Data-Heterogeneous Hierarchical Federated Learning with Mobility

Tan Chen*, Jintao Yan*, Yuxuan Sun†, Sheng Zhou*, Deniz Gündüz‡, Zhisheng Niu*
*Beijing National Research Center for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
†School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
‡Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2BT, UK
Email:{chent21,yanjt22}@mails.tsinghua.edu.cn, yxsun@bjtu.edu.cn,
sheng.zhou@tsinghua.edu.cn, d.gunduz@imperial.ac.uk, niuzhs@tsinghua.edu.cn

***Abstract*—Federated learning enables distributed training of machine learning (ML) models across multiple devices in a privacy-preserving manner. Hierarchical federated learning (HFL) is further proposed to meet the requirements of both latency and coverage. In this paper, we consider a data-heterogeneous HFL scenario with mobility, mainly targeting vehicular networks. We derive the convergence upper bound of HFL with respect to mobility and data heterogeneity, and analyze how mobility impacts the performance of HFL. While mobility is considered as a challenge from a communication point of view, our goal here is to exploit mobility to improve the learning performance by mitigating data heterogeneity. Simulation results verify the analysis and show that mobility can indeed improve the model accuracy by up to 15.1% when training a convolutional neural network on the CIFAR-10 dataset using HFL.**

I. Introduction

The advent of 5G has revolutionized intelligent vehicles, enabling them to generate and share significant data volumes through vehicle-to-everything (V2X) services [1]. In this context, machine learning (ML) becomes an essential tool thanks to its ability to efficiently analyze vast amounts of data while adapting to the dynamics of the mobile environment [2]–[4]. Conventional ML solutions rely on offloading edge data to cloud servers. Such centralized solutions, however, encounter challenges of limited communication resources and privacy concerns in vehicular networks [5]–[7]. As an alternative, federated learning (FL) has gained popularity for its ability to efficiently utilize communication resources while preserving privacy [8]. Furthermore, to surmount the unstable communication links in cloud-based FL and the limited coverage and vehicle density in edge-based FL [1], hierarchical FL (HFL) is further proposed in [9] and [10]. The goal of HFL is to train a global model of a central cloud server in a federated manner with the help of edge servers. The edge servers are closer to the devices, and aggregate model parameters of the devices in their coverage area, while the global aggregation at the cloud server takes place less frequently, resulting in a trade-off between the training time and the covering vehicles.

In this paper, we consider HFL in a vehicular network (see Fig. 1 for an illustration). Implementing HFL in vehicular networks faces two challenges. Firstly, different vehicles have different routes and may collect data with different statistics (e.g., different classes), resulting in the production of heterogeneous (or non-i.i.d.) data [8], [11]. Heterogeneous data causes the local objective function to diverge from the global objective function, thereby degrading the learning performance [12], [13]. By minimizing the Kullback-Leibler divergence among the data distributions across the edge servers, a user-edge association method is proposed in [14] to reduce the total number of communication rounds. A user-edge association problem is presented in [15], aiming to minimize the parameter difference between the global and optimal models. The connection between the parameter difference and the data distribution difference is then established.

Secondly, unlike in traditional HFL, where the server and clients are fixed, the mobility of vehicles causes the network topology to be constantly changing. The impact of mobility in HFL is examined in [16]. The convergence speed in relation to mobility rate is analyzed, showing that mobility decreases the convergence speed of HFL. An algorithm is then proposed to alleviate the impact of mobility by aggregating local models based on cosine similarity among model parameters. Experiments show the proposed algorithm improves the performance of HFL with heterogeneous data and mobility. However, the authors view mobility as merely a negative factor in training, and ignore vehicles that cross the coverage of edge servers when the edge server aggregates models of vehicles. Contrastively, we think the data on crossing vehicles is important, and thus these vehicles also need to be scheduled.

In reality, mobility has two-fold effects on HFL. On the one hand, the variations in channel quality make it difficult to adapt to channels for reliable communication. On the other hand, mobility promotes the mixing of heterogeneous data, which can potentially improve the learning performance [2].

