Mobility Accelerates Learning: Convergence Analysis on Hierarchical Federated Learning in Vehicular Networks

Tan Chen, *Student Member, IEEE*, Jintao Yan, *Student Member, IEEE*, Yuxuan Sun, *Member, IEEE*, Sheng Zhou, *Senior Member, IEEE*, Deniz Gündüz, *Fellow, IEEE*, Zhisheng Niu, *Fellow, IEEE*

***Abstract*—Hierarchical federated learning (HFL) enables distributed training of models across multiple devices with the help of several edge servers and a cloud edge server in a privacy-preserving manner. In this paper, we consider HFL with highly mobile devices, mainly targeting at vehicular networks. Through convergence analysis, we show that mobility influences the convergence speed by both fusing the edge data and shuffling the edge models. While mobility is usually considered as a challenge from the perspective of communication, we prove that it increases the convergence speed of HFL with edge-level heterogeneous data, since more diverse data can be incorporated. Furthermore, we demonstrate that a higher speed leads to faster convergence, since it accelerates the fusion of data. Simulation results show that mobility increases the model accuracy of HFL by up to 15.1% when training a convolutional neural network on the CIFAR-10 dataset.**

***Index Terms*—Hierarchical federated learning, mobility, convergence analysis, heterogeneous data**

I. Introduction

The advent of 5G has revolutionized intelligent vehicles, enabling them to generate and share vast amounts of data through vehicle-to-everything (V2X) services [2]. In this context, machine learning (ML) has become an essential tool to efficiently exploit patterns of data while adapting to the dynamics of the mobile environment [3]–[5]. Conventional ML solutions rely on offloading the data collected by mobile devices to the cloud server. Such centralized solutions, however, encounter challenges of limited communication resources and privacy concerns in vehicular networks [6]–[8]. Federated learning (FL), where multiple devices cooperatively train a model by updating the model parameters with their local data and sharing the gradients through a parameter server, emerges as a privacy-preserving alternative [2], [9]. FL has great potential in various vehicular network scenarios, such as edge caching [10], resource allocation [11], radio-based localization [12], and augmented reality [13].

Nevertheless, FL faces challenges when applied to vehicular networks. Traditionally, the parameter server of FL is located on a remote cloud. However, the communication links between the cloud server and vehicles lead to large communication delays and dropout rates [14], resulting in a long training time [15]. Training tasks such as edge caching and channel estimation are required to be trained up-to-date based on the changing data generated by the dynamic environment, and the timeliness of these tasks can not be guaranteed by the cloud-based FL. Federated edge learning (FEEL), where an edge server acts as the parameter server, has a lower communication delay to vehicles [16], [17]. However, the edge server generally covers a small area, and the density of vehicles is low. As a result, the number of participating vehicles is often too small to make the model converge [2]. To balance the training delay and coverage range, hierarchical federated learning (HFL) has been proposed [18], [19], which trains a global model at a central cloud server in a federated manner with the help of edge servers. The wide coverage of the cloud server ensures sufficient participating vehicles, while the edge servers are mediators for model aggregation, which help decrease the total training time. This three-layer structure has shown its potential in balancing the volume of training data and latency in mission cognitive wireless emergency networks [20].

In this work, we consider HFL in vehicular networks (see Fig. 1 for an illustration). The performance of HFL in vehicular networks is mainly affected by two factors. Firstly, different vehicles have diverse routes and may collect data with diverse statistics, e.g., different classes, resulting in data heterogeneity (or, non-i.i.d. data) [9], [21]. Heterogeneous data causes the local objective function to diverge from the global objective function, thereby degrading the learning performance [22], [23]. Many research efforts have been devoted to the problem of heterogeneous data in FL and HFL, but few in the context of vehicular networks. Secondly, unlike traditional HFL where the devices are fixed, the mobility of vehicles dynamically changes the network topology. Some studies have considered the impact of mobility on scenarios in vehicular networks, such as edge attacks [24], and opportunistic communication between vehicles the edge server for collaborative

This work was presented in part at the 35th International Teletraffic Congress (ITC-35) [1].

This work is sponsored in part by the Natural Science Foundation of China (No. 62341108, No. 62301024, No. 62221001), by the Beijing Natural Science Foundation under Grant L222044, by the Talent Fund of Beijing Jiaotong University under grant 2023XKRC030, and by the project of Tsinghua University-Toyota Joint Research Center for AI Technology of Automated Vehicle (No. TTAD2023-08).

T. Chen, J. Yan, S. Zhou, and Z. Niu are with Department of Electronic Engineering, Tsinghua University, Beijing 100084, China, and the Beijing National Research Center for Information Science and Technology (e-mail: {chent21,yanjt22}@mails.tsinghua.edu.cn, {sheng.zhou,niuzhs}@tsinghua.edu.cn).

Y. Sun (Corresponding Author) is with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: yxsun@bjtu.edu.cn).

D. Gündüz is with the Department of Electrical and Electronic Engineering, Imperial College London, SW7 2BT, UK (e-mail: d.gunduz@imperial.ac.uk).

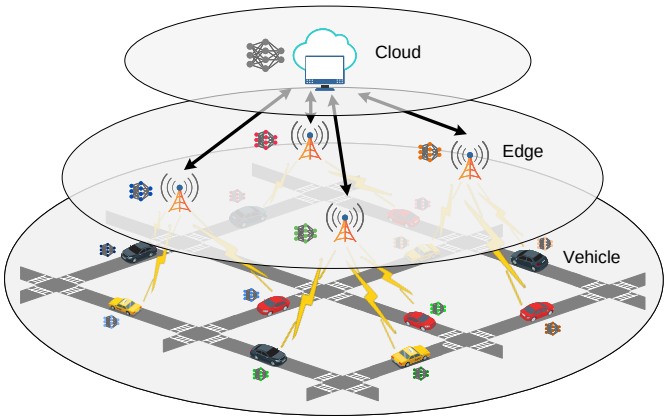


Fig. 1. Illustration of HFL in vehicular networks.

learning [25]. The impact of mobility in HFL is examined in [26], but the authors claim that mobility deteriorates the training performance by decreasing the number of vehicles that successfully upload their models.

We have a different perspective on the impact of mobility on HFL. On the one hand, the mobility of vehicles cause frequent cell handovers and channel variations, which makes the uploading of models from the vehicles to the edge servers more difficult. Vehicles may frequently receive the edge model from one edge server, while uploading its local model to another. This phenomenon is called *shuffling the edge models*. On the other hand, mobility promotes mixing of heterogeneous data, because vehicles can act as 'data mules' [27], which potentially improve the learning performance. This is called *fusing the edge data*.

Therefore, in this work, we investigate the effect of mobility on the performance of HFL. The main contributions of our work can be summarized as follows.

- We analyze the upper bound of the loss function of general HFL tasks *w.r.t.* heterogeneous data and general mobility models, showing how mobility influences the bound.
- We further analyze the performance of HFL for classification tasks with Markov mobility of vehicles. Three typical initial label distributions are discussed and theoretical results demonstrate that mobility *improves* the performance of HFL with edge-level label imbalance. Specially, when the edge servers form a ring topology, a higher vehicle speed leads to faster convergence.
- Through simulations we show that mobility indeed *enhances* the convergence speed and accuracy of HFL with edge-level label imbalance. Furthermore, the potential of mobility to reduce resource consumption is studied.

The rest of this paper is organized as follows. Section II introduces the related work. Section III describes the system model, characterizes the learning task, and presents the training algorithm. In Section IV, convergence analysis of the system is conducted, showing the impact of mobility on HFL with heterogeneous data. Section V presents simulation results and discussions. Finally, Section VI concludes the work.

Throughout the paper, we use $\mathbb{E}[\cdot]$ to represent the expectation operation, and $\mathbb{P}[\cdot]$ to represent the probability of an event. $\|\cdot\|$ maps the l2-norm of a vector, while $\|\cdot\|_1$ maps the l1-norm of a vector. The cardinality of a set is denoted by $|\cdot|$.

II. Related Work

The issue of heterogeneous data has gained attention ever since the inception of FL. In [9], it is highlighted that non-i.i.d. data across devices is a natural occurrence in FL. To date, numerous efforts have been made to address this problem, which can be broadly categorized into three types: data-based, algorithm-based, and framework-based [28], [29]. Data-based methods aim to balance data among different devices by sharing a small portion of global data [30] or generating augmented samples [31]. Algorithm-based methods focus on designing specific aggregation [32], [33] or optimization strategies [34]. Framework-based methods employ new paradigms to model the data heterogeneity of devices, such as multi-task learning [35], personalized learning [36], clustering [37], [38], and split learning [39]. Nonetheless, these studies treat data heterogeneity as a generic phenomenon [40], and most of them examine the performance of their algorithms by simulations without much analysis on the convergence bound of FL.

The authors of [41] classify data heterogeneity into four specific types: feature distribution skew, label distribution skew, concept shift between features and labels, and quantity skew. Among them, label distribution skew, also known as *class imbalance*, is more prevalent in realistic scenarios. Consequently, most subsequent researches focus on this aspect.

Simulations have demonstrated that imbalanced data negatively impacts the training performance [23]. A mitigation strategy is then proposed, which measures the class imbalance by the ratio between the number of samples in the majority class and the minority class, and adds the ratio loss to the training loss. In [42], devices are categorized into uniform and biased groups based on their data distributions. Mediators are then introduced to manage the training and balance biased devices according to the Kullback-Leibler (KL) divergence of their data distributions. The authors of [40] adjust the cross-entropy loss based on the occurrence of each class to train a more balanced local model. The final classifier layer of the neural network, typically a softmax layer, is modified in [42], [43]. A scaling factor is introduced into the softmax function to accommodate missing classes in [42], while the i.i.d. raw features are employed to calibrate the trained model [43]. The accuracy decrease in FL with non-i.i.d. data is attributed to weight divergence in [30], where earth mover's distance (EMD) is introduced as a measure to quantitatively assess weight divergence. Although the class imbalance problem is effectively addressed in these work, vehicular network scenarios are not considered, where the mobility of vehicles impact the training performance.

In the context of vehicular networks, to balance the training time and the number of participating vehicles, HFL structure is considered, where vehicles cooperatively train a global model with the help of the cloud and edge servers. Several work attempt to reduce the degree of heterogeneity of HFL by

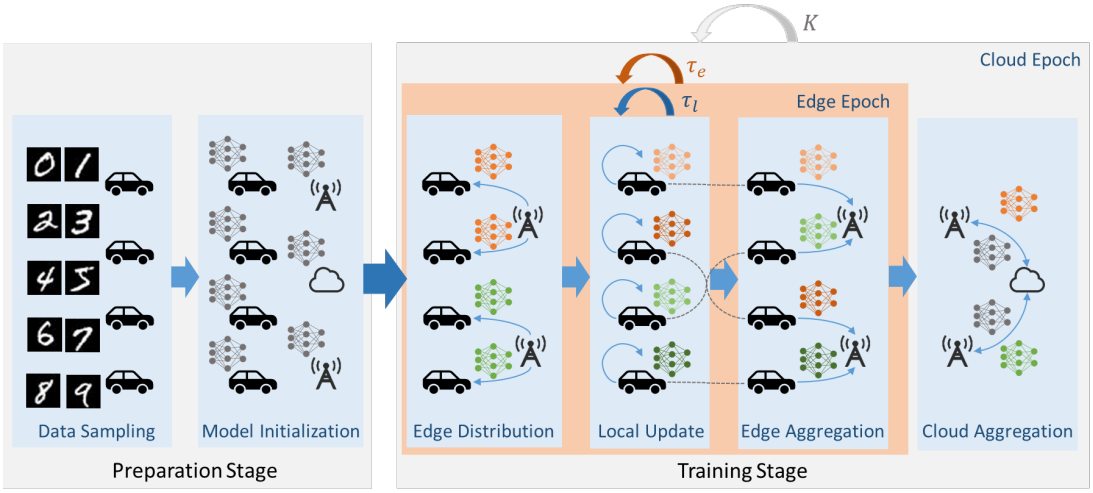


Fig. 2. The training procedure of HFL.

shaping the data distribution at the edge. The user-edge association problem is often considered, which involves allocating users that fall under the coverage of multiple edge servers. By minimizing the KL divergence among the data distribution of each edge server, a user-edge association method is adopted in [44], reducing the total number of communication rounds during training. The authors of [45] model the optimal user-edge association problem by minimizing the weight divergence between the global and optimal models, which can be further converted to data distribution imbalance. As a result, the assignment problem is transformed into a distributed optimization problem. Despite the value of these work, they do not consider the change in data distribution when vehicles move.

The impact of mobility in HFL is examined in [26]. The convergence speed of HFL in relation to mobility rate is analyzed, showing that mobility decreases the convergence speed of HFL, and an algorithm is then proposed to mitigate the impact of mobility by adopting cosine similarity as the aggregating weights. However, the authors ignore vehicles that cross the coverage of edge servers (crossing vehicles) when edge servers aggregate models of vehicles. In contrast, we think the data on the crossing vehicles is important as they help mix diverse data, so these vehicles should also be considered.

Theoretical analysis of FL with heterogeneous data has been conducted by several work. The authors of [46], [47] measure non-i.i.d. by the upper bound of gradient divergence and the difference between the optimal values of the local and global loss functions, respectively. The corresponding convergence bounds with respect to the non-i.i.d. metrics are then derived. Subsequent work [18], [26], [48]–[51] further analyze the convergence bound for HFL with heterogeneous data. These work does not consider the scenario of vehicular networks, so the impact of mobility is not examined.

