel capacity into account. Next, we introduce the MACH algorithm, consisting of two underlying components: experience updating and edge sampling. Experience updating utilizes an upper confidence bound method to estimate device statistical information online, and edge sampling customizes a sampling strategy on each edge based on the estimated device statistical information. Finally, extensive experimental results through real-world mobile device trajectories validate that MACH can reduce the time required to achieve a target accuracy by $25.00\% - 56.86\%$.**

I. Introduction

Hierarchical federated learning (HFL) is a typical implementation of federated learning (FL) in mobile edge computing (MEC) [1]–[3]. Under such a network paradigm, a cluster of edges serves as relays between mobile devices and the cloud server, which can coordinate mobile devices within clusters and communicate with the cloud server [4], [5]. In this way, FL is also implemented with a hierarchical aggregation structure [3], [6]. Edges first aggregate local models from the coordinated mobile devices to form an edge model[1], and the cloud server periodically aggregates these edge models into a global model [7], [8].

However, the statistical heterogeneity of data on mobile devices and then on edges still hinders the convergence of HFL, resulting in instability and slow model convergence progress [9], [10]. The non-independent identical (Non-IID) data distributions across devices create the Non-IID data distribution across edges, causing edge models to be trained in various directions, and potentially deviating from the global optimization directions. To overcome the statistical heterogeneity in traditional server-to-client FL, device sampling is considered as a standard approach [11]–[15]. Device sampling assigns a fixed sampling probability to each device individually, allowing devices that contribute more to global model convergence to participate more in training, which helps reduce the impact of statistical heterogeneity. Some typical device sampling approaches have been demonstrated to be effective in mitigating the statistical heterogeneity in general FL through rigorous theoretical analysis, such as class-balance sampling [14] or gradient-norm based sampling [11], [15]. However, in HFL, mobile devices are geographically distributed, exhibiting natural mobility patterns, introducing time-varying devices coordinated by each edge [16], [17]. Since the edge models are always optimized according to the current data within the edge [18], it further leads to edge models being optimized toward dynamic directions. It makes traditional device sampling strategies fail to apply to HFL with mobile devices, and designing a specific device sampling strategy for device mobility in HFL is necessary.

Developing an appropriate sampling strategy in HFL with mobile devices is non-trivial, and has the following two challenges. The first and fundamental challenge lies in deriving an analytical model convergence bound for HFL with mobile devices for any arbitrary device sampling probabilities. Given the device mobility, each edge coordinates different devices to participate in edge model training during every training round, causing the edge model to be updated along the dynamic optimization direction. Additionally, the edge communicates periodically with the cloud server, and the current updated edge model will serve as the starting point for the next edge training round. Therefore, a comprehensive assessment of the impact of all devices involved in edge model training, from the last global aggregation to the current time, is essential when analyzing the model convergence bound for HFL with mobile devices. Moreover, when designing the sampling strategy for each edge, it is crucial to consider the communication capacity. All of these factors become critical in ensuring effective customization of the sampling process.

The second challenge is determining the optimal device sampling solution based on the above new HFL model convergence bound. Some recent works leverage device training experiences to facilitate device selection [19]–[21]. However,

[1]The term edge model, local model and global model refer to the models on edge, device and cloud, respectively.

2575-8411/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDCS60910.2024.00067

in HFL with mobile devices, devices dynamically participate in the model training processes of different edges, generating various training experiences. Furthermore, the convergence bound in FL heavily relies on certain assumptions concerning the statistical heterogeneity of devices' data [11], [22], such as the upper bounds of local stochastic gradient norms. However, these assumptions introduce unknown parameters that cannot be directly observed before model training. This raises two unsolved issues: 1) whether training experiences from different edges can be shared across edges, and 2) how to leverage these training experiences to accurately estimate the unknown parameters, facilitating the derivation of the device sampling strategy. Addressing the challenge of evaluating unknown parameters in the HFL convergence bound during the training process becomes crucial.

In this work, we address the above two challenges by proposing MACH, which is a Mobility-Aware deviCe sampling algorithm in Hierarchical federated learning, aiming to overcome the notorious statistical heterogeneity in HFL with mobile devices. We first formalize the general scenario of mobile devices participating in HFL, and derive a new HFL convergence bound for non-convex loss functions with arbitrary mobile device sampling probabilities. Our new bound indicates that each edge can independently maintain a specific edge sampling strategy based on the devices within that edge to facilitate the convergence of the global model. Considering the communication constraints among edges in hierarchical wireless networks and the newly derived convergence bound, we tailor a sampling optimization problem, aiming to dynamically adjust the current edge sampling strategy within each edge to minimize the convergence error subject to time-averaged cost constraints. To solve the proposed optimization problem, we introduce an online mobility-aware device sampling algorithm MACH. A key advantage of MACH is that it requires no prior knowledge of device data statistical information, and MACH can customize the edge sampling strategy based on the currently accessible devices within the edge. MACH comprises two components: experience updating and edge sampling. The experience updating maintains a training experience buffer on each device, utilizing the upper confidence bound (UCB) method to estimate device statistical information for edge sampling strategies. On the other hand, edge sampling is employed by each edge to individually customize device sampling probabilities for the devices within that edge to solve the proposed optimization problem.

We summarize our key contributions in this work as follows:

- We investigate the mobility-aware device sampling in HFL, which is the first work to consider device sampling in the context of HFL with mobile devices, and regulate edge model training using device sampling probabilities to address statistical heterogeneity.
- We derive a new HFL convergence bound with arbitrary device sampling probabilities, based on which, we formulate an optimization problem of device sampling to minimize the convergence error of model training.
- We proposed MACH, an online mobility-aware device sampling algorithm. MACH employs the UCB method to perform online experience updating, which relies on no prior knowledge of device data statistics, and independently makes edge sampling strategies for each edge.
- The extensive data-driven simulations with various learning tasks and real-world Telecom datasets demonstrate that MACH can significantly reduce the time required to achieve a target accuracy by $25.00\% - 56.86\%$ compared to other competitive sampling algorithms.

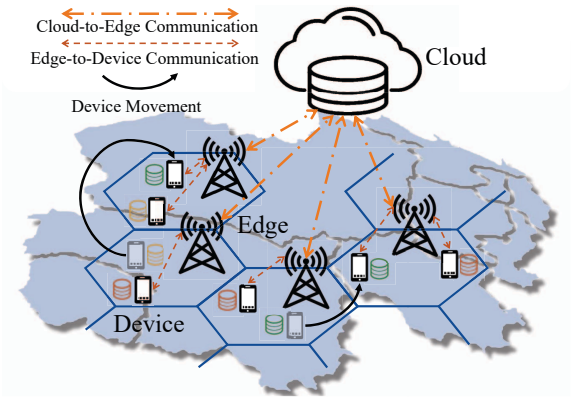


Fig. 1: Hierarchical Wireless Networks with Mobile Devices.

II. Preliminaries

In this section, we first introduce the architecture of hierarchical wireless networks with mobile devices in MEC. Then, we describe the implementation of HFL in such a scenario with arbitrary device sampling probability.

A. Hierarchical Wireless Networks with Mobile Devices

Wireless networks usually introduced edges (*e.g.*, base stations, routers and switches) as relays between the cloud and mobile devices, forming a three-layer device-edge-cloud architecture, as shown in Figure 1. We consider discrete time steps, and mobile devices can move across edges over different time steps[2]. All mobile devices follows a simple clustering scheme based on physical accessibility, *i.e.*, mobile devices tend to select the nearest edge to access according to their geographical locations[3]. The cloud coordinates all edges and mobile devices to satisfy a customized service requirement.

We introduce the important variables and equations used in this work as follows. Let $\mathcal{N}$ be the set of all edges, and $\mathcal{M}$ the set of all devices. In the hierarchical wireless network, $|\mathcal{N}|$ edges and $|\mathcal{M}|$ devices are considered, where $|\cdot|$ represents the cardinality of a set. In each time step $t \in \mathcal{T}$, mobile devices can move across edges while performing local tasks.

Mobile Devices: Each mobile device $m \in \mathcal{M}$ holds a local dataset $\mathcal{D}_m$ of size $|\mathcal{D}_m|$. The devices are geographically

[2]The time steps align with the iterations in FL training process, *i.e.*, time step t is also the basic unit of local model training and all mobile devices can complete local training within a time step.

[3]The mobile device accesses the nearest edge to reduce communication latency and obtain higher quality of service.

distributed and mobile, connecting to different edges in various time steps. To capture this characteristic, we introduce a binary indicator $B_{n,m}^t \in \{0,1\}$ to represent whether device m accesses edge n at time step t. When we have sufficient prior knowledge about device mobility at each time step, obtaining $B_{n,m}^t \in \{0,1\}$ is straightforward. If we are uncertain about device mobility in future time steps and need to make predictions, we can utilize classical mobility models such as Markov mobility model to capture device locations [23], [24]. For instance, we can set a variable $P_{n,m}^t \in [0,1]$ as the probability that device m is accessed to edge n at time step t. Considering the modeling and predicting device trajectories have been extensively studied [25], [26], we consider $B_{n,m}^t \in \{0,1\}$ to be a known quantity [27], [28], and emphasize that our solution is orthogonal to them.