This work is sponsored in part by the National Key R&D Program of China No. 2020YFB1806605, the Natural Science Foundation of China (No. 62341108, No. 62221001, No. 62022049, No. 62111530197), the Beijing Natural Science Foundation under grant L222044, the Fundamental Research Funds for the Central Universities No. 2022JBXT001, the Talent Fund of Beijing Jiaotong University under grant 2023XKRC030 and Hitachi Ltd.

The work of Deniz Gündüz has been supported by the UK EPSRC (EP/W035960/1) through the CHIST-ERA program (CHIST-ERA-20-SICT-004).

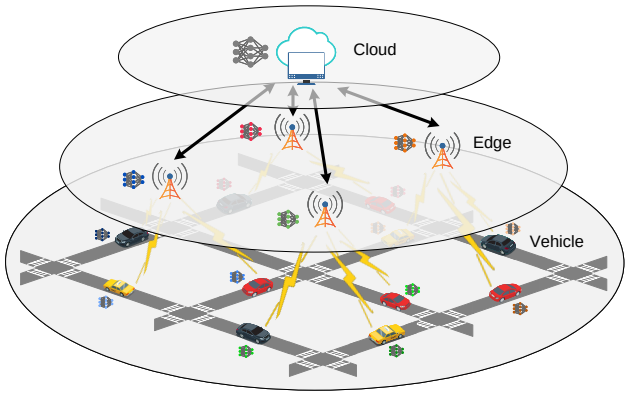


Fig. 1. Illustration of HFL in vehicular networks.

In this work, we investigate the effect of mobility on the performance of HFL. The main contributions are as follows.

- We analyze the convergence upper bound of HFL with respect to data heterogeneity and mobility, showing that mobility influences the data heterogeneity, and thus, the convergence upper bound.
- Through numerical experiments on data-heterogeneous HFL with mobility, we show that mobility can indeed *enhance* the convergence speed and accuracy of HFL.

The rest of this paper is organized as follows. In Section II, we describe the system model, characterize the learning task, and present the training algorithm. In Section III, a convergence analysis is conducted, and a convergence bound is derived for data-heterogeneous HFL with mobility. Section IV presents simulation results and discussions. Finally, Section V concludes the work.

II. HFL SYSTEMS

We consider HFL in vehicular networks. A central cloud server controls several edge servers, which can represent base stations or roadside units. Each edge server is static, and covers a limited area of streets used by vehicles. Vehicles move on the streets stochastically and occasionally cross the coverage of an edge server. We assume that the cloud server has a large enough coverage, so all the vehicles are always within its coverage. Denote the cloud server as c, and assume that there are N edge servers and M vehicles. In practice, M is much larger than N. Vehicles want to 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 model parameters $\boldsymbol{w}$. 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 $\min_{\boldsymbol{w}} F(\boldsymbol{w})$.

A. Mob-HierFAVG

The HierFAVG algorithm in [9] provides a solution to the HFL problem. We build our solution upon HierFAVG by adding the mobility factor.

We denote the gradient descent process on a batch of samples at each vehicle as a local update. We maintain an iteration enumerator τ shared by all nodes in the system, recording the number of local updates each vehicle has carried out in total. A synchronized system is assumed, so τ is tracked by all the vehicles. We denote the model parameters 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.

We assume vehicles sample data before training, and the dataset carried by each vehicle does not change during training. All the participants initialize a global model $\boldsymbol{w}^{[0]}$. The training procedure is composed of the following stages:

1) local update: During training, vehicles repeat updating the model with its local data. For each local update, vehicle m performs stochastic gradient descent using its local data by $\tilde{\boldsymbol{w}}_m^{(\tau)} = \boldsymbol{w}_m^{(\tau-1)} - \eta \nabla f_m\left(\boldsymbol{w}_m^{(\tau)}, \boldsymbol{\xi}_m^{(\tau)}\right)$, 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 $\mathcal{D}_m$.