The authors of [3] suggest that mobility has two-fold impacts on data-heterogeneous FL. On the one hand, it brings uncertainties to the network topology, which may decrease the number of vehicles that successfully upload the model or deteriorate the quality of the model. On the other hand, it facilitates data fusion which potentially accelerates the convergence speed of training. Simulations show that mobility increases the test accuracy of the edge-based FL.

In our previous work [1], the upper bound of the loss function of HFL with respect to mobility and data heterogeneity is derived, which shows the potential of mobility to improve the performance, but the gain of mobility is not analyzed. In this work, we show the two-fold effects of mobility by a more comprehensive convergence analysis and exploit the gain of mobility on HFL by constructing a mobility model and plugging it into the bound. Further simulations show the potential of mobility to reduce training costs.

III. HFL in Vehicular Networks

We consider HFL in vehicular networks. A central cloud server equipped with a macro base station controls several edge servers, which can represent base stations or roadside units. Each edge server is static, and covers a limited area of streets. The vehicles move on the streets and cross the coverage of an edge server occasionally. We assume that the cloud server has a large enough coverage, so all the vehicles are always within its coverage. Therefore, the cloud and edge servers, and the vehicles together form a closed system without input and output flows. We denote the central cloud as C, and assume that there are N edge servers and M mobile vehicles. In practice, M is much larger than N. The vehicles in the system cooperatively train a model with the help of the cloud and edge servers.

The HFL task attempts to find the connection between inputs $\boldsymbol{x}_i$ and labels y_i in the global dataset $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{|\mathcal{D}|}$. Let $\mathcal{D}_m$ denote the dataset of the m-th vehicle; we have $\mathcal{D} = \cup_{m=1}^{M} \mathcal{D}_m$.

For a sample $\{\boldsymbol{x}_i, y_i\}$, let $g_i(\boldsymbol{w})$ be the sample loss function for models $\boldsymbol{w}$. The loss function of the m-th vehicle is given by $f_m(\boldsymbol{w}) = \frac{1}{|\mathcal{D}_m|} \sum_{i \in \mathcal{D}_m} g_i(\boldsymbol{w})$, and the global loss function is then determined by an average of the sample loss functions, $F(\boldsymbol{w}) = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} g_i(\boldsymbol{w}) = \sum_{m=1}^{M} \frac{|\mathcal{D}_m|}{|\mathcal{D}|} f_m(\boldsymbol{w})$. The training objective of HFL is $\min_{\boldsymbol{w}} F(\boldsymbol{w})$.

Algorithm 1 Mobility-aware Hierarchical Federated Averaging (Mob-HierFAVG)

Require: η

procedure MOB-HIERFAVG
  Initialize the cloud model with parameter $\boldsymbol{w}^{(0)}$
  for $k = 0, \ldots, K$ **do**
    for $t_e = 0, \ldots, \tau_e$ **do**
      EdgeDistribute$\left(\{\boldsymbol{w}_{e,n}^{(\tau)}\}_{n=1}^{N}, \{\boldsymbol{w}_{m}^{(\tau)}\}_{m=1}^{M}\right)$
      for $t_l = 1, \ldots, \tau_l$ **do**
        $\tau \leftarrow \tau + 1$
        LocalUpdate$\left(\{\boldsymbol{w}_{m}^{(\tau)}\}_{m=1}^{M}\right)$
      end for
      for all edge server $n = 1, \ldots, N$ in parallel **do**
        $\mathcal{E}_n^{(\tau)} \leftarrow \mathcal{E}_n^{(\tau-1)}$
        $\boldsymbol{w}_{e,n}^{(\tau)} \leftarrow \sum_{m \in \mathcal{E}_n^{(\tau)}} \alpha_{m,n} \boldsymbol{w}_m^{(\tau)}$
      end for ▷ EdgeAggregation
    end for
    $\boldsymbol{w}^{(\tau)} \leftarrow \sum_n \theta_n \boldsymbol{w}_{e,n}^{(\tau)}$
    $\{\boldsymbol{w}_{e,n}^{(\tau)}\}_{n=1}^{N} \leftarrow \boldsymbol{w}^{(\tau)}$ ▷ CloudAggregation
  end for
end procedure
function EDGEDISTRIBUTE($\{\boldsymbol{w}_{e,n}^{(\tau)}\}_{n=1}^{N}, \{\boldsymbol{w}_{m}^{(\tau)}\}_{m=1}^{M}$)
  for all edge server $n = 1, \ldots, N$ in parallel **do**
    for all vehicle $m \in \mathcal{E}_n^{(\tau)}$ in parallel **do**
      $\boldsymbol{w}_m^{(\tau)} \leftarrow \boldsymbol{w}_{e,n}^{(\tau)}$
    end for
  end for
end function
function LOCALUPDATE($\{\boldsymbol{w}_{m}^{(\tau)}\}_{m=1}^{M}$)
  for all client $m = 1, \ldots, M$ in parallel **do**
    $\boldsymbol{w}_m^{(\tau)} \leftarrow \boldsymbol{w}_m^{(\tau-1)} - g\left(\boldsymbol{w}_m^{(\tau-1)}\right)$
  end for
end function

A. Mobility-aware Hierarchical Federated Averaging

We follow the HFL framework in [18], [19] while further incorporating the mobility of vehicles. We call the stochastic gradient descent (SGD) process on a batch of samples at each vehicle as a local update. We maintain an iteration enumerator τ, which records the number of local updates each vehicle has carried out in total. A synchronized system is considered, so τ is tracked by all the vehicles. We denote the models of the m-th vehicle, the n-th edge server, and the cloud server at iteration τ as $\boldsymbol{w}_m^{(\tau)}, \boldsymbol{w}_{e,n}^{(\tau)}, \boldsymbol{w}^{(\tau)}$, respectively.

The training procedure is illustrated in Fig. 2. We assume that vehicles sample data before training in the preparation stage, and the dataset carried by each vehicle does not vary during training. After sampling, all the cloud and edge servers initialize with a global model $\boldsymbol{w}^{(0)}$. The training stage is consist of the following four steps:

1) Edge distribution: Let each edge server maintain a covering vehicle set $\mathcal{E}_n^{(\tau)}$, and assume that each vehicle is associated with the nearest edge server only. Each edge server distributes the edge model to all the vehicles within its coverage, i.e., $\boldsymbol{w}_m^{(\tau)} = \boldsymbol{w}_{e,n}^{(\tau)}, m \in \mathcal{E}_n^{(\tau)}$.

TABLE I
SUMMARY OF MAIN NOTATIONS

Notation	Definition
$\mathcal{D}$; $\mathcal{D}_m$	global dataset; dataset of the m-th vehicle
$F(\boldsymbol{w})$; $f_m(\boldsymbol{w})$	global loss function; loss function of the m-th vehicle
τ	number of total local updates
$\boldsymbol{w}^{(\tau)}$; $\boldsymbol{w}_{e,n}^{(\tau)}$; $\boldsymbol{w}_m^{(\tau)}$	model at the cloud server, n-th edge server and m-th vehicle
$\boldsymbol{\xi}_m^{(\tau)}$	the batch sampled by the m-th vehicle
$g\left(\boldsymbol{w}_m^{(\tau)}\right)$	gradients calculated over the batch $\boldsymbol{\xi}_m^{(\tau)}$
τ_l; τ_e; K	local period; edge period; cloud period
$\theta_n^{(\tau)}$; $\alpha_{m,n}^{(\tau)}$	aggregation weights of the n-th edge server and m-th vehicle

2) Local update: For each local update, the m-th vehicle performs SGD using its local data by $\tilde{\boldsymbol{w}}_m^{(\tau)} = \boldsymbol{w}_m^{(\tau-1)} - \eta \nabla f_m\left(\boldsymbol{w}_m^{(\tau-1)}, \boldsymbol{\xi}_m^{(\tau-1)}\right)$, where η is the learning rate, and $\nabla f_m\left(\boldsymbol{w}_m^{(\tau)}, \boldsymbol{\xi}_m^{(\tau)}\right)$ is the stochastic gradient of the loss function with data batch $\boldsymbol{\xi}_m^{(\tau)}$ sampled from $\mathcal{D}_m$. For notation simplicity, we define $g\left(\boldsymbol{w}_m^{(\tau)}\right) \triangleq \eta \nabla f_m\left(\boldsymbol{w}_m^{(\tau)}, \boldsymbol{\xi}_m^{(\tau)}\right)$. Local update is repeated τ_l times until the next edge aggregation starts.

3) Edge aggregation: Each edge server first updates the covering vehicle set $\mathcal{E}_n^{(\tau)}$. Then it collects and aggregates models from all the vehicles within its coverage at that time. The aggregated model at the n-th edge server is denoted by $\tilde{\boldsymbol{w}}_{e,n}^{(\tau)} = \sum_{m \in \mathcal{E}_n^{(\tau)}} \alpha_{m,n}^{(\tau)} \tilde{\boldsymbol{w}}_m^{(\tau)}$, where $\alpha_{m,n}^{(\tau)} \triangleq \frac{|\mathcal{D}_m|}{\sum_{m' \in \mathcal{E}_n^{(\tau)}} |\mathcal{D}_{m'}|}$ denotes the aggregation weight of the m-th vehicle.

The edge distribution, the local update, and the edge aggregation form one edge epoch, which is repeated τ_e times until the next cloud aggregation starts.

4) Cloud aggregation: The cloud server receives, aggregates, and distributes edge models. The aggregation process is denoted by $\boldsymbol{w}_{e,n}^{(\tau)} = \boldsymbol{w}^{(\tau)} = \sum_{n=1}^{N} \theta_n^{(\tau)} \tilde{\boldsymbol{w}}_{e,n}^{(\tau)}$, where $\theta_n^{(\tau)} \triangleq \frac{\sum_{m \in \mathcal{E}_n^{(\tau)}} |\mathcal{D}_m|}{\sum_{m=1}^{M} |\mathcal{D}_m|}$ denotes the weight of the n-th edge server.

All four steps above form one cloud epoch, which is repeated K times. We call τ_l, τ_e, K the local period, the edge period and the cloud period, respectively.

Following the steps above, the evolution of the local model at the m-th vehicle is denoted by

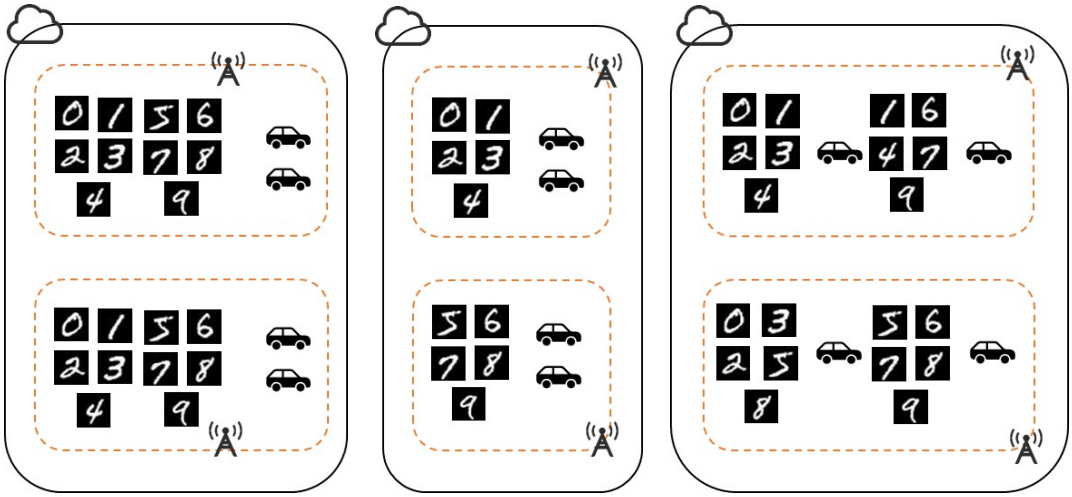
$$\boldsymbol{w}_m^{(\tau)} = \begin{cases} \tilde{\boldsymbol{w}}_m^{(\tau)}, & \tau_l \nmid \tau \\ \tilde{\boldsymbol{w}}_{e,n}^{(\tau)}, & \tau_l \mid \tau, \tau_l \tau_e \nmid \tau \\ \boldsymbol{w}^{(\tau)}, & \tau_l \tau_e \mid \tau \end{cases},$$

Here $a \mid b$ means b is divisible by a, while $a \nmid b$ means b is not divisible by a.

The main notations are listed in Table I, and the details of the training procedure are described in Algorithm 1.

B. Data Distribution

The performance of HFL is affected by the data heterogeneity during training, which is mainly related to the initial data distribution and the mobility of vehicles. Except for the



(a) The i.i.d. case. (b) The edge non-i.i.d. case. (c) The (local) non-i.i.d. case.

Fig. 3. Illustration of three initial data distributions.

vehicle-level heterogeneity that has been widely taken into account in literature, we further consider edge-level heterogeneity, where the dataset of an edge server is denoted by the union of the datasets of all the vehicles within its coverage. Three typical sampling cases of initial data distributions are considered:

(1) *i.i.d.* The data patterns are homogeneous for all vehicles, and each vehicle owns a dataset of the same distribution. As shown in Fig. 3(a), taking the MNIST dataset as a visualizing example, each vehicle holds the same number of images containing digits from 0 to 9.