Edges: At each time step t, each edge n can coordinate the mobile devices that access it. Edge n examines all the mobile devices connected to it at the current time step, and let $\mathcal{M}_n^t$ be the set of devices within the edge n at time step t, *i.e.*, $\mathcal{M}_n^t = \left\{m | B_{n,m}^t = 1, \forall m \in \mathcal{M}\right\}$. Considering that each mobile device can only connect to the nearest edge, it has:

$$\mathcal{M}_n^t \cap \mathcal{M}_{n'}^t = \varnothing, \underset{n\in\mathcal{N}}{\cup} \mathcal{M}_n^t = \mathcal{M}, \forall t \in \mathcal{T}, \forall n, n' \in \mathcal{N}. \quad (1)$$

With device mobility, the set of mobile devices $\mathcal{M}_n^t$ associated with edge n changes over time.

B. Mobility-aware HFL based on Arbitrary Sampling

Based on the above hierarchical wireless network in MEC, the HFL is implemented to perform a specific classification learning task to get a global cloud model by solving the following optimization problem:

$$\min_w f(w) = \frac{1}{|\mathcal{M}|} \sum_{m\in\mathcal{M}} F_m(w), \quad (2)$$

which is derived from the general FL algorithm FedAvg [29], and $F_m(\cdot)$ represents the local loss function of device m. The average can also be replaced by a weighted average [30], and we consider a simplified scenario where the number of local dataset is the same across all devices. Then, the cloud, edge $n \in \mathcal{N}$ and device $m \in \mathcal{M}$ iteratively update the global model w^t, edge model w_n^t and local model w_m^t, respectively. The HFL model training is performed over the sequential time steps $\mathcal{T}$, which contains the following main steps:

1) Device Sampling: Due to the resource cost in wireless networks, requiring all devices participating in the FL training is unrealistic [31], [32]. The edge n need to select a subset of devices for training, and each device has an arbitrary probability of being sampled to participate in FL training, denoted as $q_{m,n}^t \in [0,1]$ for device m sampled by edge n at time step t. Let $\mathbb{1}_{m,n}^t \in \{0,1\}$ be an indicator function to denote whether device m is sampled in time step t, and $q_{m,n}^t := Pr\{\mathbb{1}_{m,n}^t = 1\}$. $\mathbb{1}_{m,n}^t$ and $\mathbb{1}_{m',n}^t$ are independent for $m \neq m'$. Due to the channel capacity of the edge, each edge $n \in \mathcal{N}$ expects that only K_n devices can participate in the edge model training in each time step, denoted by:

$$\mathbb{E}\left[\sum_{m\in\mathcal{M}_n^t} \mathbb{1}_{m,n}^t\right] \leq K_n. \quad (3)$$

Finally, the global sampling strategy in each time step t is represented by $\mathcal{Q}^t = \left\{q_{m,n}^t | m \in \mathcal{M}\right\}$.

2) Local Updating: When mobile device $m \in \mathcal{M}$ are sampled to participate the training within the current edge, the device $m \in \mathcal{M}$ first downloads the edge model w_n^t from the accessed edge n at the beginning of time step t. Then, the device m trains the local model based on its local data samples by running I local updates:

$$w_m^{t+1} = w_n^t - \gamma \sum_{\tau=0}^{I-1} g_m\left(w_m^{t,\tau}, \xi_m^{t,\tau}\right), \quad (4)$$

where w_m^t is the local model of the device m at time step t, $w_m^{t,\tau}$ is the interim model during local updating and $w_m^{t,0} = w_n^t$, $\xi_m^{t,\tau}$ is the randomly selected data samples from device m at each local updating, γ is the learning rate, and $g_m(\cdot)$ is the stochastic gradient of $F_m(\cdot)$.

3) Edge Aggregation: The edge aggregate the new edge model w_n^{t+1} for the next time step when receiving the updated local model w_m^{t+1} from all devices:

$$w_n^{t+1} = \sum_{m\in\mathcal{M}_n^t} \frac{1}{|\mathcal{M}_n^t|} \frac{\mathbb{1}_{m,n}^t}{q_{m,n}^t} w_m^{t+1}. \quad (5)$$

Notice that each device's aggregation weight is inversely proportional to its probability of being selected, which ensures the gradient updates remain unbiased. After every T_g time steps, the edge communicates with the cloud server. The device sampling probability $q_{m,n}^t$ in edge n constitutes the edge sampling strategy $\mathcal{Q}_n^t = \left\{q_{m,n}^t | m \in \mathcal{M}_n^t\right\}$ at time step t.

4) Edge-to-Cloud Communication: The cloud server aggregates all uploaded edge models to obtain the global model w^{t+1} in each edge-to-cloud communication time step, *i.e.*, $t \ mod \ T_g = 0$:

$$w^{t+1} = \sum_{n\in\mathcal{N}} \frac{|\mathcal{M}_n^t|}{|\mathcal{M}|} w_n^{t+1}. \quad (6)$$

Then, the cloud distributes the new global model w^{t+1} to all edges and devices. Similar to the classical FL, the cloud server aims to obtain the optimal global model w^* by solving the optimization problem in Eq. (2).

III. Mobility-aware Device Sampling

In this section, we first derive the HFL convergence bound in terms of the mobility patterns of devices and device sampling probabilities, and formulate a new optimization problem, which minimizes the HFL convergence bound by adjusting the device sampling strategy. Then, based on insights inspired by the new proposed convergence bound, we analyze and design MACH, which involves two underlying components: experience updating and edge sampling.

A. Convergence Analysis

We first provide an analysis of the convergence bound on the mobility-aware HFL for arbitrary device sampling probabilities. To ensure a tractable convergence analysis, we stick to the following assumptions:

Assumption 1 *L-smooth: $F_m(w)$ is $L-$Lipschitz smoothness for each device $m \in \mathcal{M}$, i.e., $\|\nabla F_m(w) - \nabla F_m(w')\| \leq L\|w - w'\|^2$ for any two parameter model w and w'.*

Assumption 2 *Unbiased local gradient: The local stochastic gradient on each device $m \in \mathcal{M}$ is unbiased, i.e., $\mathbb{E}_{\xi_m \sim \mathcal{D}_m}[g_m(w, \xi_m)] = \nabla F_m(w)$ for any parameter model w.*

Assumption 3 *Bounded local gradient: The expected squared norm of stochastic gradients on each device $m \in \mathcal{M}$ is bounded, i.e., $\mathbb{E}\|g_m(w, \xi_m)\|^2 \leq G_m^2$ for any parameter model w and randomly selected local data ξ_m.*

Assumptions 1-3 are standard in the classical theoretical analysis of FL algorithms [22], [33], [34]. Assumption 3 sets an upper bound on the gradient norm for each device m. As Assumption 3 holds for any parameter model w and is solely dependent on local data on each device, it reflects the statistical heterogeneity of devices, informing our optimal device sampling design. Furthermore, an additional virtual global model $\overline{w}^{t+1}$ is introduced to represent the aggregation of local models after time step $t+1$:

$$\overline{w}^{t+1} = \sum_{n \in \mathcal{N}} \frac{|\mathcal{M}_n^t|}{|\mathcal{N}|} \sum_{m \in \mathcal{M}_n^t} \frac{1}{|\mathcal{M}_n^t|} \frac{\mathbb{1}_{m,n}^t}{q_{m,n}^t} w_m^{t+1}. \tag{7}$$

$\overline{w}^{t+1}$ is equal to w^{t+1} at the time step when the edge communicates with the cloud server, *i.e.*, $t \ mod \ T_g = 0$.

Lemma 1 (Unbiasedness of Global Gradient Updating). *With the global sampling strategy $\mathcal{Q}^t$, we have:*

$$\mathbb{E}\left[\overline{w}^t | \mathcal{Q}^t\right] = \frac{1}{\mathcal{M}} \sum_{m \in \mathcal{M}} w_m^t. \tag{8}$$

Proof *Since $q_{m,n}^t = Pr\{\mathbb{1}_{m,n}^t = 1\}$, and $\mathbb{1}_{m,n}^t$ are independent $\forall m \in \mathcal{M}$, we can derive Eq.*(8) *by taking the expectation over the virtual global model.* ■

We present the main convergence result on mobility-aware HFL for arbitrary device sampling probability in Theorem 1.