2) edge aggregation: The edge servers aggregate vehicle models every τ_l local updates. During an edge aggregation, each edge server collects models from vehicles, aggregates them, and distributes the aggregated model back to the vehicles. Assume that each vehicle is associated with only the nearest edge server, and each edge server aggregates the models of all the vehicles within its coverage, including the crossing vehicles. Let $\mathcal{E}_n^{(\tau)}$ represent the vehicle set connected to the n-th edge server at iteration τ, and the aggregation process of the n-th edge 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 $\alpha_{m,n}^{(\tau)} \triangleq \frac{|\mathcal{D}_m|}{\sum_{m' \in \mathcal{E}_n^{(\tau)}} |\mathcal{D}_{m'}|}$.

3) cloud aggregation: The cloud server aggregates edge models every τ_e edge aggregations. Similarly, the cloud aggregation is expressed as $\boldsymbol{w}^{(\tau)} = \sum_n \theta_n^{(\tau)} \tilde{\boldsymbol{w}}_{e,n}^{(\tau)}$, where $\theta_n^{(\tau)} \triangleq \frac{\sum_{m \in \mathcal{E}_n^{(\tau)}} |\mathcal{D}_m|}{\sum_{m=1}^{M} |\mathcal{D}_m|}$.

Following the stages above, the evolution of the m-th local model is then 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.

B. Data Distribution

Since the data is sampled before training, the initial data distribution influences the distribution of training data. We mainly consider these three typical scenarios:

(1) *i.i.d.* The learning tasks are homogeneous for all vehicles, and each vehicle owns a dataset of the same distribution.

(2) *local non-i.i.d.* The learning tasks are heterogeneous for vehicles, such as trajectory prediction and edge caching. For these tasks, vehicles own datasets of different distributions.

(3) *edge non-i.i.d.* The learning tasks are relevant to the environment and thus have a spatial correlation, such as intersection management, beam selection, and channel estimation. In the third scenario, the data distributions are non-i.i.d. across edges, but i.i.d. across the vehicles associated with the same edge server at the time of data collection.

III. Convergence Analysis

A. Definitions

We quantify data heterogeneity by the gradient difference.

Definition 1. *Define δ_m and $\Delta_n^{(\tau)}$ as*

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

where $F_{e,n}^{(\tau)}(\boldsymbol{w}) = \sum_{m\in\mathcal{E}_n^{(\tau)}} \alpha_{m,n}^{(\tau)} f_m(\boldsymbol{w})$. Here, δ_m and $\Delta_n^{(\tau)}$ represent the upper bound on the local gradient difference between the m-th vehicle and the cloud server, and the upper bound on the edge gradient difference between the n-th edge server and the cloud server at the τ-th local iteration, respectively.

Furthermore, the edge-level local gradient difference is represented by $\delta_n^{(\tau)} = \sum_{m\in\mathcal{E}_n^{(\tau)}} \alpha_{m,n}^{(\tau)}\delta_m$, and the cloud-level local gradient difference and edge gradient difference by $\delta = \sum_m \alpha_m\delta_m$ and $\Delta^{(\tau)} = \sum_n \theta_n^{(\tau)}\Delta_n^{(\tau)}$, respectively, where $\alpha_m \triangleq \frac{|\mathcal{D}_m|}{|\mathcal{D}|}$.

The definition indicates that the local gradient difference is not time-varying, while the edge gradient difference is time-varying. This is because the local datasets remain unchanged during training, but the edge datasets are changing due to the mobility of vehicles. Therefore, δ_m and $\Delta_n^{(\tau)}$ represent both the **data heterogeneity** and **mobility** of vehicles.

B. Convergence

In this part, we aim to connect δ_m and $\Delta_n^{(\tau)}$ with the convergence bound of HFL. We first introduce a general convergence bound.