(2) *edge non-i.i.d.* The training tasks have spatial dependence, such as trajectory forecasting and beam selection, where the data patterns depend on the location of vehicles. In this case, the data distributions are non-i.i.d. across edges, but i.i.d. among the vehicles within the same edge server. For example, in Fig. 3(c), the top edge server holds 0 to 4, and the bottom edge server holds 5 to 9. All of the vehicles within an edge server hold the same digits.

(3) *local non-i.i.d.* The data patterns are heterogeneous for different vehicles. In this case, vehicles own datasets of different distributions, as shown in Fig. 3(b).

In the local non-i.i.d. case, the edge-level heterogeneity usually exists, but it happens when the edge servers hold homogeneous data. For example, if the top right vehicle and the bottom right vehicle swap their positions, the data on edge servers would become i.i.d. This is called *local non-i.i.d. edge i.i.d.*. Generally, if the number of vehicles is large enough, the dataset of each edge server is more likely to be homogeneous, since the data of vehicles is sufficiently mixed up.

C. Mobility Model

Since the edge server that each vehicle belongs to at the time of edge aggregation is needed, the mobility of vehicles is modeled by a discrete Markov chain. The observation interval of the Markov chain is equal to the interval between two consecutive edge aggregations. The states of the Markov chain represent the relationship between the edge server and the vehicle, with a state space size N. Denote the transition probability matrix by $\boldsymbol{P} \in \mathcal{R}^{N\times N}$, where the i-th row and j-th column element $\boldsymbol{P}_{ij}$ represents the probability of a vehicle that stays within the i-th edge server at the last epoch and moves to the coverage of the j-th edge server at the current epoch.

IV. Convergence Analysis

In this section, the upper bound on the loss function of the Mob-HierFAVG algorithm for general tasks and classification tasks are derived. The impact of mobility is then discussed based on the bounds.

A. Convergence of HFL in static and mobile scenarios

We first quantify data heterogeneity by the gradient difference. Define δ_m and $\Delta_n^{(\tau)}$ as

$$\left\|\nabla f_m\left(\boldsymbol{w}\right)-\nabla F\left(\boldsymbol{w}\right)\right\| \leq \delta_m,$$
$$\left\|\nabla F_{e,n}^{(\tau)}\left(\boldsymbol{w}\right)-\nabla F\left(\boldsymbol{w}\right)\right\| \leq \Delta_n^{(\tau)},$$

where $F_{e,n}^{(\tau)}\left(\boldsymbol{w}\right) \triangleq \sum_{m\in\mathcal{E}_n^{(\tau)}} \alpha_{m,n}^{(\tau)} f_m\left(\boldsymbol{w}\right)$. Here, δ_m and $\Delta_n^{(\tau)}$ represent the data distribution difference between the dataset of the m-th vehicle and the cloud server, and between the n-th edge server and the cloud server at the τ-th local iteration, respectively.

Furthermore, the edge-local gradient difference is represented by

$$\delta_n^{(\tau)} \triangleq \sum_{m\in\mathcal{E}_n^{(\tau)}} \alpha_{m,n}^{(\tau)}\delta_m,$$

and the cloud-local and cloud-edge gradient differences are

$$\delta \triangleq \sum_{m=1}^{M} \alpha_m\delta_m, \quad \Delta^{(\tau)} \triangleq \sum_{n=1}^{N} \theta_n^{(\tau)}\Delta_n^{(\tau)},$$

respectively, where $\alpha_m \triangleq \frac{|\mathcal{D}_m|}{|\mathcal{D}|}$.

The definition indicates that the cloud-local gradient difference is not time-varying, while the cloud-edge and edge-local gradient differences are. This is because the whole datasets of vehicles remain the same during training, while

the dataset of each edge server varies over time due to the mobility of vehicles. Therefore, δ_m and $\Delta_n^{(\tau)}$ represent the **data heterogeneity** impacted by the **mobility** of vehicles.

Next, a general convergence bound for HFL is introduced. Define the following virtual models:

1) $\boldsymbol{u}^{(\tau)}$: The virtual cloud model. It records the weighted sum of local models at each iteration by:
$$\boldsymbol{u}^{(\tau)} \triangleq \sum_m \alpha_m \boldsymbol{w}_m^{(\tau)}.$$
2) $\boldsymbol{v}^{(\tau)}$: The virtual centralized model. It monitors the process of centralized learning and synchronizes with the virtual cloud model periodically. It evolves as
$$\boldsymbol{v}^{(\tau)} \triangleq \begin{cases} \tilde{\boldsymbol{v}}^{(\tau)} = \boldsymbol{v}^{(\tau-1)} - \eta \nabla F\left(\boldsymbol{v}^{(\tau-1)}\right), \tau_l\tau_e \nmid \tau \\ \boldsymbol{u}^{(\tau)}, \quad \tau_l\tau_e \mid \tau \end{cases}. \quad (1)$$

The upper bound on the central-federate (CF) difference at the k-th cloud epoch is denoted by U_k, where
$$\left\| \boldsymbol{u}^{(k\tau_l\tau_e)} - \tilde{\boldsymbol{v}}^{(k\tau_l\tau_e)} \right\| \leq U_k.$$
U_k represents the difference of the convergence speed between HFL and centralized learning. Also, denote the optimal model parameters by $\boldsymbol{w}^*$.

Assumption 1. *We assume the following for all m:*

1) $f_m(\boldsymbol{w})$ *is convex;*
2) $f_m(\boldsymbol{w})$ *is ρ-Lipschitz, i.e.,* $\|f_m(\boldsymbol{w}) - f_m(\boldsymbol{w}')\| \leq \rho \|\boldsymbol{w} - \boldsymbol{w}'\|$ *for any $\boldsymbol{w}, \boldsymbol{w}'$;*
3) $f_m(\boldsymbol{w})$ *is β-smooth, i.e.,* $\|\nabla f_m(\boldsymbol{w}) - \nabla f_m(\boldsymbol{w}')\| \leq \beta \|\boldsymbol{w} - \boldsymbol{w}'\|$ *for any $\boldsymbol{w}, \boldsymbol{w}'$.*

Proposition 1. *For any m, if assumption 1 holds, and for some $\epsilon \geq 0$, we have*

(1) $\eta \leq \frac{1}{\beta}$,
(2) $\eta\varphi - \frac{\rho U_k}{\tau_l\tau_e\epsilon^2} > 0$ *for all k, where* $\varphi \triangleq \min_k \frac{1-\frac{\beta\eta}{2}}{\left\|\tilde{\boldsymbol{v}}^{((k-1)\tau_l\tau_e)} - \boldsymbol{w}^*\right\|^2}$,
(3) $F\left(\tilde{\boldsymbol{v}}^{(k\tau_l\tau_e)}\right) - F(\boldsymbol{w}^*) \geq \epsilon$ *for all k,*
(4) $F\left(\boldsymbol{w}^{(K\tau_l\tau_e)}\right) - F(\boldsymbol{w}^*) \geq \epsilon$,

after $T \triangleq K\tau_l\tau_e$ local updates, the loss function of HFL is bounded by
$$F\left(\boldsymbol{w}^{(T)}\right) - F(\boldsymbol{w}^*) \leq \frac{1}{T\eta\varphi - \frac{\rho}{\epsilon^2}\sum_{k=1}^{K} U_k}.$$

Proof. See Appendix A. □

We then focus on bounding the CF difference in the static scenario and mobile scenario, respectively.

1) Static Scenario: In this scenario, we assume vehicles move in a small range and thus do not cross the coverage of edge servers during training. Then the edge gradient difference satisfies $\Delta^{(\tau)} = \Delta$.

Theorem 1. *In the static scenario, the CF difference has an upper bound*
$$U_{k,\text{nomob}} = r(\tau_l\tau_e, \eta, \delta) - \eta(\delta - \Delta)\left[\frac{1}{2}\tau_e(\tau_e - 1)\tau_l + H(\tau_l, \tau_e)\right], \quad (2)$$
where $r(\tau, \eta, \delta) = \frac{\delta}{\beta}\left[(1+\eta\beta)^\tau - 1\right] - \tau\eta\delta$, and $H(\cdot)$ is a non-polynomial function of τ_l and τ_e.

Proof. See Appendix B. □

Theorem 1 shows that the upper bound on the CF difference is irrelevant to the cloud epoch k, so we further set $U = U_{k,\text{nomob}}$.

Here $r(\tau_l\tau_e, \eta, \delta)$ is equals to the CF difference of FL with aggregation period $\tau_l\tau_e$, learning rate η, and gradient difference δ. The right hand side of Eq. (2) can be recognized as the improvement of HFL over FL by introducing edge servers as the intermediate layer. It is shown that the improvement of HFL is determined by both the data heterogeneity δ, Δ and aggregation periods τ_l, τ_e.

Besides, τ_l and τ_e are not symmetric. If $\tau_l\tau_e$ is set to a fixed value, then the choice of τ_l and τ_e still influences the convergence speed. This is aligned with the conclusions in [26].

2) Mobility Scenario: We now consider that vehicles have high mobility and may cross the coverage of edge servers.

Theorem 2. *In the mobility scenario, the CF difference has an upper bound*
$$U_{k,\text{mob}} = r(\tau_l\tau_e, \eta, \delta) - \eta\tau_l\left[\frac{1}{2}\tau_e(\tau_e - 1)\delta - \sum_{j=1}^{\tau_e - 1} j\Delta^{[(k-1)\tau_e + j]}\right], \quad (3)$$
where $\Delta^{[j]} = \Delta^{(j\tau_l)}$.

Proof. See Appendix C. □

Compared with Eq. (2), Δ is substituted by $\Delta^{(\tau)}$ in Eq. (3), because the dataset of the edge server is varying with mobility of vehicles.

To compare the convergence bounds of static and mobility scenarios, we define the mobility factor and derive the form of this factor in some typical cases.

Definition 1. *Define the mobility factor γ_k as*
$$\gamma_k = \frac{U_{k,\text{nomob}} - U_{k,\text{mob}}}{\tau_l\tau_e} \quad (4)$$
$$= \frac{\eta}{\tau_e}\left[\sum_{j=1}^{\tau_e - 1} j\left(\Delta^{[0]} - \Delta^{[(k-1)\tau_e + j]}\right) - \left(\delta - \Delta^{[0]}\right)\frac{H(\tau_l, \tau_e)}{\tau_l}\right],$$
which characterizes the effect of mobility on the convergence speed at the k-th cloud epoch normalized by local updates.

The convergence bound of HFL with mobility is then expressed by
$$F\left(\boldsymbol{w}_{\text{mob}}^{(T)}\right) - F(\boldsymbol{w}^*) \leq \frac{1}{T\left(\eta\varphi - \frac{1}{K}\sum_{k=1}^{K}\frac{\rho U_{k,\text{mob}}}{\tau_l\tau_e\epsilon^2}\right)} = \frac{1}{T\left[\eta\varphi + \frac{\rho}{\epsilon^2}\left(\frac{1}{K}\sum_{k=1}^{K}\gamma_k - \frac{U}{T}\right)\right]}.$$

Therefore, the effect of mobility with fixed η, T and U is determined by $\gamma \triangleq \frac{1}{K}\sum_{k=1}^{K}\gamma_k$. If $\gamma > 0$, mobility accelerates the convergence speed of HFL; if $\gamma = 0$, mobility has no apparent influence on the convergence speed; if $\gamma < 0$, mobility degrades the performance of HFL.

We also notice that γ_k is composed of two parts, with the core elements $\Delta^{[0]} - \Delta^{[(k-1)\tau_e+j]}$ and $(\delta - \Delta^{[0]})H(\tau_l, \tau_e)$, respectively. Here the first element expresses the evolution of the cloud-edge gradient difference, which is caused by the fusion of data on vehicles as they move across the coverage of edge servers. The second element emerges because some vehicles receive models from one edge server and upload their model updates to another edge server, which decreases the quality of models. Therefore, mobility influences the convergence bound of data-heterogeneous HFL in two ways, **fusing the edge data** and **shuffling the edge models**.

B. Convergence of HFL for classification tasks

To further analyze whether the mobility factor γ is positive or negative, the magnitude relationship among δ, $\Delta^{[0]}$ and $\Delta^{[(k-1)\tau_e+j]}$ should be figured out.

1) Mobility Factor for Classification Problem:

Since δ, $\Delta^{[0]}$ and $\Delta^{[(k-1)\tau_e+j]}$ are all related to the loss function, we further consider a classification task with C classes, which adopts the cross-entropy loss. Denote $p\{y=c\}$ and $p_n^{[j]}\{y=c\}$ by the proportion of samples with label c in the global dataset and the dataset of the n-th edge server at the $j\tau_l$-th iteration, respectively. Then the probability vectors is expressed by $\boldsymbol{p} = [p\{y=1\}, ..., p\{y=c\}]$, $\boldsymbol{p}_n^{[j]} = [p_n^{[j]}\{y=1\}, ..., p_n^{[j]}\{y=c\}]$.

Assumption 2. *The global and edge loss functions for the classification task are denoted by*

$$F(\boldsymbol{w}) = \sum_{c=1}^{C} p\{y=c\}\,\mathbb{E}_{\boldsymbol{x}|y=c}[\log g_i(\boldsymbol{w})],$$

$$F_{e,n}^{(j\tau_l)}(\boldsymbol{w}) = \sum_{c=1}^{C} p_n^{[j]}\{y=c\}\,\mathbb{E}_{\boldsymbol{x}|y=c}[\log g_i(\boldsymbol{w})],$$

Assumption 2 claims that the gradient difference can be expressed by the weighted sum of expectation of loss function over each class. In other words, the loss function, as well as its gradients, are determined by the class distribution.