Theorem 1 (Convergence Upper Bound). *Let Assumptions 1-3 hold, for given device sampling strategy $\mathcal{Q}^t$, the HFL with mobile devices satisfies that:*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left\|\nabla f\left(\overline{w}^t\right)\right\|^2 \leq \frac{2\left(f^0 - f^*\right)}{\gamma I T} + \tag{9}$$
$$\sum_{t=0}^{T-1} \frac{\gamma L I\left(2 + \gamma L I\right) + 4(1 + |\mathcal{M}|)T_g^2 L^2 \gamma^2}{2|\mathcal{M}|T} \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}_n^t} \frac{G_m^2}{q_{m,n}^t},$$

where f^* represents the optimal solution to Eq.(2).

Proof *We omit the detailed proof due to page limitation, but a proof sketch can be found in Appendix A.* ■

Remark 1 *This convergence bound characterizes the effect under the arbitrary device sampling probabilities. It shows that the more often devices participate, the less time steps will be required to converge. The device mobility mainly influences the HFL convergence bound by term $\sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}_n^t} \frac{G_m^2}{q_{m,n}^t}$. Each edge can adjust the edge sampling strategy $\mathcal{Q}_n^t$ and minimize $\sum_{m \in \mathcal{M}_n^t} \frac{G_m^2}{q_{m,n}^t}$ to accelerate the convergence.*

However, due to the channel capacity of edges, it is impractical for all mobile devices to access the edge nodes and participate in training simultaneously. Based on Eq. (3), the maximum expected number of accessed devices for each edge $n \in \mathcal{N}$, we can formulate an optimization problem to minimize the convergence bound in Eq. (9) by designing a new sampling strategy, *i.e.*, the mobility-aware device sampling in HFL can be solved through the following problem:

Problem 1

$$\min \quad \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left\|\nabla f\left(\overline{w}^t\right)\right\|^2, \tag{10}$$

$$s.t. \quad \sum_{m \in \mathcal{M}_n^t} q_{m,n}^t \leq K_n, \tag{11}$$

$$q_{m,n}^t \in [0,1], \forall m \in \mathcal{M}, \forall n \in \mathcal{N}. \tag{12}$$

By solving Problem **P1**, we can get the theoretical optimal sampling strategy.

Remark 2 *The optimal sampling strategy at different time steps is independent, and each edge $n \in \mathcal{N}$ maintains a independent optimal sampling strategy. For device $m \in \mathcal{M}_n^t$, without considering the value ranges of $q_{m,n}^t$ (Eq.* (12)*), the optimal device sampling probability $q_{m,n}^{t*}$ follows:*

$$q_{m,n}^{t*} = \frac{K_n G_m^2}{\sum_{m' \in \mathcal{M}_n^t} G_{m'}^2}, \tag{13}$$

which can be easily solved by the method of Lagrange multipliers in closed form. It indicates that the edge sampling strategy $\mathcal{Q}_n^t$ for each edge is solely determined based on the devices within the current edge. The parameter G_m^2 represents the upper bound of the local gradient for each device $m \in \mathcal{M}$, and it is essential to assign higher sampling probabilities to devices with larger gradient norms within each edge.

However, directly observing the local stochastic gradient $\ell 2$-norm G_m^2 of each device $m \in \mathcal{M}_n^t$ is difficult. In the following, we propose MACH, which achieves mobility-aware device sampling in HFL by online estimating device stochastic gradient norms and solving the formulated Problem **P1**.

B. Design of MACH

In this section, we will provide a comprehensive presentation of the principles and design details of MACH. The design of MACH has to address the following two questions: 1) How to evaluate the unknown $\ell 2$-norm of local stochastic gradient

Algorithm 1: MACH

Input: Initial global model w^0, device sampling probability$\{q^t_{m,n}\}$, learning rate γ, local updating epochs I, total training rounds T

Output: Final Global Aggregated Model w^T

1 **for** $t \leftarrow 0, ..., T-1$ **do**
2 **for** *Edge* $n \in \mathcal{N}$ *in parallel* **do**
3 $\mathcal{Q}^t_n \leftarrow$ ***EdgeSampling***$(\{\tilde{G}^2_m | m \in \mathcal{M}^t_n\})$;
4 **for** *Device* $m \in \mathcal{M}^t_n$ *in parallel* **do**
 // Device Sampling
5 Sampled $\mathbb{1}^t_{m,n} \sim q^t_{m,n}, \forall m \in \mathcal{M}^t_n$;
 // Local Updating when $\mathbb{1}^t_{m,n} = 1$
6 $w^{t,0}_m \leftarrow w^t_n$;
7 **for** $\tau \leftarrow 0, ..., I-1$ **do**
8 $w^{t,\tau+1}_m \leftarrow w^{t,\tau}_m - \gamma g_m\left(w^{t,\tau}_m, \xi^{t,\tau}_m\right)$;
9 $w^{t+1}_m \leftarrow w^{t,I}_m$;
10 $\tilde{G}^2_m,\ \mathcal{G}^{t+1}_m \leftarrow$ ***ExperienceUpdating***$\left(\left\{g_m\left(w^{t,\tau}_m, \xi^{t,\tau}_m\right)\right\}, \tilde{G}^2_m\right)$;
 // Edge Aggregation
11 $w^{t+1}_n \leftarrow \sum_{m \in \mathcal{M}^t_n} \frac{1}{|\mathcal{M}^t_n|} \frac{\mathbb{1}^t_{m,n}}{q^t_{m,n}} w^{t+1}_m$;
 // Edge-to-Cloud Communication
12 **if** $t\ mod\ T_g = 0$ **then**
13 $w^{t+1} \leftarrow \sum_{n \in \mathcal{N}} \frac{|\mathcal{M}^t_n|}{|\mathcal{N}|} w^{t+1}_n$;
14 **Return** final global model w^T;

for each device m, *i.e.*, G^2_m, during the training process? 2) How does each edge make a sampling strategy to solve Problem **P1** based on the estimated gradient norm?

Although typical FL sampling approaches have proposed the estimation approaches to address the first question [11], [15], it is unpractical in HFL with mobile devices. Because the mobile devices dynamically participate in the training of different edges, which makes estimating the value of G^2_m difficult. Therefore, mobility-aware device sampling should not only update the device training experience when the mobile device dynamically participates in different edge training processes, but also solve independent edge sampling strategies for edges.

Based on the above principles, the MACH can be achieved by introducing two underlying components: experience updating and edge sampling. Algorithm 1 summarizes the process of MACH. At each time step $t \in \mathcal{T}$, each edge $n \in \mathcal{N}$ performs training in parallel. Firstly, based on the devices in the current edge, each edge n generates the edge sampling strategy $\mathcal{Q}^t_n$ to solve Problem **P1** (Line 3). Then, each device $m \in \mathcal{M}^t_n$ completes device sampling and local updating (Lines 5-9). Further, to obtain the $\ell 2$-norm of the local stochastic gradient for each mobile device m during the FL process, we formulate the online experience updating as a bandit learning problem, and each device m employs a UCB method to get the estimated maximum gradient norm $\tilde{G}^2_m$. Upon receiving all uploaded local models, each edge n aggregates the new edge models

Algorithm 2: Experience Updating

Input: Local gradients $\{g_m\left(w^{t,\tau}_m, \xi^{t,\tau}_m\right)\}$, the estimated maximum gradient norm $\tilde{G}^2_m$

Output: The updated gradient experience buffer $\mathcal{G}^t_m$, the new estimated $\tilde{G}^2_m$

1 $\mathcal{G}^{t+1}_m \leftarrow \mathcal{G}^t_m \cup \{\|g_m\left(w^{t,\tau}_m, \xi^{t,\tau}_m\right)\|^2 | \tau = 0, .., I-1\}$;
2 **if** $t\ mod\ T_g = 0$ **then**
3 $\tilde{G}^2_m \leftarrow \max\left\{\mathbb{1}^{t'}_{m,n} Avg\left(\mathcal{G}^{t'}_m\right) | t' = 0, .., t\right\} + \sqrt{\frac{log(t)}{\sum^t_{t'=0} \mathbb{1}^{t'}_{m,n}}}$;
4 $\mathcal{G}^{t+1}_m \leftarrow \emptyset$;
5 **Return** $\tilde{G}^2_m$, $\mathcal{G}^t_m$;

w^{t+1}_n (Line 11). Finally, the cloud and edges communicate periodically to update the global model w^t (Lines 12-13).

1) Experience Updating: In this part, each mobile device m captures the estimated maximum gradient norm $\tilde{G}^2_m$ during the online FL process.