Based on this assumption, the relationship between δ and $\Delta^{[0]}$ is derived as follows.

Proposition 2. *For classification tasks with the i.i.d. and edge non-i.i.d. initial distributions, $\delta - \Delta^{[0]} = 0$.*

Proof. For the i.i.d. case, the local and edge data distribution is identical to the global data distribution. According to Assumption 2, $\delta = \Delta^{[0]} = 0$. Similarly, for the edge non-i.i.d. case, the local data distribution is identical to the edge data distribution, so we obtain $\delta = \Delta^{[0]}$. □

Proposition 2 points out that the convergence bounds of the i.i.d. and edge i.i.d. cases are not impacted by shuffling the edge model.

To compare $\Delta^{[0]}$ with $\Delta^{[(k-1)\tau_e+j]}$, we consider deriving an upper bound of $\Delta_n^{[j]}$. Firstly, we have

$$\begin{aligned}
&\Delta_n^{[j]} = \left\|\nabla F_{e,n}^{(j\tau_l)}(\boldsymbol{w}) - \nabla F(\boldsymbol{w})\right\| \\
&= \left\|\sum_{c=1}^{C}\left(p\{y=c\} - p_n^{[j]}\{y=c\}\right)\nabla_w \mathbb{E}_{\boldsymbol{x}|y=c}[\log g_i(\boldsymbol{w})]\right\| \\
&\le \sum_{c=1}^{C}\left\|p\{y=c\} - p_n^{[j]}\{y=c\}\right\|\left\|\nabla_w \mathbb{E}_{\boldsymbol{x}|y=c}[\log g_i(\boldsymbol{w})]\right\| \\
&\le \|\boldsymbol{p} - \boldsymbol{p}_n^{[j]}\|_1 G, \qquad (5)
\end{aligned}$$

where $G = \max_{c,\boldsymbol{w}}\{\left\|\nabla_{\boldsymbol{w}} \mathbb{E}_{\boldsymbol{x}|y=c}[\log g_i(\boldsymbol{w})]\right\|\}$, and $\|\boldsymbol{p} - \boldsymbol{p}_n^{[j]}\|_1 = \sum_{c=1}^{C}\left\|p\{y=c\} - p_n^{[j]}\{y=c\}\right\|$ denotes the probability difference. Therefore, for a classification problem with cross-entropy loss, the variation of edge gradient difference is bounded by the variation of probability difference $\|\boldsymbol{p} - \boldsymbol{p}_n^{[j]}\|_1$.

Next, the mobility model is introduced to further decompose the probability difference. Assume the mobility of vehicles follows a discrete Markov model as stated in Section II. Then the transition probability of vehicles from edge server n to edge server n' during the last edge epoch is denoted by

$$\boldsymbol{Q}_{n,n'} = \mathbb{P}\left[m \in \mathcal{E}_{n'}^{((j+1)\tau_l)} | m \in \mathcal{E}_n^{(j\tau_l)}\right].$$

To simplify the model, we assume that the number of vehicles within the coverage of the n-th edge server does not change sharply in two adjacent edge aggregations. That is to say, $\theta_n^{(j\tau_l)} \approx \theta_n^{(j+1)\tau_l}$ for all n.

As the training goes on, the label distributions of edge servers change accordingly by

$$\boldsymbol{p}_n^{[j+1]} = \frac{\sum_{n'}\theta_{n'}^{(j\tau_l)}\boldsymbol{Q}_{n',n}\boldsymbol{p}_{n'}^{[j]}}{\theta_n^{((j+1)\tau_l)}} = \sum_{n'}\boldsymbol{Q}_{n',n}\boldsymbol{p}_{n'}^{[j]}.$$

Define $\boldsymbol{P}^{[j]} = [\boldsymbol{p}_1^{[j]}, \boldsymbol{p}_2^{[j]}, \cdots, \boldsymbol{p}_N^{[j]}]$ as the label probability matrix, and the transition of $\boldsymbol{P}^{[j]}$ can be written as

$$\boldsymbol{P}^{[j+1]} = \boldsymbol{P}^{[j]}\boldsymbol{Q}.$$

Then the probability difference is bounded *w.r.t.* the mobility model.

Lemma 1. *If all vehicles follow a discrete Markov model with a transition probability matrix $\boldsymbol{Q}$, then the probability difference of edge server n in $j\tau_l$-th iteration is bounded by:*

$$\|\boldsymbol{p} - \boldsymbol{p}_n^{[j]}\|_1 \le NL_n\|\lambda_*\|^j, \qquad (6)$$

where L_n is a constant determined by n, and λ_* is the largest eigenvalue of $\boldsymbol{Q}$ satisfying $\|\lambda_*\| < 1$.

Proof. See Appendix D. □

Lemma 1 indicates that the probability difference decreases as the training goes on with a speed of $\|\lambda_*\|$. Taking Eq. (5) into Eq. (6), we obtain

$$\Delta^{[j]} \le GN\|\lambda_*\|^j\sum_{n=1}^{N}\theta_n L_n = G\|\lambda_*\|^j NL, \qquad (7)$$

where $L = \sum_{n=1}^{N}\theta_n L_n$.

Now we derive γ *w.r.t.* initial data distributions.

Theorem 3. *For classification tasks, the mobility factors with different initial distributions have the following form:*

$$\textit{i.i.d.}: \gamma_k = 0, \tag{8}$$

$$\textit{local non-i.i.d. / edge i.i.d.}: \gamma_k = -\frac{\eta\delta}{\tau_l \tau_e} H(\tau_l, \tau_e), \tag{9}$$

$$\textit{edge non-i.i.d.}: \gamma_k = \frac{\eta GNL}{\tau_e} \sum_{j=1}^{\tau_e - 1} j \left(1 - \|\lambda_*\|^{(k-1)\tau_e + j}\right), \tag{10}$$

Proof. For the i.i.d. case, since the local and edge data distributions are always identical to the global data distribution, we have $\delta = \Delta^{[j]} = 0$ for all j, so Eq. (8) is derived.

For the local non-i.i.d. / edge i.i.d. case, $\delta \gg 0$ because the datasets of all vehicles are heterogeneous, and $\Delta^{[j]} = 0$ for all j because the datasets of edge servers are assumed to be i.i.d.. Since the upper bound of $\Delta^{[j]}$ decreases over j as shown in Eq. (7), we can further assume edge i.i.d. always holds during training, i.e. $\Delta^{[j]} = 0$ for all j. Taking it into Eq. (4), Eq. (9) is obtained. Here $\gamma_k < 0$ for all k, so $\gamma < 0$.

For the edge non-i.i.d. case, we assume that $\Delta^{[0]} - \Delta^{[(k-1)\tau_e + j]} \approx GNL(\|\lambda_*\|^0 - \|\lambda_*\|^j)$. Inserting (7) into (4), and based on Proposition 2, Eq. (10) is obtained. Here $\gamma_k > 0$ for all k since $\|\lambda_*\| \leq 1$, so we have $\gamma > 0$. □

From Theorem 3, the impact of mobility on data-heterogeneous HFL depends on the initial data distribution. Mobility has no apparent influence on HFL with i.i.d. data. In the local non-i.i.d. / edge i.i.d. case, mobility decreases the convergence speed of HFL by **shuffling the edge models**. In the edge non-i.i.d. case, mobility accelerates the convergence speed of HFL by **fusing the edge data**.

On the other hand, the performance of HFL with different data distributions can also be compared.

In the static scenario, the CF difference is bounded by Eq. (2). For the i.i.d. case, $\delta = \Delta = 0$, so the CF difference equals to 0. For the local non-i.i.d. and edge non-i.i.d. case, assuming the vehicle-level data heterogeneity δ is the same, we have $\Delta_{\text{local niid}} < \Delta_{\text{edge niid}}$, since the data heterogeneity of edge servers is as much as that of vehicles in the edge non-i.i.d. case. Therefore, the performance of the local non-i.i.d. case should be better than that of the edge non-i.i.d. case.

In the mobile scenario, the performance of the i.i.d. case does not change. Meanwhile, the edge non-i.i.d. case converges faster, while the local non-i.i.d. case may converge more slowly, so it's possible that $\Delta^{[j]}_{\text{local niid}} > \Delta^{[j]}_{\text{edge niid}}$ for some j, and ultimately the performance of the edge non-i.i.d. case may surpass that of the local non-i.i.d. case.

2) Mobility Factor for Ring Topology: Next, we want to further dig into the edge non-i.i.d. case to study the improvement brought by mobility in more detail. The mobility factor in Eq. (10) connects the convergence bound of HFL with the eigenvalues of the transition probability matrix, but the relation between the eigenvalues and the speed of vehicles is not clear. In this part, a closed-form solution of eigenvalues is derived given specific topology and mobility model.

Consider edge servers are connected from head to tail to form a ring, so all edge servers are homogeneous in topology. The transition probabilities of vehicles is assumed to satisfy

(1) Vehicles within any edge server have the same sojourn probability p_s,

(2) At each transient step, vehicles can only move to the coverage of adjacent edge servers with equal probability.

Under these assumptions, the transition probability matrix is denoted by

$$\boldsymbol{Q} = \begin{bmatrix} p_s & \frac{1-p_s}{2} & 0 & \cdots & 0 & \frac{1-p_s}{2} \\ \frac{1-p_s}{2} & p_s & \frac{1-p_s}{2} & \cdots & 0 & 0 \\ 0 & \frac{1-p_s}{2} & p_s & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & p_s & \frac{1-p_s}{2} \\ \frac{1-p_s}{2} & 0 & 0 & \cdots & \frac{1-p_s}{2} & p_s \end{bmatrix}. \tag{11}$$

Then the closed-form solution of eigenvalues can be derived.

Lemma 2. *The transition probability matrix* (11) *is a circulant matrix with eigenvalues*

$$\lambda_n = p_s + (1 - p_s)\cos\left(\frac{2n\pi}{N}\right), \quad n = 0, 1, \cdots, N-1. \tag{12}$$

Proof. From (3.2) in [52], the eigenvalues of circulant matrices take the following form

$$\lambda_n = \sum_{k=0}^{N-1} c_k e^{\frac{-2\pi i k}{N}}, n = 0, 1, \cdots, N-1. \tag{13}$$

Here c_k denotes the k-th component in the first row of the matrix. Taking Eq. (11) into Eq. (13), Eq. (12) is obtained. □

Proposition 3. *For HFL with classification tasks, when the edge servers form a ring topology, the mobility factor in the edge non-i.i.d. case is expressed by*

$$\gamma_k = \frac{\eta GNL}{\tau_e} \sum_{j=1}^{\tau_e - 1} j \left[(1 - p_s)\left(1 - \cos\left(2\pi/N\right)\right)\right]^{k\tau_e + j}. \tag{14}$$

Proof. From (12) we have $\lambda_* = p_s + (1 - p_s)\cos\left(\frac{2\pi}{N}\right)$. Inserting it into (10), the proposition is proved. □

From (14) we see that with a ring topology for the edge servers, the mobility factor is directly correlated to the sojourn probability of vehicles. Specifically, when vehicles move faster, their sojourn probability decreases, and the mobility factor γ increases correspondingly, indicating the acceleration of the convergence speed of HFL.

V. Simulation Results

In this section, we conduct simulations on HFL to show the influences of data heterogeneity and mobility and to verify our theoretical analysis.

A. Simulation Settings

We consider a city road system on a square grid, where each side of the square is covered by an edge server, as shown in Fig. 4. There are $N = 4$ edge servers and a total of $M = 32$ vehicles in the system, and each edge server covers 8 vehicles initially. The initial positions of vehicles are uniformly distributed. The Simulation of Urban MObility

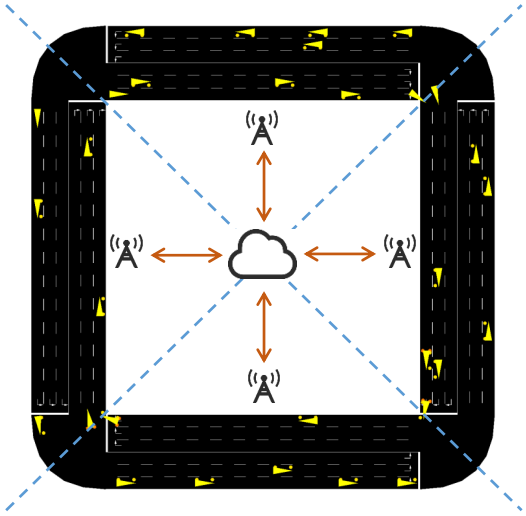


Fig. 4. Simulation scenario: HFL with one cloud server, $N = 4$ edge servers, and $M = 32$ vehicles. The edge servers form a ring topology to compose a square.