However, achieving online experience updating is not trivial and involves addressing two key issues. First, in the initial stages of FL training, the cloud server has limited knowledge about the truth value of the expected stochastic gradient norm G^2_m for each mobile device m, and requires a period of training to explore the estimated maximum gradient norm $\tilde{G}^2_m$ for edge sampling decision-making. Simply relying on insufficient experiential exploration will fail to accurately assess the gradient update differences caused by the statistical heterogeneity of devices, and mislead the edge into making suboptimal decisions when making edge sampling decisions. Therefore, balancing the exploration and exploitation of the estimated maximum gradient norm $\tilde{G}^2_m$ in the FL training process is a challenge that needs to be addressed. Second, when mobile devices move across edges and dynamically participate in FL training from different edges, all mobile devices download different edge models from different edges, which can lead to biases in the evaluation of the estimated maximum gradient norm $\tilde{G}^2_m$ at the same time step. Moreover, due to the inherent Non-IID data distribution across devices and the random local training sampling, biases in local data sampling can introduce randomness to the estimated maximum gradient norm $\tilde{G}^2_m$. Particularly, the limited computing resources of mobile devices result in smaller batch sizes for local training, further increasing the randomness of the estimation. As a consequence, directly utilizing training experiences from each time step to evaluate the estimated norm $\tilde{G}^2_m$ is imprecise.

To address these issues, the cloud server employs a classical bandit learning approach for online experience updating, and each device m independently maintains a gradient experience buffer $\mathcal{G}^t_m$, which stores all training experiences between sequential edge-to-cloud communications. By utilizing the mean of these experiences, each mobile device update the estimated maximum gradient norm $\tilde{G}^2_m$ by exploring the UCB score. The procedure of experience updating is summarized

in Algorithm 2. After device m completes its local update, it updates the gradient experience buffer $\mathcal{G}_m^t$ with the training experience based on the gradients from the current training round (Line 1):

$$\mathcal{G}_m^{t+1} = \mathcal{G}_m^t \cup \left\{ \left\| g_m\left(w_m^{t,\tau}, \xi_m^{t,\tau}\right) \right\|^2 | \tau = 0, .., I-1 \right\}. \quad (14)$$

Before the next edge-to-cloud communication begins, the UCB score of the estimated maximum gradient norm $\tilde{G}_m^2$ is calculated as follows (Line 3):

$$\tilde{G}_m^2 = \underbrace{\max\left\{ \mathbb{1}_{m,n}^{t'} Avg\left(\mathcal{G}_m^{t'}\right) | t' = 0, .., t \right\}}_{A} + \underbrace{\sqrt{\frac{log\left(t\right)}{\sum_{t'=0}^{t} \mathbb{1}_{m,n}^{t'}}}}_{B}, \quad (15)$$

where $Avg\left(\cdot\right)$ is a function used to calculate the mean of the gradient experience buffer $\mathcal{G}_m^t$. Terms A and B are commonly denoted as the exploitation and exploration terms, respectively. Exploration term B also represents the corresponding confidence radius of the online estimation. When mobile device m is not sufficiently sampled for exploring and updating the $\tilde{G}_m^2$ value, it leads to an high score for term B, thereby increasing the sampling frequency of mobile device m. Finally, the gradient experience buffer $\mathcal{G}_m^t$ will be cleared (Line 4).

2) Edge Sampling: In this part, based on the HFL convergence bound and Problem **P1**, each edge $n \in \mathcal{N}$ individual generates the current edge sampling strategy.

According to Remark 2, each edge should assign higher sampling probabilities to devices with larger gradient norms. However, when considering the edge channel constraint Eq. (11), directly solving the joint Eq. (11) and (13) may result in some sampling probabilities exceeding their valid range. Moreover, during the initial training stages, the estimated $\tilde{G}_m^2$ may be inaccurate and subject to significant randomness, due to insufficient training. As a consequence, inaccurate estimation of $\tilde{G}_m^2$ may lead to extreme values of the device sampling probability $q_{m,n}^t$, which in turn can result in training failures. When a device with an extremely small sampling probability $q_{m,n}^t \to 0$ is selected, the edge aggregation step can cause an explosive increase in the norm of the parameters of the aggregated edge model in the current training round, leading to gradient vanishing. Therefore, when leveraging the estimated $\tilde{G}_m^2$ to solve Problem **P1**, it is necessary to apply appropriate scaling to the theoretically optimal solution.

The procedure of edge sampling is summarized in Algorithm 3, and each edge n maintains its sampling strategy $\mathcal{Q}_n^t$ through the following steps. Based on the estimated maximum gradient norm $\tilde{G}_m^2$ and Remark 2, each edge can calculate a virtual sampling probability $\hat{q}_{m,n}^t$ for each device m (Line 2):

$$\hat{q}_{m,n}^t = \frac{K_n \tilde{G}_m^2}{\sum_{m' \in \mathcal{M}_n^t} \tilde{G}_{m'}^2}, \quad \forall m \in \mathcal{M}_n^t. \quad (16)$$

We note that it is possible for $\hat{q}_{m,n}^t > 1$. To constrain the range of the actual sampling probability $\hat{q}_{m,n}^t$ and avoid significant variance in all device sampling probabilities among edges,

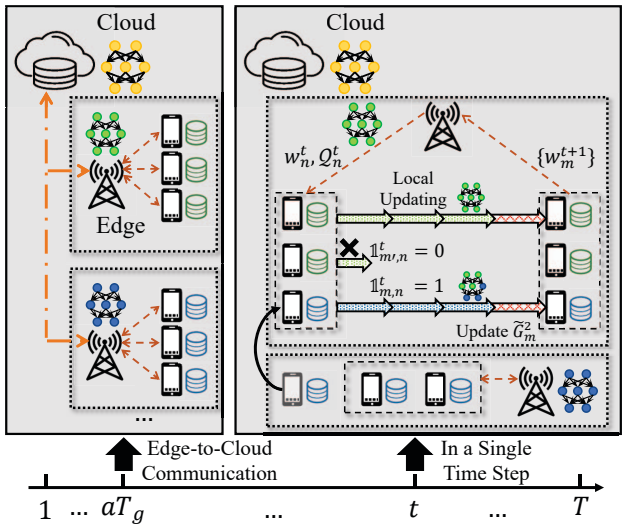


Fig. 2: The Framework of MACH.

we employ an transfer function $S\left(\cdot\right)$ to smooth the sampling probabilities $q_{m,n}^t$ within each edge (Line 3):

$$S\left(\hat{q}_{m,n}^t\right) = 1 + \alpha\left(\frac{1}{1+e^{\beta \hat{q}_{m,n}^t}} - \frac{1}{2}\right), \quad (17)$$

where α and β are task-specific control coefficients, depending on the current neural network architecture and training task. During the early stages of training, α and β should be small to ensure that $\tilde{G}_m^2$ can be adequately estimated through random sampling. By leveraging transfer function $S\left(\cdot\right)$, the values of $S\left(\hat{q}_{m,n}^t\right)$ are constrained to be close to 1.

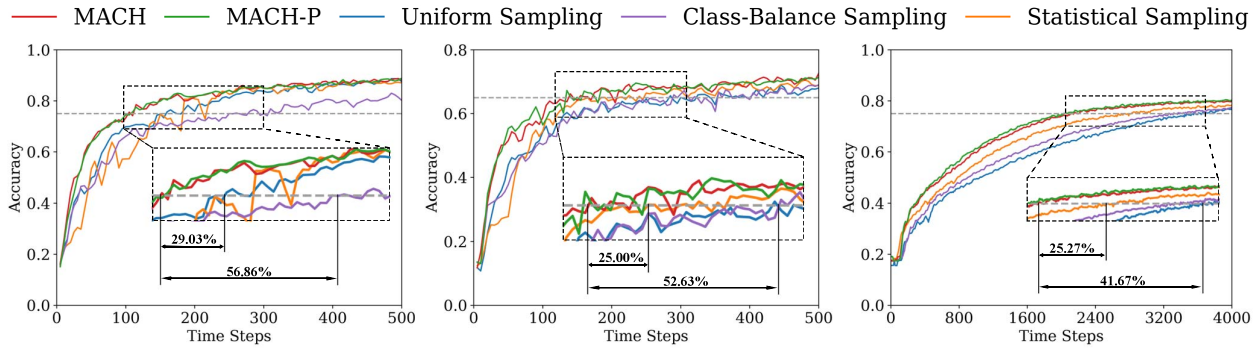
Finally, considering the edge channel constraints in Eq. (11), each edge $n \in \mathcal{N}$ maintains its sampling strategy $\mathcal{Q}_n^t$ at the current time step t as follows (Line 5):

$$q_{m,n}^t = \frac{K_n S\left(\hat{q}_{m,n}^t\right)}{\sum_{m' \in \mathcal{M}_n^t} S\left(\hat{q}_{m',n}^t\right)}, \quad \mathcal{Q}_n^t = \left\{ q_{m,n}^t | m \in \mathcal{M}_n^t \right\}. \quad (18)$$

Based on the above, Figure 2 presents the implementation process of MACH in HFL with mobile devices. In each edge-to-cloud communication, all edge models are aggregated in the cloud and then redistributed to edges and devices. Subsequently, at each time step $t \in \mathcal{T}$, the edge generates an edge sampling strategy $\mathcal{Q}_n^t$ based on the devices within the current edge. The edge model w_n^t and the current edge sampling strategy $\mathcal{Q}_n^t$ are sent to the coordinated devices $m \in \mathcal{M}_n^t$. Each devices m performs local training and updates the estimated $\tilde{G}_m^2$. In this way, the experience updating and edge sampling in MACH alternate to achieve mobility-aware device sampling in HFL with mobile devices.

IV. Evaluation Results

In this section, we evaluate MACH through the real-world Telecom datasets and extensive numerical experiments. We first introduce the experiment settings, and then provide the experimental results with corresponding analysis.



(a) MNIST (b) FMNIST (c) CIFAR10

Fig. 3: Time-to-accuracy performance over all learning tasks.

Algorithm 3: Edge Sampling

Input: The estimated gradient norm $\{\tilde{G}_m^2 | m \in \mathcal{M}_n^t\}$

Output: The current edge sampling strategy $\mathcal{Q}_n^t \leftarrow \{q_{m,n}^t | m \in \mathcal{M}_n^t\}$

for *Device* $m \in \mathcal{M}_n^t$ *in parallel* **do**

  Calculate virtual sampling probability: $\hat{q}_{m,n}^t \leftarrow \frac{K_n \tilde{G}_m^2}{\sum_{m' \in \mathcal{M}_n^t} \tilde{G}_{m'}^2}$;

  Smooth sampling probability: $S\left(\hat{q}_{m,n}^t\right) \leftarrow 1 + \alpha\left(\frac{1}{1+e^{\beta \hat{q}_{m,n}^t}} - \frac{1}{2}\right)$;

for *Device* $m \in \mathcal{M}_n^t$ *in parallel* **do**

  $q_{m,n}^t \leftarrow \frac{K_n S\left(\hat{q}_{m,n}^t\right)}{\sum_{m' \in \mathcal{M}_n^t} S\left(\hat{q}_{m',n}^t\right)}$;

Return $\mathcal{Q}_n^t \leftarrow \{q_{m,n}^t | m \in \mathcal{M}_n^t\}$;

A. Experiment Settings

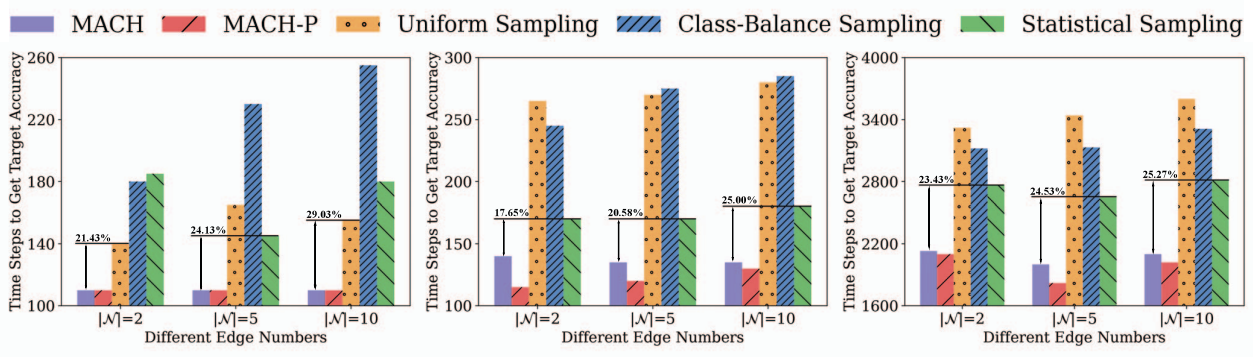
*1) **Dataset**:* We used the Shanghai Telecom dataset to simulate the trajectory of mobile users moving between base stations [35]–[37]. The dataset contains 9,481 mobile devices with over 7.2 million records of dynamic access to 3,233 base stations over 6 consecutive months. Each record in the dataset contains detailed timestamps of when each mobile user started and ended their access to a specific base station. Considering the limited mobile data at some base stations, neighboring base stations cluster together to form several main base stations. The FL training process is performed using three open source datasets, including MNIST, FMNIST and CIFAR10, which are commonly used in image classification tasks and extensively employed to validate FL research work. Each dataset consists of ten image classes.

*2) **Parameter Settings**:* To validate the proposed MACH, we simulate 10 edges and 100 mobile devices. We assume 50% of the devices participating in training at each time step,*i.e.*, the average of all edge channel capacity K_n is 5 in the case of 10 edges. The data distribution of all mobile devices is set to be Non-IID. Both the global and the devices' data distribution follow a long-tailed distribution. The initial state of the edge data distribution is not assumed and is random. The MNIST and FMNIST are trained on the convolutional neural network (CNN) with 2 convolutional layers and 2 fully connected layers with the edge-to-cloud communication interval $T_g = 5$ and an initial learning rate of 0.002 on devices. The CIFAR10 is trained on the convolutional neural network with 3 convolutional layers and 2 fully connected layers with the edge-to-cloud communication interval $T_g = 10$ and an initial learning rate of 0.02 on devices. The local updating epochs I is set as 10. The convergence speed of different algorithms is reflected in the time steps of reaching the target accuracy, which are set as 0.75, 0.65, and 0.75 for MNIST, FMNIST, and CIFAR10, respectively.

*3) **Benchmarks**:* We compare MACH with three other benchmarks. Firstly, we consider three typical and theoretically guaranteed sampling algorithms, uniform sampling [22], class-balance sampling [38] and statistical sampling [14], [39]. Additionally, we assume that the training experiences for each device in every time step are known, *i.e.*, without online experience updating, denoted as MACH-P. We conduct each set of experiments three times and take the average for smoothing. Each edge independently makes sampling strategies based on the devices within the current edge.

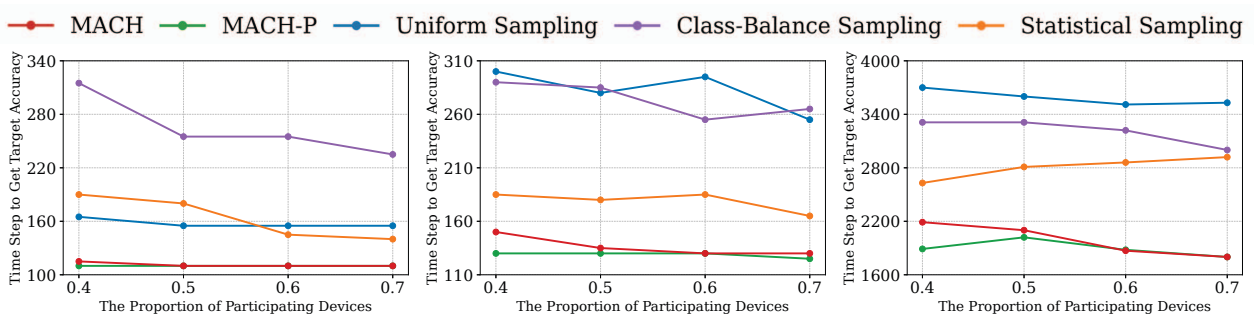
B. Experimental Results and Analysis

1) Overall Performance: First, a set of experiments is conducted to verify the performance of MACH over various learning tasks. In Figure 3, MACH outperforms the basic sampling methods by 25.00% to 56.86% on all learning tasks. By maintaining an edge-specific sampling strategy, MACH enables each edge to better customize its sampling approach based on the current devices within the edge. This allows for more effective adjustments to the optimization direction of the edge model, making it more conducive to global aggregation. In Figures 3(b) and 3(c), the performance of statistical sampling is slightly better than the other two basic sampling methods. This indicates that statistical sampling remains a viable approach to address device statistical heterogeneity in



(a) MNIST (b) FMNIST (c) CIFAR10

Fig. 4: When to get the target accuracy over different number of edges.



(a) MNIST (b) FMNIST (c) CIFAR10

Fig. 5: Time to achieve the target accuracy under different device participation proportions.

HFL. Moreover, the superior performance of MACH over basic sampling methods across all learning tasks highlights the importance of maintaining a unique edge sampling strategy in HFL. Comparing the experimental results of MACH and MACH-P in Figures 3(b) and 3(c), it is evident that MACH-P performs better than MACH in the initial training stages. However, as training progresses, the gap between MACH and MACH-P gradually narrows. This demonstrates the effectiveness of the experience updating step in MACH, which can estimate training experiences during training and use them to adjust edge training strategies. The comparison with all benchmarks demonstrates the significance of the two components, experience updating and edge sampling, in MACH. The experience updating allows for effective iterative updates of training experiences, while the edge sampling effectively addresses the issue of data statistical heterogeneity in HFL.