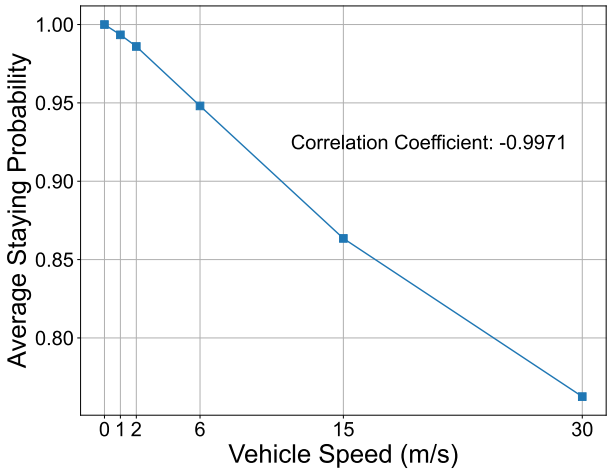


Fig. 5. Average sojourn probability of vehicles with different vehicle speeds. The length of each side of the road is set to $a = 1000$ m.

(SUMO) [53] platform with the Manhattan model is adopted to simulate the mobility of vehicles, which monitors real-world vehicle dynamics. The length of each side of the road is a m, and vehicles travel on the road at a maximum speed of v m/s. Two main scenarios of the vehicle speed in simulations are considered:(1) the static scenario, where $v = 0$; (2) the mobile scenario, where $v = 30$ m/s. The interval between each edge aggregation is assumed to be 1 second. Since the edge servers follow the ring topology in our scenario, the mobility of vehicles is denoted by (11) with respect to the sojourn probability. As shown in Fig. 5, the average sojourn probability is negatively linearly correlated with the vehicle speed v with a high correlation coefficient. Therefore, we can approximately change the sojourn probability by changing the vehicle speed with a linear scaling factor.

HFL simulations are conducted on the CIFAR-10 dataset[1]. We consider three initial data distributions mentioned above: i.i.d. case, local non-i.i.d. case, and edge non-i.i.d. case. To create heterogeneous data by uniformly distributing the classes to edge servers, we only choose data of 8 classes from the dataset, with 40000 training samples and 8000 testing samples in total. For the i.i.d. case, the 40000 training samples are uniformly divided into M disjoint subsets, and allocated to vehicles so that each vehicle owns one subset. For the local non-i.i.d. case, all the data are sorted by labels, and l classes are allocated to each vehicle (denoted by 'local niid(l)' in the following), with all vehicles holding the same number of samples. For the edge non-i.i.d. case, the sorted data is first allocated to each edge server so that each edge server owns l classes of samples, and then the edge servers uniformly allocate their data to vehicles within their coverage area (denoted by 'edge niid(l)' in the following). If not specially noted, l is set to 2.

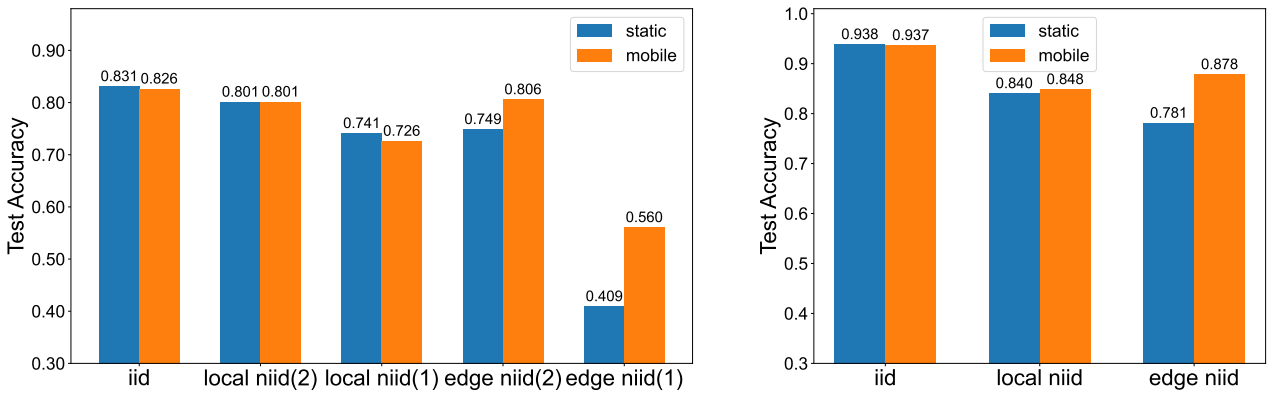
We train a convolutional neural network (CNN) with a batch size of 20 using the following architecture: the first convolutional block consists of two 3×3 convolution layers with 32 channels, ReLU activation, followed by a 2×2 max pooling layer and 0.2 dropout; the second convolutional block contains two 3×3 convolution layers with 64 channels, ReLU activation, followed by a 2×2 max pooling layer and 0.3 dropout; a fully connected layer with 120 units and ReLU activation; and a final 10-unit softmax layer. The network takes an input of size $20 \times 3 \times 32 \times 32$ and produces an output of size 20×10. The ResNet18 model [56], which has shown superior performance on the CIFAR-10 dataset, is also adopted to prove that the system can reach a high accuracy. As stated and proved in [57], models containing batch normalization layers do not converge in FL training when the data is non-i.i.d. To obtain converged results, we further replace all the batch normalization layers with the group normalization layers in the local non-i.i.d. and edge non-i.i.d. case.

The learning rate is set to $\eta = 0.1$. If not especially noted, the local period and edge period are set to $\tau_l = 6$ and $\tau_e = 10$, separately. The Mob-HierFAVG algorithm is run for up to $K = 600$ cloud epochs.

B. Performance

1) Impact of initial data distribution: The performance of HFL under several initial data distributions is shown in Fig 6(a). In the i.i.d. case, the maximum test accuracies of the static and mobile scenarios are approximately equal, which is consistent with (8). In the local non-i.i.d.(2) case, the static accuracy is the same as the mobile accuracy, which is different from the theoretical result of local non-i.i.d. / edge i.i.d. case in (9). This is because the number of simulated vehicles is not large enough to produce edge i.i.d., and as long as edge non-i.i.d. happens in any cloud epoch, the effect of data fusion emerges, which offsets the negative effect of model shuffling. While in the local non-i.i.d.(1) case with

[1]https://www.cs.toronto.edu/ kriz/cifar.html



(a) CNN

(b) ResNet18

Fig. 6. The test accuracies within 600 cloud epochs for different initial data distributions on CNN and ResNet18 models.

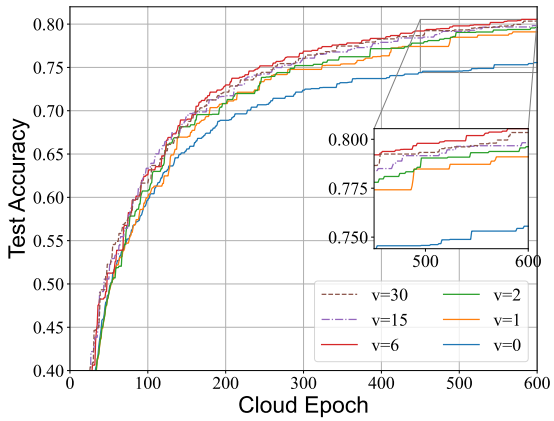
$M = 80$, the test accuracy of $v = 0$ is 1.5% more than that of $v = 30$, which is consistent with the conclusion in (9). In the edge non-i.i.d. case, mobility obviously increases the performance. Specifically, when $l = 2$, mobility improves the accuracy by 5.7% (74.9% to 80.6%); when $l = 1$, mobility improves the accuracy by 15.1% (40.9% to 56.0%). These results reflect that if an edge server holds fewer classes, the accuracy of HFL decreases, while the accuracy improvement brought by mobility increases. Experiments have also been carried out with the ResNet18 model. The results lead to similar conclusions, but with a much higher accuracy, where the i.i.d. and edge non-i.i.d.(2) case reaches the accuracy of 93.3% and 87.8%, separately.

Furthermore, by comparing the results of local non-i.i.d. and edge non-i.i.d. cases when $l = 2$, we find that the local non-i.i.d. case achieves higher accuracy than the edge non-i.i.d. case in the static scenario, but reaches a similar or even lower accuracy with mobility. This is aligned with our analysis in Section IV. This phenomenon can also be described by the change in the equivalent aggregation period of heterogeneous data. In the local non-i.i.d. case, the data heterogeneity exists at the local level, so the edge aggregation sees a fusion of heterogeneous data, and thus the equivalent aggregation period is τ_l local updates. In the edge non-i.i.d. case, the data heterogeneity exists at the edge level, so only the cloud aggregation sees a fusion of heterogeneous data with the equivalent aggregation period of $\tau_l\tau_e$ local updates. As demonstrated in [54], a greater aggregation period results in worse training performance for heterogeneous data. However, when the vehicles start moving, data of different edge servers gradually mix up, and thus the equivalent aggregation period goes back to τ_l.

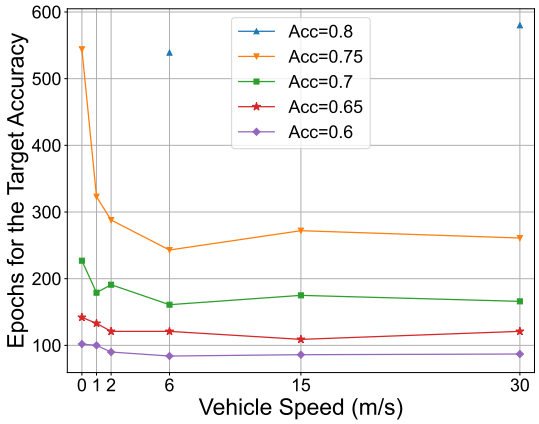
2) Impact of vehicle speed: For the edge non-i.i.d. case, the impact of vehicle speed v on the convergence speed is further studied. As is shown in Fig. 7(a), the test accuracies of the scenarios with mobility, i.e., $v > 0$, are obviously higher than the one achieved when $v = 0$. Furthermore, the rounds to reach certain target accuracies are shown in Fig. 7(b), which demonstrates that with high accuracy requirements, mobility greatly reduces the training epochs, and thus saves resource utilization and power consumption. Specifically, at Acc=0.8, only two points are plotted, since when the vehicle speed takes other values, the test accuracy does not reach 0.8 during training. Also, Fig. 7(b) indicates that it takes nearly the same number of cloud epochs to reach the accuracy of 60% for all scenarios. Models of this accuracy contain basic features of the datasets without the information of class variety and generally can be achieved by pre-training. Actually, this situation usually happens in vehicular networks. For example, for the beam selection task, when the channel condition changes, the model needs to be fine-tuned, and the original model, which carries the basic features for beams, can be set as the pre-trained model.

Therefore, we further assume that the training is started from a pre-trained model with a test accuracy of 60%. With a pre-trained model, the training time is obviously decreased, as shown in Fig. 7(c) and 7(d). Besides, Fig. 7(d) illustrates that a higher speed generally results in faster convergence. In particular, it takes only 142 cloud epochs for the $v = 30$ case to reach an accuracy of 75%, reduced by 65.5% compared with the $v = 0$ case (412 epochs), and 39.8% compared with the $v = 1$ case (236 epochs). These results are in line with the convergence analysis in Section IV.

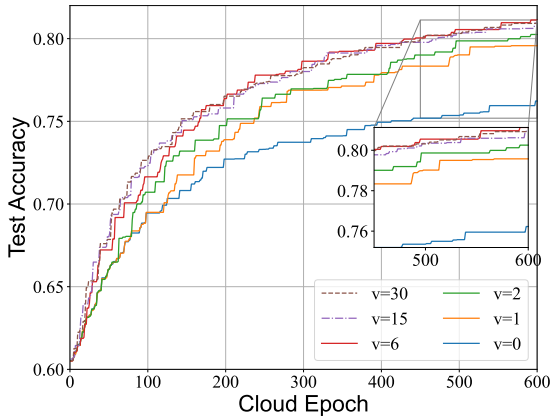
On the other hand, the improvement brought by high mobility is limited. In our simulations, both the final accuracy and the convergence speed typically saturate beyond $v = 6\ m/s$. Fig. 8 shows the evolving trends of the average probability difference which is the average of the probability differences between edge servers and the cloud. A smaller average probability difference indicates that the data on the edge servers is more similar to the data on the cloud, meaning that the data is more sufficiently mixed. The results demonstrate that trends of the average probability differences in $v = 6, 15, 30\ m/s$ cases are similar. Therefore, the convergence performance saturates possibly because data from different edge servers is sufficiently mixed at $v = 6\ m/s$, and no further gains can be achieved with even higher mobility. We also notice that the average probability difference does not monotonically decrease with time, but oscillates downward in a certain cycle. Besides, the theoretical analysis indicates that the average



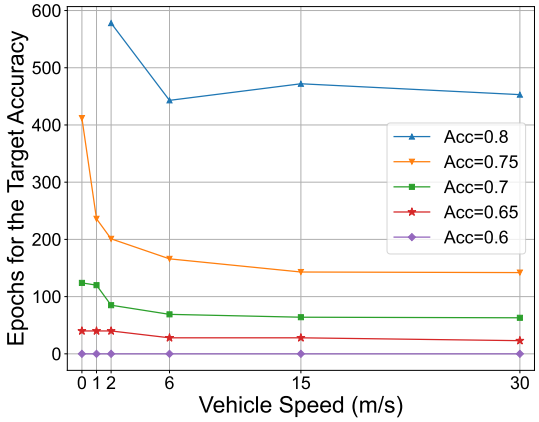
(a) Accuracy on the test dataset when training from scratch.



(b) Epochs to reach the target accuracies when training from scratch.



(c) Accuracy on the test dataset when training from a pretrained model with a test accuracy of 60%.



(d) Epochs to reach the target accuracies when training from a pretrained model with a test accuracy of 60%.