2) Performance under different edge numbers: We compare the training speeds of MACH on all learning tasks under different edge numbers to validate the necessity of each edge maintaining an edge-specific sampling strategy and the advantages of our proposed MACH. Specifically, we measure the training time cost of achieving the target accuracy as a performance metric for training speed. In Figure 4, we present the experimental results for edge numbers of 2, 5, and 10, while the edge channel capacity is adjusted to ensure approximately 50% device participation in each group of experiments. As the number of edges decreases, the training speeds of all methods seem to accelerate, but the improvement is not significantly evident from direct observation of the experimental results. Only the class-balanced sampling exhibits a noticeable trend in all learning tasks. We specifically mark the training time saved by MACH compared to the best-performing basic sampling method in each group of experiments. Across all training tasks in Figure 4, the improvement of MACH decreases monotonically as the number of edges decreases, *e.g.*, from 29.03% to 21.43% in Figure 4(a). This is because HFL tends to transform towards a simpler server-client two-layer structure with fewer edges, reducing the necessity for edges to maintain edge-specific sampling strategies. Additionally, in Figure 4(a), we notice that the training speeds of MACH and MACH-P are nearly identical. Because MNIST is a relatively simple dataset for handwritten digit recognition, experience updating can effectively capture training experiences for subsequent generating edge sampling strategies.

3) Performance under different device participation proportions: We compare the training performance under different device participation proportions. According to the conclusion of Remark 1, the newly proposed HFL convergence bound indicates that more devices participating in training can effectively accelerate model convergence even in HFL with mobile

TABLE I: The time steps consumed under different local updating epochs I when reaching different accuracies.[4]

Dataset	Target Accuracy	Local Updating Epochs	Time Steps to Get the Target Accuracy				- Time Steps %
			MACH	US	CS	SS	
MNIST	70% Target	$0.8I$	**35**	60	80	65	41.67%
		I	**30**	55	60	50	40.00%
		$1.2I$	**30**	45	55	50	33.33%
	Target	$0.8I$	**110**	160	245	185	31.25%
		I	**110**	155	255	180	29.03%
		$1.2I$	**110**	140	245	170	21.43%
FMNIST	70% Target	$0.8I$	**35**	80	90	100	56.25%
		I	**30**	50	60	65	40.00%
		$1.2I$	**25**	40	55	50	37.50%
	Target	$0.8I$	**140**	320	285	190	26.32%
		I	**135**	280	285	180	25.00%
		$1.2I$	**125**	245	250	165	24.24%
CIFAR10	70% Target	$0.8I$	**710**	1460	1280	1060	33.02%
		I	**670**	1200	1040	880	23.86%
		$1.2I$	**600**	1000	870	720	16.67%
	Target	$0.8I$	**2420**	4220	3870	3250	25.54%
		I	**2100**	3600	3310	2810	25.27%
		$1.2I$	**1800**	3080	2830	2350	23.40%

device participation. To investigate the relationship between the number of participating devices and the convergence speed of the global model in HFL, we adjusted the average edge channel capacity under the setting of 10 edges. As shown in Figure 5, most sampling strategies can effectively reduce the time cost to achieve the target accuracy as the proportion of participating devices increases. However, the results of statistical sampling in Figure 5(c) contradict our intuition, which may be due to the increase in statistical variance caused by the increase in the number of participating devices, hindering the training process. Moreover, in Figure 5, two additional conclusions were verified. MACH consistently outperforms other basic sampling strategies but is slightly inferior to MACH-P. From Figures 5(a) to 5(c), the class-balance sampling shows more significant improvement in reducing training time on more complex datasets. As the proportion of participating devices increases, the performance improvement of MACH compared to baseline sampling gradually diminishes.

4) Performance under different local updating epochs: Finally, we count the time steps consumed by different sampling methods under different local updating epochs I for different learning tasks to reach 70% and 100% target accuracy, as shown in Table I. Based on the experimental results, we have an intuitive observation: for different testing tasks, all sampling methods consume fewer time steps as the local updating epochs I increase, representing the convergence speed increases. We further compared the MACH saved time step percentage compared to the best benchmark in different experiments. As local updating epochs I increase, the saved time step percentage gradually decreases. Because the data distribution of different devices is set to be Non-IID, as local training goes on, each local device has a more biased gradient updating, affecting the online experience updating of MACH and thereby reducing the convergence speed. Furthermore, for the MNIST and FMNIST, MACH's saved time step percentage when reaching the 70% target accuracy is significantly higher than when reaching the final target accuracy. This indicates that in the early stages of training, by maintaining a distinct edge sampling strategy and selecting devices that contribute more to global convergence for training, each edge can more effectively accelerate HFL convergence.

V. Related Work

A. Hierarchical Federated Learning

HFL is widely regarded as a typical implementation of FL in MEC, where the master aggregator dynamically schedules multiple aggregators to scale and update training steps based on the number of devices [1], [34], [40]. From the perspective of model gradient divergence, Wang et al. [34] rigorously analyzed and demonstrated why hierarchical aggregation accelerates the convergence of the global model. Zhong et al. [41] and Wang et al. [7] early proposed improving system efficiency and convergence speed in wireless networks through hierarchical aggregation by leveraging base stations. However, in MEC, clients are often mobile devices capable of randomly moving across different edges. Addressing this characteristic, Feng et al. [42] formulated it as a system reliability problem in HFL, where device mobility may lead to disconnection from the currently associated edge and hinder HFL convergence. Based on device mobility, Fan et al. [43] proposed a device scheduling and resource allocation algorithm for HFL across multiple base stations, aiming to minimize training latency under limited communication resources. Considering the increase in energy consumption by devices for communication due to device mobility, Farcas et al. [44] proposed a dynamic device community selection algorithm in HFL, which can enhance the energy efficiency of the FL system and improve

[4]The term US, CS and SS refer to the uniform sampling, class-balance sampling and statistical sampling, respectively. The best benchmark in each experiment is marked with the underline.

FL training performance. Peng et al. [45] and Chen et al. [46] leveraged the dynamic distribution of data samples within each edge, which results from device cross-edge mobility in HFL, mitigating data heterogeneity to enhance learning performance. However, they have not formally constructed the HFL convergence bound under arbitrary sampling probabilities when mobile devices are involved.

B. Device Sampling

Device sampling is an approach in FL used to handle the statistical heterogeneity of devices [11]–[14]. Most device sampling algorithms are designed based on various optimization objectives, striving to develop unbiased global gradient updating and aggregation algorithms utilizing device sampling probabilities. Luo et al. [11] considered the system wall-clock time to customize the device sampling strategy to enhance FL training efficiency, which leverages the assumption of strong convexity in machine learning. Perazzone et al. [12] addressed the challenge of communication-efficient device sampling while accounting for the associated energy cost of communication. Wang et al. [13] took a comprehensive approach by jointly considering infrequent model transmission, device sampling, and model compression in FL, proposing a flexible control decision algorithm to address this series of interconnected problems. Zhang et al. [38] proposed accelerating the convergence speed by reducing the class imbalance in the selected client groups, where actively chosen clients generate more balanced grouped datasets with theoretical guarantees. Cho et al. [14] proposed an analysis of biased client selection in federated learning convergence and quantifies how this bias affects FL training efficiency.

VI. Conclusion

In this work, we highlight the challenge of device data statistical heterogeneity when implementing FL in MEC, and propose MACH, a mobility-aware device sampling algorithm, to tackle this issue. First, we formalize the general form of HFL with mobile devices under arbitrary sampling probabilities and derive a new HFL convergence bound. Based on the derived convergence bound, we customize an optimization problem for arbitrary device sampling probabilities, aiming to dynamically adjust the current edge sampling strategy to minimize the convergence error under time-averaged cost constraints. Then, to solve the proposed optimization problem, we introduce MACH, which is composed of two underlying components: experience updating and edge sampling. Finally, we validate the effectiveness of MACH through real-world mobile device trajectories and various FL training tasks.

Acknowledgment

This work was supported in part by National Key R&D Program of China (No. 2022ZD0119100), in part by China NSF grant No. 62322206, 62132018, U2268204, 62025204, 62272307, 62372296. 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. Zhenzhe Zheng is the corresponding author.