Fig. 7. Performance of HFL with different vehicle speeds.

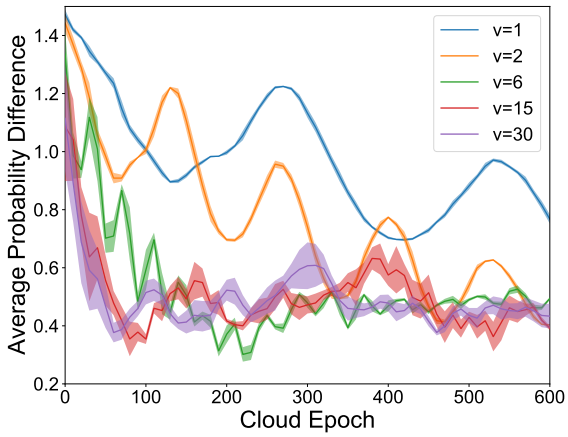


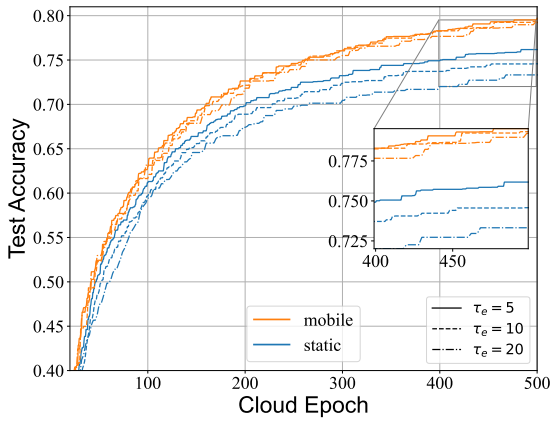
Fig. 8. Variation of the average probability difference $\frac{1}{N}\sum_{n=1}^{N}\|\boldsymbol{p}-\boldsymbol{p}_n^{[j]}\|_1$ during training at different speeds. The values are calculated by averaging the results of 100 SUMO simulations. To make the curve smooth, the total 6000 points (600 cloud epochs $\times$ 10 edge aggregations/cloud epoch) are divided into 60 groups. The average value of each group is plotted and the standard deviation of each group is represented by the shadow around the curve.

probability difference goes to zero after a long enough time, but during the simulation time, the value only decreases to around 0.4. These results may originate from the limited number of vehicles and training epochs.

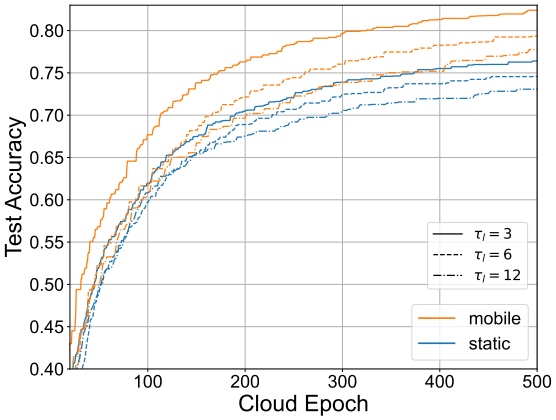
3) Impact of aggregation periods: Eq. (10) shows that the impact of mobility is correlated with not only the sojourn probability but also the aggregation periods and the number of vehicles. Further simulations for the edge non-i.i.d. case with $l = 2$ are carried out for verification. Fig. 9(a) shows the test accuracy of the static and mobile scenarios for several values of τ_e. In the static scenario, a small τ_e results in faster convergence speed and higher accuracy; while in the mobile scenario, different values of τ_e report similar convergence speed and test accuracy values. This finding indicates that mobility reduces the reliance on cloud aggregation, and thus saves communication resources.

The impact of local period τ_l is shown in Fig. 9(b). Although not explicitly expressed in the mobility factor, mobility slightly enhances the benefits of edge aggregation. For the $\tau_l = 12$ case, mobility improves the max test accuracy by 5.2% (73.5% to 78.7%); while for the $\tau_l = 3$ case, mobility improves the max test accuracy by 6.0% (76.4% to 82.4%).

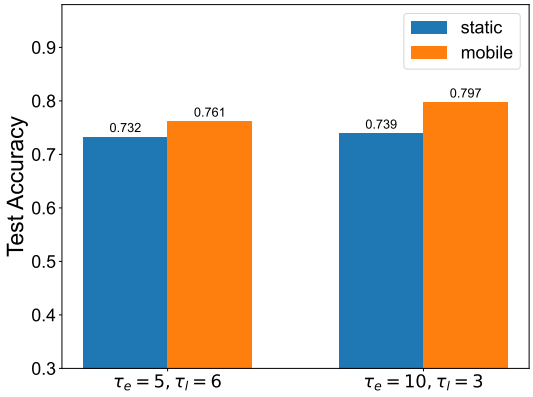
From another perspective, if the local updates per cloud



(a) Different τ_e. ($\tau_l = 6$)



(b) Different τ_l. ($\tau_e = 10$)



(c) Test accuracy within 600 cloud epochs for different τ_e and τ_l with $\tau_e\tau_l = 30$.

Fig. 9. Test accuracy of HFL with different aggregation periods in static and mobile scenarios.

epoch $\tau_l\tau_e$ are fixed, we can learn the trade-off between the edge aggregation and cloud aggregation. Fig. 9(c) tells that mobility brings greater improvement when edge aggregation is more frequent.

4) Impact of the number of participants: For the edge non-i.i.d. case with $l = 2$, we also change the number of vehicles and edge servers to see the change in the test accuracy. As demonstrated in Fig. 10(a), the number of edge servers does not influence the convergence in the static scenario, but if the vehicles have mobility, fewer edge servers contribute to higher test accuracy. To understand the results, we consider two extreme cases: when $N = 1$, the system degenerates to edge-based FL with $\tau = \tau_l$; when $N = M$, the system degenerates to cloud-based FL with $\tau = \tau_l\tau_e$. The former case converges fast because the vehicles aggregate more frequently. As N increases, the number of vehicles within one edge decreases, and if the data of vehicles are non-i.i.d., the models of edge servers will become more and more unstable, which causes the performance to degrade. However, this result is based on the assumption that the cloud server has a certain coverage. If the number of edge servers decreases, each edge server needs to cover a larger area, with a longer communication delay during the edge aggregation.

Fig. 10(b) demonstrates that more vehicles lead to higher test accuracy for both the static and mobile scenarios. This result matches the conclusion in FL: with more clients participating in the edge aggregation, the aggregated model is more stable, thus bringing a higher convergence speed. Besides, the improvement caused by adding vehicles is sharper for the mobile scenario, because the probability difference of edges with more vehicles becomes smaller as more data mixes up.

VI. Conclusion

In this article, we have investigated the impact of mobility on data-heterogeneous HFL. The convergence analysis of HFL has been conducted, showing that mobility influences the performance of HFL by fusing the edge data and shuffling the edge models. Next, the mobility factors under three initial data distributions for classification tasks have been derived, which proves that mobility increases the convergence speed of HFL in the edge non-i.i.d. case. Furthermore, when the edge servers form a ring topology, a higher speed of vehicles lead to faster convergence of HFL. Simulations have been carried out on the CIFAR-10 dataset and the SUMO platform, demonstrating the benefits of mobility from multiple perspectives. Results have shown that mobility brings an increase in test accuracy by up to 15.1% in the edge non-i.i.d. case. Higher speed results in higher test accuracy, while the achieved accuracy saturates when the speed of vehicles exceeds 6 m/s. Moreover, mobility has also shown the potential for reducing training costs.

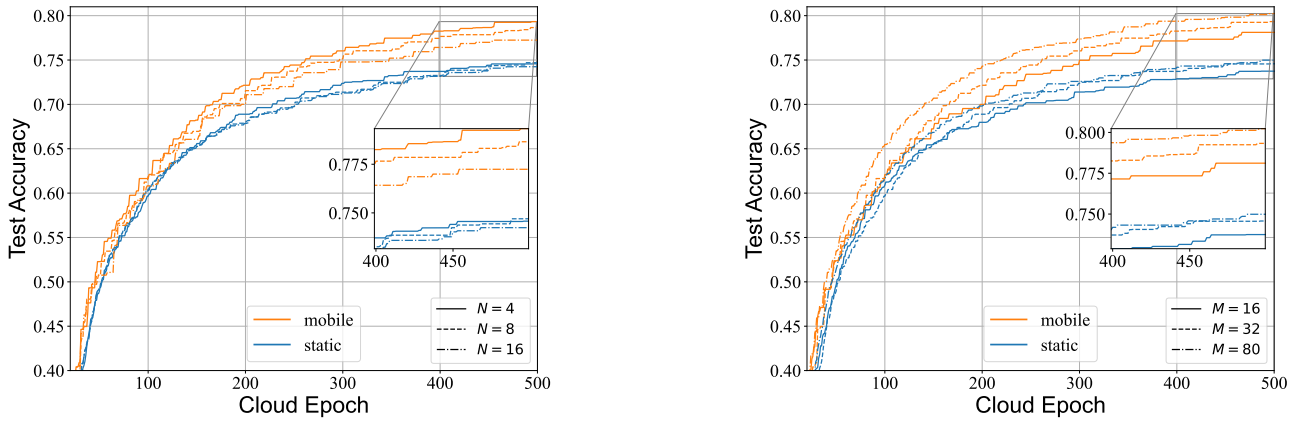
In the future, we plan to consider communication constraints. Due to the limited bandwidth, not all vehicles will be able to successfully upload their models. Convergence analysis will need to be reperformed to account for the partial aggregation of vehicle models. Subsequently, optimization strategies can be employed to maximize the performance of HFL with mobility based on the convergence bound.

Appendix A
Proof of Proposition 1

This is an extension of Lemma 2 in [46]. The proof is provided in [58].

Appendix B
Proof of Theorem 1

In the static scenario, the vehicles do not step from the coverage of one vehicle to another, so the covering vehicle



(a) Different numbers of edge servers. ($M = 32$)

(b) Different numbers of vehicles. ($N = 4$)

Fig. 10. Test accuracy of HFL with different numbers of vehicles for the static and mobile scenarios.

set satisfies $\mathcal{E}_n^{(\tau)} = \mathcal{E}_n$, and the aggregation weights satisfies $\alpha_{m,n}^{(\tau)} = \alpha_{m,n}, \theta_n^{(\tau)} = \theta_n$ for all m, n. Denote the virtual edge model as $\boldsymbol{u}_n^{(\tau)} \triangleq \sum_{m\in\mathcal{E}_n} \alpha_{m,n}\boldsymbol{w}_m^{(\tau)}$.

To prove Theorem 1, we first propose some lemmas. We provide a brief proof for each lemma in the following, while the complete proof can be found in [58].

Lemma 3. *For the virtual cloud model, we have*

$$\left\|\boldsymbol{u}^{(\tau)}-\tilde{\boldsymbol{v}}^{(\tau)}\right\| \leq \begin{cases} \left\|\boldsymbol{u}^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\|+\eta\beta\sum_m \alpha_{m,n}\left\|\boldsymbol{w}_m^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\|, & \tau_l \nmid \tau-1, \\ \left\|\boldsymbol{u}^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\|+\eta\beta\sum_n \theta_n\left\|\boldsymbol{u}_n^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\|, & \tau_l \mid \tau-1, \tau_l\tau_e \nmid \tau-1, \\ 0, & \tau_l\tau_e \mid \tau-1, \end{cases} \tag{15}$$

Proof. When $\tau_l \nmid \tau-1$, we obtain

$$\begin{aligned} &\left\|\boldsymbol{u}^{(\tau)}-\tilde{\boldsymbol{v}}^{(\tau)}\right\| \\ =&\Bigg\|\left[\boldsymbol{u}^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right] \\ &-\eta\sum_m \alpha_{m,n}\left[\nabla f_m\left(\boldsymbol{w}_m^{(\tau-1)}\right)-\nabla f_m\left(\boldsymbol{v}^{(\tau-1)}\right)\right]\Bigg\| \\ \leq&\left\|\boldsymbol{u}^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\|+\eta\beta\sum_m \alpha_{m,n}\left\|\boldsymbol{w}_m^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\|, \end{aligned}$$

where the inequality holds because of the triangle inequality and the assumption of β-smoothness.

Similarly, when $\tau_l \mid \tau-1$ and $\tau_l\tau_e \nmid \tau-1$, we have

$$\begin{aligned} &\left\|\boldsymbol{u}^{(\tau)}-\tilde{\boldsymbol{v}}^{(\tau)}\right\| \\ \leq&\left\|\boldsymbol{u}^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\|+\eta\beta\sum_n \theta_n\left\|\boldsymbol{u}_n^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\|. \end{aligned}$$

When $\tau_l\tau_e \mid \tau-1$, let $\tau = k\tau_l\tau_e+1$, and thus

$$\begin{aligned} \left\|\boldsymbol{u}^{(\tau)}-\tilde{\boldsymbol{v}}^{(\tau)}\right\| =&\Bigg\|\left[\boldsymbol{u}^{(k\tau_l\tau_e)}-\eta\nabla F\left(\boldsymbol{u}^{(k\tau_l\tau_e)}\right)\right] \\ &-\left[\boldsymbol{v}^{(k\tau_l\tau_e)}-\eta\nabla F\left(\boldsymbol{v}^{(k\tau_l\tau_e)}\right)\right]\Bigg\| = 0. \end{aligned}$$

where the first equality holds because $\boldsymbol{w}_m^{(k\tau_l\tau_e)} = \boldsymbol{u}^{(k\tau_l\tau_e)}$ for all m and $\sum_m \alpha_{m,n} f_m(\cdot) = F(\cdot)$, and the last equality holds because $\boldsymbol{u}^{(k\tau_l\tau_e)} = \boldsymbol{v}^{(k\tau_l\tau_e)}$ according to the definition. □

Lemma 4. *For the virtual edge model, we have*

$$\left\|\boldsymbol{u}_n^{(\tau)}-\tilde{\boldsymbol{v}}^{(\tau)}\right\| \leq \begin{cases} (1+\eta\beta)\left\|\boldsymbol{u}_n^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\|+\eta\Delta_n, \tau_l \mid \tau-1 \\ \left\|\boldsymbol{u}_n^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\| \\ +\eta\beta\sum\limits_{m\in\mathcal{E}_n}\alpha_{m,n}\left\|\boldsymbol{w}_m^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\|+\eta\Delta_n, \tau_l \nmid \tau-1. \end{cases}$$

The proof is similar to Lemma 3.

Lemma 5. *Denote $\tau = k\tau_l\tau_e+\tau_0, \tau_0 \in (0, \tau_l\tau_e]$, and we have the following for the m-th local model:*

$$\left\|\boldsymbol{w}_m^{(\tau)}-\tilde{\boldsymbol{v}}^{(\tau)}\right\| \leq \frac{\delta_m}{\beta}[(1+\eta\beta)^{\tau_0}-1]. \tag{16}$$

Proof. The m-th local model is bounded by

$$\begin{aligned} &\left\|\boldsymbol{w}_m^{(\tau)}-\tilde{\boldsymbol{v}}^{(\tau)}\right\| \\ =&\Bigg\|\left[\boldsymbol{w}_m^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right]-\left[\eta\nabla f_m\left(\boldsymbol{w}_m^{(\tau-1)}\right)-\eta\nabla f_m\left(\boldsymbol{v}^{(\tau-1)}\right)\right] \\ &-\left[\eta\nabla f_m\left(\boldsymbol{v}^{(\tau-1)}\right)-\eta\nabla F\left(\boldsymbol{v}^{(\tau-1)}\right)\right]\Bigg\| \\ \leq&(1+\eta\beta)\left\|\boldsymbol{w}_m^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\|+\eta\delta_m. \end{aligned} \tag{17}$$

We then prove (16) by mathematical induction.

Firstly, the conclusion holds when $\tau_0 = 1$, since $\boldsymbol{w}_m^{(\tau-1)} = \boldsymbol{v}^{(\tau-1)}$ when $\tau_l\tau_e \mid \tau-1$. Secondly, if the conclusion holds for $\tau_0 = t-1$, then we have the following:

$$\begin{aligned} &\left\|\boldsymbol{w}_m^{(k\tau_l\tau_e+t)}-\tilde{\boldsymbol{v}}^{(k\tau_l\tau_e+t)}\right\| \\ \leq&(1+\eta\beta)\left\|\boldsymbol{w}_m^{(k\tau_l\tau_e+t-1)}-\boldsymbol{v}^{(k\tau_l\tau_e+t-1)}\right\|+\eta\delta_m \\ =&(1+\eta\beta)\left\|\boldsymbol{w}_m^{(k\tau_l\tau_e+t-1)}-\tilde{\boldsymbol{v}}^{(k\tau_l\tau_e+t-1)}\right\|+\eta\delta_m \\ =&(1+\eta\beta)\frac{\delta_m}{\beta}[(1+\eta\beta)^{t-1}-1]+\eta\delta_m \\ =&\frac{\delta_m}{\beta}[(1+\eta\beta)^{t}-1]. \end{aligned}$$

where the inequality comes from (17), and the first equality holds because $\boldsymbol{v}^{(\tau)}=\tilde{\boldsymbol{v}}^{(\tau)}$ when $\tau_l\tau_e \nmid \tau$ according to Eq. (1).

So Eq. (16) holds for $\tau_0=t$, and the lemma is proved. □

Lemma 6. *Denote $\tau=k\tau_l\tau_e+\tau_0, \tau_0\in(0,\tau_l\tau_e]$, and we have the following for the virtual edge model:*

$$\left\|\boldsymbol{u}_n^{(\tau)}-\tilde{\boldsymbol{v}}^{(\tau)}\right\| \leq \frac{\delta_n}{\beta}[(1+\eta\beta)^{\tau_0}-1] - \eta\left(\delta_n-\Delta_n\right)[\tau_0+h\left(\tau_0\right)], \quad (18)$$

where $h\left(t\right)=\tau_l\big[\sum_{r=1}^{R_t}\left(1+\eta\beta\right)^r-R_t\big]$ and $R_t=\lfloor\frac{t-1}{\tau_l}\rfloor$.

Proof. We first derive the recursive form of the virtual edge model similar to (17), and then prove the lemma by mathematical induction. The complete proof can be found in [58]. □

Now we can turn to prove Theorem 1. By substituting (16) and (18) into (15), we obtain

$$\left\|\boldsymbol{u}^{(\tau)}-\tilde{\boldsymbol{v}}^{(\tau)}\right\| \leq \begin{cases} 0, \quad \tau_l\tau_e \mid \tau-1, \\ \left\|\boldsymbol{u}^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\|+\eta\delta[(1+\eta\beta)^{\tau-1}-1], \tau_l \nmid \tau-1, \\ \left\|\boldsymbol{u}^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\|+\eta\delta[(1+\eta\beta)^{\tau-1}-1] \\ -\eta\left(\delta-\Delta\right)[(\tau-1)+h(\tau-1)], \tau_l \mid \tau-1, \tau_l\tau_e \nmid \tau-1. \end{cases}$$

Summing up over τ_0, we further get

$$\begin{aligned} &\left\|\boldsymbol{u}^{(k\tau_l\tau_e+\tau_0)}-\tilde{\boldsymbol{v}}^{(k\tau_l\tau_e+\tau_0)}\right\| \\ &=\frac{\delta}{\beta}\big[\left(1+\eta\beta\right)^{\tau_0}-1\big]-\eta\delta\tau_0 \\ &\quad -\eta\left(\delta-\Delta\right)\Big[\frac{1}{2}R_{\tau_0-1}\left(1+R_{\tau_0-1}\right)\tau_l+\sum_{r=1}^{R_{\tau_0-1}}h\left(r\tau_l\right)\Big]. \quad (19) \end{aligned}$$

Taking $\tau_0=\tau_l\tau_e, k=k$–1 into (19), and denoting $H(\tau_l,\tau_e)=\sum_{r=1}^{\tau_e-1}h\left(r\tau_l\right)$, we finally get

$$\begin{aligned} U_k \leq &\frac{\delta}{\beta}\big[\left(1+\eta\beta\right)^{\tau_l\tau_e}-1\big]-\tau_l\tau_e\eta\delta \\ &-\eta\left(\delta-\Delta\right)\Big[\frac{1}{2}\tau_e\left(\tau_e-1\right)\tau_l+H\Big]. \end{aligned}$$

Appendix C
Proof of Theorem 2

In the mobile scenario, we denote the virtual edge mobility model as $\boldsymbol{u}_n^{(\tau)}\triangleq\sum_{m\in\mathcal{E}_n}\alpha_{m,n}\boldsymbol{w}_m^{(\tau)}$, where the vehicle set and the aggregation weights vary in time because of mobility of vehicles. We first introduce two lemmas.

Lemma 7. *For the mobility case, Eq.* (15) *holds for the virtual cloud model if we substitute $\boldsymbol{u}_n^{(\tau-1)}$ by $\hat{\boldsymbol{u}}_n^{(\tau-1)}$.*

The proof is similar to Lemma 3.

Lemma 8. *Denote $\tau=k\tau_l\tau_e+\tau_0, \tau_0\in(0,\tau_l\tau_e]$, and we have the following for the virtual edge mobility model:*

$$\left\|\hat{\boldsymbol{u}}_n^{(\tau)}-\tilde{\boldsymbol{v}}^{(\tau)}\right\| \leq \frac{\delta_n^{(\tau)}}{\beta}[(1+\eta\beta)^{\tau_0}-1]-\eta\tau_0\Big(\delta_n^{(\tau)}-\Delta_n^{(\tau)}\Big). \quad (20)$$

This is a simple extension of (18) if we consider a no-mobility training procedure where vehicles follow the topology of τ-th iteration.

Taking (20) and (16) into (15), and summing up over τ_0, we get

$$\begin{aligned} \left\|\boldsymbol{u}^{(\tau)}-\tilde{\boldsymbol{v}}^{(\tau)}\right\| \leq &\eta\delta\sum_{t=1}^{\tau_0-1}\big[\left(1+\eta\beta\right)^t-1\big] \\ &-\sum_{t=1}^{\tau_0-1}\sum_{r=1}^{R_{t-1}}\Big[r\tau_l\eta\left(\delta-\Delta^{(k\tau_l\tau_e+r\tau_l)}\right)\Big]. \end{aligned}$$

Let $\tau_0=\tau_l\tau_e$ and $k=k-1$, and we obtain

$$\begin{aligned} U_{k,\text{mob}} \leq &\frac{\delta}{\beta}[(1+\eta\beta)^{\tau_l\tau_e}-1]-\tau_l\tau_e\eta\delta \\ &-\eta\tau_l\Big[\frac{1}{2}\tau_e\left(\tau_e-1\right)\delta-\sum_{j=1}^{\tau_e-1}j\Delta^{[k\tau_e+j]}\Big]. \end{aligned}$$

Appendix D
Proof of Proposition 2

To prove the proposition, we first introduce a lemma.

Lemma 9. *[55]Suppose transition probability matrix $\boldsymbol{Q}$ satisfies $\|\lambda_*\|<1$ and is diagonalizable. Then there is a unique stationary distribution π, and given an initial distribution μ_0, the k-th distribution $\mu_k=\mu_0\boldsymbol{Q}^k$ satisfies*

$$\|\mu_k\left(x\right)-\pi\left(x\right)\| \leq l_x\|\lambda_*\|^j,$$

where $\mu_k\left(x\right)$ denotes the x-th element of μ_k.

Based on the lemma, let $\boldsymbol{Q}^{[j]}=[\boldsymbol{s}_1^{[j]};\boldsymbol{s}_2^{[j]},...;\boldsymbol{s}_C^{[j]}]$, and the evolution of $\boldsymbol{Q}^{[j]}$ can be regarded as C separate distributions that converge to steady distribution. Assume the global dataset is balanced, and thus we have $\|\boldsymbol{s}_c^{[j]}\|_1=\|\boldsymbol{s}_{c'}^{[j]}\|_1$ for all c,c', and we can calculate them by

$$\sum_{c=1}^{C}\|\boldsymbol{s}_c^{[j]}\|_1=\sum_{n=1}^{N}\|\boldsymbol{q}_n^{[j]}\|_1=N.$$

So $\|\boldsymbol{s}_c^{[j]}\|_1=\frac{N}{C}$ for all c. Then we define $\tilde{\boldsymbol{s}}_c^{[j]}=\frac{C}{N}\boldsymbol{s}_c^{[j]}$ to make it a regularized distribution, and take it into (9) to get

$$\|\tilde{\boldsymbol{s}}_c^{[j]}\left(n\right)-\frac{1}{N}\| \leq l_{n,c}\|\lambda_*\|^j, \quad n=1,2,\cdots,N$$

Then we can bound the probability divergence by

$$\begin{aligned} \|\boldsymbol{q}-\boldsymbol{q}_n^{[j]}\|_1 &= \sum_{c=1}^{C}\|\boldsymbol{s}_c^{[j]}\left(n\right)-\frac{1}{C}\|=\sum_{c=1}^{C}\|\frac{N}{C}\tilde{\boldsymbol{s}}_c^{[j]}\left(n\right)-\frac{1}{C}\| \\ &= \frac{N}{C}\sum_{i=1}^{C}\|\tilde{\boldsymbol{s}}_c^{[j]}\left(n\right)-\frac{1}{N}\| \leq NL_n\|\lambda_*\|^j, \end{aligned}$$

where $L_n=\sum_{c=1}^{C}l_{n,c}$.

References

[1] T. Chen, J. Yan, Y. Sun, S. Zhou, D. Gunduz, and Z. Niu, "Data-heterogeneous hierarchical federated learning with mobility," in *35th Int. Teletraffic Congr. (ITC-35)*, Torino, Italy, Sep. 2023, pp. 1–5.

[2] A. M. Elbir, B. Soner, S. Çöleri, D. Gündüz, and M. Bennis, "Federated learning in vehicular networks," in *IEEE Int. Mediterranean Conf. on Communications and Networking (MeditCom)*, Athens, Greece, Sep. 2022, pp. 72–77.

[3] Y. Sun, B. Xie, S. Zhou, and Z. Niu, "MEET: Mobility-Enhanced Edge inTelligence for smart and green 6G networks," *IEEE Commun. Mag.*, vol. 61, no. 1, pp. 64–70, Jan. 2023.

[4] K. Tan, D. Bremner, J. Le Kernec, and M. Imran, "Federated machine learning in vehicular networks: A summary of recent applications," *in Int. Conf. on UK-China Emerging Technologies (UCET)*, Glasgow, UK, Aug. 2020, pp. 1–4.