Appendix A
Proof Sketch of Theorem 1

For the ease of notations, we define $g_m\left(w_m^{t,\tau}\right) := g_m\left(w_m^{t,\tau}, \xi_m^{t,\tau}\right)$ for any w. First, based on Lemma 1, we have:

$$\begin{aligned}
&\mathbb{E}f\left(\overline{w}^{t+1}\right) \\
=&\mathbb{E}f\left(\overline{w}^{t}-\gamma\sum_{n\in\mathcal{N}}\frac{|\mathcal{M}_n^t|}{|\mathcal{M}|}\sum_{m\in\mathcal{M}_n^t}\frac{1}{|\mathcal{M}_n^t|}\frac{\mathbb{1}_{m,n}^t}{q_{m,n}^t}\sum_{\tau=0}^{I-1}g_m\left(w_m^{t,\tau}\right)\right) \\
\overset{(a)}{\leq}&\mathbb{E}f\left(\overline{w}^{t}\right)+\frac{\gamma^2L}{2}\mathbb{E}\left\|\sum_{n\in\mathcal{N}}\frac{|\mathcal{M}_n^t|}{|\mathcal{M}|}\sum_{m\in\mathcal{M}_n^t}\frac{1}{|\mathcal{M}_n^t|}\frac{\mathbb{1}_{m,n}^t}{q_{m,n}^t}\sum_{\tau=0}^{I-1}g_m\left(w_m^{t,\tau}\right)\right\|^2 \\
&-\gamma\mathbb{E}\left\langle\nabla f\left(\overline{w}^{t}\right),\sum_{n\in\mathcal{N}}\frac{|\mathcal{M}_n^t|}{|\mathcal{M}|}\sum_{m\in\mathcal{M}_n^t}\frac{1}{|\mathcal{M}_n^t|}\frac{\mathbb{1}_{m,n}^t}{q_{m,n}^t}\sum_{\tau=0}^{I-1}g_m\left(w_m^{t,\tau}\right)\right\rangle.
\end{aligned} \tag{19}$$

where (a) is a proposition of Assumption 1. For the last term in Eq. (19), we have:

$$\begin{aligned}
&-\gamma\mathbb{E}\left\langle\nabla f\left(\overline{w}^{t}\right),\sum_{n\in\mathcal{N}}\frac{|\mathcal{M}_n^t|}{|\mathcal{M}|}\sum_{m\in\mathcal{M}_n^t}\frac{1}{|\mathcal{M}_n^t|}\frac{\mathbb{1}_{m,n}^t}{q_{m,n}^t}\sum_{\tau=0}^{I-1}g_m\left(w_m^{t,\tau}\right)\right\rangle \\
=&-\gamma\sum_{\tau=0}^{I-1}\mathbb{E}\left\langle\nabla f\left(\overline{w}^{t}\right),\sum_{n\in\mathcal{N}}\frac{|\mathcal{M}_n^t|}{|\mathcal{M}|}\sum_{m\in\mathcal{M}_n^t}\frac{1}{|\mathcal{M}_n^t|}g_m\left(w_m^{t,\tau}\right)\right\rangle \\
=&\gamma\sum_{\tau=0}^{I-1}\mathbb{E}\left\langle\nabla f\left(\overline{w}^{t}\right),\pm\nabla f\left(\overline{w}^{t}\right)-\frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}}g_m\left(w_m^{t,\tau}\right)\right\rangle \\
&-\gamma\sum_{\tau=0}^{I-1}\mathbb{E}\left\|\nabla f\left(\overline{w}^{t}\right)\right\|^2 \\
\overset{(a)}{\leq}&\frac{\gamma L^2}{2|\mathcal{M}|}\sum_{\tau=0}^{I-1}\sum_{m\in\mathcal{M}}\mathbb{E}\left\|\overline{w}^{t}-w_m^{t,\tau}\right\|^2-\frac{\gamma I}{2}\mathbb{E}\left\|\nabla f\left(\overline{w}^{t}\right)\right\|^2.
\end{aligned} \tag{20}$$

where (a) comes from $\langle a,b\rangle \leq \frac{a^2}{2}+\frac{b^2}{2}$, $\nabla f\left(\overline{w}^{t}\right) = \sum_{m\in\mathcal{M}}\frac{1}{|\mathcal{M}|}\nabla F_m\left(\overline{w}^{t}\right)$ and Assumption 1. For the second term in Eq. (19), it has:

$$\begin{aligned}
&\frac{\gamma^2L}{2}\mathbb{E}\left\|\sum_{n\in\mathcal{N}}\frac{|\mathcal{M}_n^t|}{|\mathcal{M}|}\sum_{m\in\mathcal{M}_n^t}\frac{1}{|\mathcal{M}_n^t|}\frac{\mathbb{1}_{m,n}^t}{q_{m,n}^t}\sum_{\tau=0}^{I-1}g_m\left(w_m^{t,\tau}\right)\right\|^2 \\
\leq&\frac{\gamma^2LI^2}{2|\mathcal{M}|}\sum_{n\in\mathcal{N}}\sum_{m\in\mathcal{M}_n^t}\frac{1}{q_{m,n}^t}G_m^2,
\end{aligned} \tag{21}$$

which comes from Jensen's inequality and Assumption 3. By plugging Eq. (21) and (20) into Eq. (19), we can get the rearranged:

$$\begin{aligned}
&\mathbb{E}\left\|\nabla f\left(\overline{w}^{t}\right)\right\|^2 \leq \frac{2\left(\mathbb{E}f\left(\overline{w}^{t}\right)-\mathbb{E}f\left(\overline{w}^{t+1}\right)\right)}{\gamma I}+ \\
&\frac{\gamma LI}{|\mathcal{M}|}\sum_{m\in\mathcal{M}}\frac{G_m^2}{q_{m,n}^t}+\frac{L^2}{|\mathcal{M}|I}\sum_{\tau=0}^{I-1}\sum_{m\in\mathcal{M}}\mathbb{E}\left\|\overline{w}^{t}-w_m^{t,\tau}\right\|^2.
\end{aligned} \tag{22}$$

Then, we will proof $\frac{L^2}{|\mathcal{M}|I}\sum_{\tau=0}^{I-1}\sum_{m\in\mathcal{M}}\mathbb{E}\left\|\overline{w}^t-w_m^{t,\tau}\right\|^2$:

$$\begin{aligned}
&\frac{L^2}{|\mathcal{M}|I}\sum_{\tau=0}^{I-1}\sum_{m\in\mathcal{M}}\mathbb{E}\left\|\overline{w}^t-w_m^{t,\tau}\right\|^2\\
=&\frac{L^2}{I}\sum_{\tau=0}^{I-1}\sum_{n\in\mathcal{M}}\frac{1}{|\mathcal{M}|}\mathbb{E}\left\|\overline{w}^t\pm w_n^t-w_m^{t,\tau}\right\|^2\\
=&L^2\sum_{n\in\mathcal{N}}\frac{|\mathcal{M}_n^t|}{|\mathcal{M}|}\mathbb{E}\left\|\overline{w}^t-w_n^t\right\|^2 \qquad (23)\\
&+\frac{L^2}{I}\sum_{\tau=0}^{I-1}\sum_{n\in\mathcal{N}}\frac{|\mathcal{M}_n^t|}{|\mathcal{M}|}\sum_{m\in\mathcal{M}_n^t}\frac{1}{|\mathcal{M}_n^t|}\mathbb{E}\left\|w_n^t-w_m^{t,\tau}\right\|^2\\
&+\frac{2L^2}{|\mathcal{M}|}\sum_{\tau=0}^{I-1}\sum_{n\in\mathcal{N}}\frac{|\mathcal{M}_n^t|}{|\mathcal{M}|}\sum_{m\in\mathcal{M}_n^t}\frac{1}{|\mathcal{M}_n^t|}\mathbb{E}\left\langle\overline{w}^t-w_n^t,w_n^t-w_m^{t,\tau}\right\rangle.
\end{aligned}$$

For the last term in Eq. (23), it is equal to 0 due to Lemma 1. Then, for the first term in Eq. (23), we have:

$$\begin{aligned}
&L^2\sum_{n\in\mathcal{N}}\frac{|\mathcal{M}_n^t|}{|\mathcal{M}|}\mathbb{E}\left\|\overline{w}^t-w_n^t\right\|^2\\
\leq&\frac{2(1+|\mathcal{M}|)T_g^2L^2\gamma^2}{|\mathcal{M}|}\sum_{n\in\mathcal{N}}\sum_{m\in\mathcal{M}_n^t}\frac{G_m^2}{q_{m,n}^t},
\end{aligned}\qquad (24)$$

which can be demonstrated by $\left\|\sum_{l\in\mathbf{L}}x_l\right\|^2\leq\sum_{l\in\mathbf{L}}L\left\|x_l\right\|^2$, $\frac{|\mathcal{M}_n^t|}{|\mathcal{M}||\mathcal{M}_n^{t'}|}\leq 1\,(\forall t\neq t')$ and Assumption 3. Then, we can prove the upper bound of the second term of (23) by Assumption 3, and:

$$\begin{aligned}
&\frac{L^2}{I}\sum_{\tau=0}^{I-1}\sum_{n\in\mathcal{N}}\frac{|\mathcal{M}_n^t|}{|\mathcal{M}|}\sum_{m\in\mathcal{M}_n^t}\frac{1}{|\mathcal{M}_n^t|}\mathbb{E}\left\|w_n^t-w_m^{t,\tau}\right\|^2\\
\leq&\frac{\gamma^2L^2I^2}{2|\mathcal{M}|}\sum_{n\in\mathcal{N}}\sum_{m\in\mathcal{M}_n^t}\frac{G_m^2}{q_{m,n}^t}.
\end{aligned}\qquad (25)$$

Finally, plugging Eq. (23), Eq. (24) and Eq. (25) into Eq. (22) and taking the average over time, the Theorem 1 can be proved.