[5] L. Liang, H. Ye, and G. Y. Li, "Toward intelligent vehicular networks: A machine learning framework," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 124–135, Feb. 2019.

[6] D. Ye, R. Yu, M. Pan, and Z. Han, "Federated learning in vehicular edge computing: A selective model aggregation approach," *IEEE Access*, vol. 8, pp. 23920–23935, Jan. 2020.

[7] J. Posner, L. Tseng, M. Aloqaily, and Y. Jararweh, "Federated learning in vehicular networks: Opportunities and solutions," *IEEE Netw.*, vol. 35, no. 2, pp. 152–159, Mar. 2021.

[8] H. Zhou, Y. Zheng, H. Huang, J. Shu, and X. Jia, "Toward robust hierarchical federated learning in Internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 5, pp. 5600–5614, May 2023.

[9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artificial Intelligence and Statistics (AIStats)*, Apr. 2017.

[10] 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.

[11] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1146–1159, Feb. 2020.

[12] F. Yin, Z. Lin, Q. Kong, Y. Xu, D. Li, S. Theodoridis, and S. R. Cui, "Fedloc: Federated learning framework for data-driven cooperative localization and location data processing," *IEEE Open J. Signal Process.*, vol. 1, pp. 187–215, Nov. 2020.

[13] L. Wang, X. Wu, Y. Zhang, X. Zhang, L. Xu, Z. Wu, and A. Fei, "Deepadain-net: Deep adaptive device-edge collaborative inference for augmented reality," *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 5, pp. 1052-1063, Sep. 2023.

[14] 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," in *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3640-3653, Dec. 2021.

[15] Y. Sun, W. Shi, X. Huang, S. Zhou and Z. Niu, "Edge learning with timeliness constraints: Challenges and solutions," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 27–33, Dec. 2020.

[16] D. Gündüz, D. B. Kurka, M. Jankowski, M. M. Amiri, E. Ozfatura, and S. Sreekumar, "Communicate to learn at the edge," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 14–19, Dec. 2020.

[17] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief, "Communication-efficient edge AI: Algorithms and systems," *IEEE Commun. Surveys & Tutorials*, vol. 22, no. 4, pp. 2167–2191, Jul. 2020.

[18] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, Jun. 2020, pp. 1–6.

[19] M. S. H. Abad, E. Ozfatura, D. Gunduz, and O. Ercetin, "Hierarchical federated learning across heterogeneous cellular networks," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, May 2020, pp. 8866–8870.

[20] L. Wang, J. Zhang, J. Chuan, R. Ma, and A. Fei, "Edge intelligence for mission cognitive wireless emergency networks," *Wireless Commun.*, vol. 27, no. 4, pp. 103–109, Aug. 2020.

[21] G. Ayache, V. Dassari, and S. El Rouayheb, "Walk for learning: A random walk approach for federated learning from heterogeneous data," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 929–940, Apr. 2023.

[22] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 453–467, Jan. 2021.

[23] L. Wang, S. Xu, X. Wang, and Q. Zhu, "Addressing class imbalance in federated learning," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2021, vol. 35, pp. 10165–10173.

[24] X. Lu, L. Xiao, T. Xu, Y. Zhao, Y. Tang, and W. Zhuang, "Reinforcement learning based PHY authentication for VANETs," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3068–3079, Mar. 2020.

[25] W. Xu, H. Wang, Z. Lu, C. Hua, N. Cheng, and S. Guo, "Mobile collaborative learning over opportunistic internet of vehicles," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 3187–3199, Apr. 2023.

[26] 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 Trans. Wireless Commun.*, vol. 21, no. 10, pp. 8441–8458, Oct. 2022.

[27] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 215–233, Sep. 2003.

[28] X. Ma, J. Zhu, Z. Lin, S. Chen, and Y. Qin, "A state-of-the-art survey on solving non-iid data in federated learning," *Future Gener. Comput. Syst.*, vol. 135, pp. 244–258, Oct. 2022.

[29] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-iid data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, Nov. 2021.

[30] 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.

[31] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data," in *Neural Inform. Process. Syst. Workshop*, Montréal, Canada, Dec. 2018.

[32] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Int. Conf. on Machine Learning*, Jul. 2020, pp. 5132–5143.

[33] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proc. Mach. Learn. Syst. (MLSys)*, vol. 2, pp. 429–450, Mar. 2020.

[34] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," in *Int. Conf. on Learning Representations (ICLR)*, May 2021.

[35] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *Advances in Neural Inform. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 4424—4434.

[36] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.

[37] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *IEEE Trans. Inf. Theory*, vol. 68, no. 12, pp. 8076-8091, Dec. 2022.

[38] D. Wang, N. Zhang, and M. Tao, "Adaptive clustering-based model aggregation for federated learning with imbalanced data," in *Int. Workshop on Signal Process. Advances in Wireless Commun. (SPAWC)*, Lucca, Italy, Nov. 2021, pp. 591–595.

[39] J. Shen, N. Cheng, X. Wang, F. Lyu, W. Xu, Z. Liu, K. Aldubaikhy, and X. Shen, "Ringsfl: An adaptive split federated learning towards taming client heterogeneity," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 5462-5478, May 2024.

[40] J. Zhang, Z. Li, B. Li, J. Xu, S. Wu, S. Ding, and C. Wu, "Federated learning with label distribution skew via logits calibration," in *Int. Conf. on Machine Learning*, Baltimore, MD, USA, Jul. 2022, pp. 26311–26329.

[41] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[42] M. Duan, D. Liu, X. Chen, Y. Tan, J. Ren, L. Qiao, and L. Liang, "Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications," in *Int. Conf. on Comput. Design (ICCD)*, Washington DC, USA, Nov. 2019, pp. 246–254.

[43] 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," *Advances in Neural Inform. Process. Syst.*, Dec. 2021, pp. 5972–5984.

[44] Y. Deng, F. Lyu, J. Ren, Y. Zhang, Y. Zhou, Y. Zhang, and Y. Yang, "Share: Shaping data distribution at edge for communication-efficient hierarchical federated learning," in *Int. Conf. on Distributed Comput. Syst. (ICDCS)*, Jul. 2021, pp. 24–34.

[45] 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, Apr. 2022.

[46] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[47] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *Proc. of the Int. Conf. on Learning Representations*, New Orleans, LA, USA, May 2019.

[48] J. Wang, S. Wang, R.-R. Chen, and M. Ji, "Demystifying why local aggregation helps: Convergence analysis of hierarchical sgd," in *Proc. of the AAAI Conf. on Artificial Intell.*, vol. 36, pp. 8548–8556, 2022.

[49] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Hierarchical federated learning with quantization: Convergence analysis and system design," *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 2-18, Jan. 2023.
[50] Y. Cai, W. Xi, Y. Shen, Y. Peng, S. Song, and J. Zhao, "High-efficient hierarchical federated learning on non-iid data with progressive collaboration," *Future Generation Comput. Systems*, vol. 137, pp. 111–128, 2022.
[51] 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 Trans. Parallel Distrib. Syst.*, vol. 34, no. 5, pp. 1560-1579, May 2023.
[52] R. M. Gray *et al.*, "Toeplitz and circulant matrices: A review," *Found. and Trends® in Comm. and Inform. Theory*, vol. 2, no. 3, pp. 155–239, 2006.
[53] P. A. Lopez et al., "Microscopic Traffic Simulation using SUMO," in *Proc. IEEE Intell. Transport. Syst. Conf. (ITSC)*, Maui, USA, Nov. 2018.
[54] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," in *Int. Conf. on Learning Representations (ICLR)*, May 2021.
[55] J. S. Rosenthal, "Convergence rates for markov chains," *Siam Review*, vol. 37, no. 3, pp. 387–405, 1995.
[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceed. IEEE Conf. on Comput. Vision and Pattern Recognition*, 2016, pp. 770–778.
[57] Y. Wang, Q. Shi, and T.-H. Chang, "Why batch normalization damage federated learning on non-iid data?" *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 2023, doi: 10.1109/TNNLS.2023.3323302.
[58] T. Chen, J. Yan, Y. Sun, S. Zhou, D. Gunduz, and Z. Niu, "Mobility accelerates learning: Convergence analysis on hierarchical federated learning in vehicular networks," *arXiv preprint arXiv:2401.09656*, 2024.



Tan Chen (S'24) Tan Chen received the B.S. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2021. He is currently working toward the Ph.D. degree with the Network Integration for Ubiquitous Linkage and Broadband Laboratory (Niulab), Department of Electronic Engineering, Tsinghua University. His research interests include federated learning, vehicular networks and edge intelligence.



Jintao Yan (S'23) Jintao Yan received the B.S. degree from the University of Electronic Science and Technology of China (UESTC) in 2022. He is currently working toward the Ph.D. degree with the Network Integration for Ubiquitous Linkage and Broadband Laboratory, Department of Electronic Engineering, Tsinghua University. His research interests include edge intelligence, vehicular networks, cooperative perception and federated learning and optimization theory.



Yuxuan Sun (S'18-M'20) received the B.S. degree in telecommunications engineering from Tianjin University, Tianjin, China, in 2015, and the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2020. From 2018 to 2019, she was a Visiting Student at the Department of Electrical and Electronic Engineering, Imperial College London, UK. From 2020 to 2022, she was a Post-Doctoral Researcher at the Department of Electronic Engineering, Tsinghua University, and a Visiting Researcher at Imperial College London. In July 2022, she joined Beijing Jiaotong University, Beijing, China, where she is now an Associate Professor at the School of Electronic and Information Engineering. She served as the Assistant to the Editor-in-Chief of IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING from 2020 to 2022. Her research interests lie in the areas of edge computing, edge intelligence, and task-oriented communications.



Sheng Zhou (S'06-M'12-SM'24) received the B.E. and Ph.D. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2005 and 2011, respectively. In 2010, he was a Visiting Student with the Wireless System Lab, Department of Electrical Engineering, Stanford University, Stanford, CA, USA. From 2014 to 2015, he was a Visiting Researcher with the Central Research Lab, Hitachi Ltd., Japan. He is currently an Associate Professor with the Department of Electronic Engineering, Tsinghua University. His research interests include cross-layer design for multiple antenna systems, mobile edge computing, vehicular networks, and green wireless communications. He received the IEEE ComSoc Asia–Pacific Board Outstanding Young Researcher Award in 2017, and IEEE ComSoc Wireless Communications Technical Committee Outstanding Young Researcher Award in 2020.



Deniz Gündüz (S'03-M'08-SM'13-F'22) received the B.S. degree in electrical and electronics engineering from METU, Turkey in 2002, and the M.S. and Ph.D. degrees in electrical engineering from NYU Tandon School of Engineering in 2004 and 2007, respectively. In 2012, he joined the Electrical and Electronic Engineering Department at Imperial College London, UK, where he is currently a Professor of Information Processing, and serves as the deputy head of the Intelligent Systems and Networks Group. In the past, he held positions at the University of Modena and Reggio Emilia (part-time faculty member, 2019-22), University of Padova (visiting professor, 2018, 2020), Centre Tecnologic de Telecomunicacions de Catalunya (CTTC) (research associate, 2009-12), Princeton University (postdoctoral researcher, 2007-09, visiting researcher, 2009-11) and Stanford University (research assistant professor, 2007-09).

His research interests lie in the areas of communications, information theory, machine learning and privacy. Dr. Gündüz is a Fellow of the IEEE. He is an elected member of the IEEE Signal Processing Society Signal Processing for Communications and Networking (SPCOM) and Machine Learning for Signal Processing (MLSP) Technical Committees. He serves as an area editor for the IEEE Transactions on Information Theory and IEEE Transactions on Communications. He is the recipient of the IEEE Communications Society - Communication Theory Technical Committee (CTTC) Early Achievement Award in 2017, Starting (2016), Consolidator (2022) and Proof-of-Concept (2023) Grants of the European Research Council (ERC), and has co-authored several award-winning papers, including the IEEE Communications Society - Young Author Best Paper Award (2022), and the IEEE International Conference on Communications Best Paper Award (2023). He received the Imperial College London - President's Award for Excellence in Research Supervision in 2023.



Zhisheng Niu (M'98-SM'99-F'12) graduated from Beijing Jiaotong University, China, in 1985, and got his M.E. and D.E. degrees from Toyohashi University of Technology, Japan, in 1989 and 1992, respectively. From 1992 to 1994, he worked for Fujitsu Laboratories Ltd., Japan, and in 1994 joined with Tsinghua University, Beijing, China, where he is now a Professor at the Department of Electronic Engineering. His major research interests include queueing theory, traffic engineering, mobile Internet, radio resource management of wireless networks, and green communication and networks.

Dr. Niu has been serving IEEE Communications Society since 2000 as Chair of Beijing Chapter (2000-2008), Director of Asia-Pacific Board (2008-2009), Director for Conference Publications (2010-2011), Chair of Emerging Technologies Committee (2014-2015), Director for Online Contents (2018-2019) and currently the Editor-in-Chief of IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING. He received the Best Paper Award of Asia-Pacific Board in 2013, Distinguished Technical Achievement Recognition Award of Green Communications and Computing Technical Committee in 2018, and Harold Sobol Award for Exemplary Service to Meetings & Conferences in 2019, all from the IEEE Communications Society. He was selected as a distinguished lecturer of IEEE Communications Society (2012-2015) as well as IEEE Vehicular Technologies Society (2014-2018). He is a fellow of both IEEE and IEICE.