REFERENCES

[1] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, B. McMahan *et al.*, "Towards federated learning at scale: System design," *Proceedings of MLSys*, vol. 1, pp. 374–388, 2019.

[2] T. Castiglia, A. Das, and S. Patterson, "Multi-level local sgd: Distributed sgd for heterogeneous hierarchical networks," in *Proceedings of ICLR*, 2021.

[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 & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[4] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2017.

[5] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[6] Y. Li, W. Liang, J. Li, X. Cheng, D. Yu, A. Y. Zomaya, and S. Guo, "Energy-aware, device-to-device assisted federated learning in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, 2023.

[7] Z. Wang, H. Xu, J. Liu, H. Huang, C. Qiao, and Y. Zhao, "Resource-efficient federated learning with hierarchical aggregation in edge computing," in *Proceedings of INFOCOM*, 2021, pp. 1–10.

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

[9] Y. Kang, B. Li, and T. Zeyl, "Fedrl: Improving the performance of federated learning with non-iid data," in *Proceedings of GLOBECOM*, 2022, pp. 3023–3028.

[10] 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 Transactions on Wireless Communications*, vol. 21, no. 10, pp. 7852–7866, 2022.

[11] B. Luo, W. Xiao, S. Wang, J. Huang, and L. Tassiulas, "Tackling system and statistical heterogeneity for federated learning with adaptive client sampling," in *Proceedings of INFOCOM*, 2022, pp. 1739–1748.

[12] J. Perazzone, S. Wang, M. Ji, and K. S. Chan, "Communication-efficient device scheduling for federated learning using stochastic optimization," in *Proceedings of INFOCOM*, 2022, pp. 1449–1458.

[13] S. Wang, J. Perazzone, M. Ji, and K. Chan, "Federated learning with flexible control," in *Proceedings of INFOCOM*, 2023.

[14] Y. J. Cho, J. Wang, and G. Joshi, "Towards understanding biased client selection in federated learning," in *Proceedings of AISTATS*, 2022, pp. 10 351–10 375.

[15] W. Chen, S. Horvath, and P. Richtarik, "Optimal client sampling for federated learning," *Transactions on Machine Learning Research*, pp. 1–32, 2022.

[16] F. Li, J. Zhao, D. Yu, X. Cheng, and W. Lv, "Harnessing context for budget-limited crowdsensing with massive uncertain workers," *IEEE/ACM Transactions on Networking*, vol. 30, no. 5, pp. 2231–2245, 2022.

[17] Y. Ma, W. Liang, J. Li, X. Jia, and S. Guo, "Mobility-aware and delay-sensitive service provisioning in mobile edge-cloud networks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 1, pp. 196–210, 2020.

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

[19] C. Li, X. Zeng, M. Zhang, and Z. Cao, "Pyramidfl: A fine-grained client selection framework for efficient federated learning," in *Proceedings of MobiCom*, 2022, pp. 158–171.

[20] Q. Wu, X. Chen, T. Ouyang, Z. Zhou, X. Zhang, S. Yang, and J. Zhang, "Hiflash: Communication-efficient hierarchical federated learning with adaptive staleness control and heterogeneity-aware client-edge association," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 5, pp. 1560–1579, 2023.

[21] Y. Jin, L. Jiao, Z. Qian, S. Zhang, S. Lu, and X. Wang, "Resource-efficient and convergence-preserving online participant selection in federated learning," in *Proceedings of ICDCS*, 2020, pp. 606–616.

[22] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proceedings of ICLR*, 2019.

[23] T. Higashino, H. Yamaguchi, A. Hiromori, A. Uchiyama, and T. Umedu, "Re-thinking: Design and development of mobility aware applications in smart and connected communities," in *Proceedings of ICDCS*, 2018, pp. 1171–1179.

[24] H. Wang, S. Zeng, Y. Li, and D. Jin, "Predictability and prediction of human mobility based on application-collected location data," *IEEE Transactions on Mobile Computing*, vol. 20, no. 7, pp. 2457–2472, 2020.

[25] Z. Wang, L. Gao, and J. Huang, "Travel with your mobile data plan: A location-flexible data service," in *Proceedings of INFOCOM*, 2020, pp. 1738–1747.

[26] N. Liu, M. Liu, J. Cao, G. Chen, and W. Lou, "When transportation meets communication: V2p over vanets," in *Proceedings of ICDCS*, 2010, pp. 567–576.

[27] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *Proceedings of INFOCOM*, 2015, pp. 2254–2262.

[28] Z. Xu, S. Wang, S. Liu, H. Dai, Q. Xia, W. Liang, and G. Wu, "Learning for exception: Dynamic service caching in 5g-enabled mecs with bursty user demands," in *Proceedings of ICDCS*, 2020, pp. 1079–1089.

[29] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of AISTATS*, 2017, pp. 1273–1282.

[30] S. Wang and M. Ji, "Alightweight method for tackling unknown participation statistics in federated averaging," *arXiv preprint arXiv:2306.03401*, 2024.
[31] W. Luping, W. Wei, and L. Bo, "Cmfl: Mitigating communication overhead for federated learning," in *Proceedings of ICDCS*, 2019, pp. 954–964.
[32] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," in *Proceedings of INFOCOM*, 2021, pp. 1–10.
[33] H. Yu, R. Jin, and S. Yang, "On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization," in *Proceedings of ICML*, 2019, pp. 7184–7193.
[34] J. Wang, S. Wang, R.-R. Chen, and M. Ji, "Demystifying why local aggregation helps: Convergence analysis of hierarchical sgd," in *Proceedings of AAAI*, 2022, pp. 8548–8556.
[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 Transactions on Mobile Computing*, vol. 20, no. 3, pp. 939–951, 2019.
[36] Y. Li, A. Zhou, X. Ma, and S. Wang, "Profit-aware edge server placement," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 55–67, 2021.
[37] Y. Guo, S. Wang, A. Zhou, J. Xu, J. Yuan, and C.-H. Hsu, "User allocation-aware edge cloud placement in mobile edge computing," *Software: Practice and Experience*, vol. 50, no. 5, pp. 489–502, 2020.
[38] J. Zhang, A. Li, M. Tang, J. Sun, X. Chen, F. Zhang, C. Chen, Y. Chen, and H. Li, "Fed-cbs: A heterogeneity-aware client sampling mechanism for federated learning via class-imbalance reduction," in *Proceedings of ICML*, 2023, pp. 41 354–41 381.
[39] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient federated learning via guided participant selection," in *Proceedings of OSDI*, 2021, pp. 19–35.
[40] Z. Jiang, W. Wang, B. Li, and Q. Yang, "Towards efficient synchronous federated training: A survey on system optimization strategies," *IEEE Transactions on Big Data*, vol. 9, no. 2, pp. 437–454, 2022.
[41] Z. Zhong, Y. Zhou, D. Wu, X. Chen, M. Chen, C. Li, and Q. Z. Sheng, "P-fedavg: Parallelizing federated learning with theoretical guarantees," in *Proceedings of INFOCOM*, 2021, pp. 1–10.
[42] C. Feng, H. H. Yang, D. Hu, Z. Zhao, T. Q. Quek, and G. Min, "Mobility-aware cluster federated learning in hierarchical wireless networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 10, pp. 8441–8458, 2022.
[43] K. Fan, W. Chen, J. Li, X. Deng, X. Han, and M. Ding, "Mobility-aware joint user scheduling and resource allocation for low latency federated learning," *arXiv preprint arXiv:2307.09263*, 2023.
[44] A.-J. Farcas, M. Lee, R. R. Kompella, H. Latapie, G. De Veciana, and R. Marculescu, "Mohawk: Mobility and heterogeneity-aware dynamic community selection for hierarchical federated learning," in *Proceedings of IoTDI*, 2023, pp. 249–261.
[45] Y. Peng, X. Tang, Y. Zhou, Y. Hou, J. Li, Y. Qi, L. Liu, and H. Lin, "How to tame mobility in federated learning over mobile networks?" *IEEE Transactions on Wireless Communications*, vol. 22, no. 12, pp. 9640–9657, 2023.
[46] T. Chen, J. Yan, Y. Sun, S. Zhou, D. Gunduz, and Z. Niu, "Data-heterogeneous hierarchical federated learning with mobility," *arXiv preprint arXiv:2306.10692*, 2023.