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., $\|f_m(\boldsymbol{w}) - f_m(\boldsymbol{w}')\| \le \rho\|\boldsymbol{w}-\boldsymbol{w}'\|$ for any $\boldsymbol{w},\boldsymbol{w}'$;*
3) *$f_m(\boldsymbol{w})$ is β-smooth, i.e., $\|\nabla f_m(\boldsymbol{w}) - \nabla f_m(\boldsymbol{w}')\| \le \beta\|\boldsymbol{w}-\boldsymbol{w}'\|$ for any $\boldsymbol{w},\boldsymbol{w}'$.*

Definition 2. *Define the following virtual models:*

1) *$\boldsymbol{u}^{(\tau)}$: The virtual cloud model that records the weighted sum of local models at each iteration:*

$$\boldsymbol{u}^{(\tau)} = \sum_m \alpha_m \boldsymbol{w}_m^{(\tau)}.$$

2) *$\boldsymbol{v}^{(\tau)}$: The virtual centralized model that synchronizes with the virtual cloud model periodically. It evolves as*

$$\boldsymbol{v}^{(\tau)} = \begin{cases} \tilde{\boldsymbol{v}}^{(\tau)} = \boldsymbol{v}^{(\tau-1)} - \eta\nabla F\left(\boldsymbol{v}^{(\tau-1)}\right), & \tau_l\tau_e \nmid \tau \\ \boldsymbol{u}^{(\tau)}, & \tau_l\tau_e \mid \tau \end{cases}.$$

Proposition 1. *For any m, assume $f_m(\boldsymbol{w})$ is ρ-Lipschitz,β-smooth and convex, and denote the optimal model as $\boldsymbol{w}^*$. Assume for some $\epsilon \ge 0$, we have*
(1) $\eta \le \frac{1}{\beta}$, (2) $\eta\varphi - \frac{\rho U_k}{\tau_l\tau_e\epsilon^2} > 0$ for all k,
(3) $F\left(\tilde{\boldsymbol{v}}^{(k\tau_l\tau_e)}\right) - F(\boldsymbol{w}^*) \ge \epsilon$,
(4) $F\left(\boldsymbol{w}^{(k\tau_l\tau_e)}\right) \ge \epsilon$ for all k, where we define

$$\varphi = \min_k \frac{1-\frac{\beta\eta}{2}}{\left\|\tilde{\boldsymbol{v}}^{((k-1)\tau_l\tau_e)} - \boldsymbol{w}^*\right\|^2}, \quad \left\|\boldsymbol{u}^{(k\tau_l\tau_e)} - \tilde{\boldsymbol{v}}^{(k\tau_l\tau_e)}\right\| \le U_k.$$

Here, U_k represents an upper bound on the central-cloud difference $\left\|\boldsymbol{u}^{(k\tau_l\tau_e)} - \tilde{\boldsymbol{v}}^{(k\tau_l\tau_e)}\right\|$ at the k-th cloud epoch. Then after $T \triangleq K\tau_l\tau_e$ local updates, we have the following convergence upper bound for HFL

$$F\left(\boldsymbol{w}^{(T)}\right) - F(\boldsymbol{w}^*) \le \frac{1}{T\eta\varphi - \frac{\rho}{\epsilon^2}\sum_{k=1}^{K} U_k}.$$

Proof. *This is an extension of Lemma 2 in [17].*

Proposition 1 indicates that for a fixed cloud epoch K, learning rate η, and aggregation periods τ_l, τ_e, the convergence bound is determined by U_k. Therefore, we then focus on bounding U_k.

Lemma 1. *We have the following for the virtual models:*

$$\left\|\boldsymbol{u}^{(\tau)} - \tilde{\boldsymbol{v}}^{(\tau)}\right\| \le \begin{cases} \left\|\boldsymbol{u}^{(\tau-1)} - \boldsymbol{v}^{(\tau-1)}\right\| + \eta\beta\sum_m \alpha_m \left\|\boldsymbol{w}_m^{(\tau-1)} - \boldsymbol{v}^{(\tau-1)}\right\|, & \\ \qquad\qquad \tau_l \nmid \tau-1 & \\ \left\|\boldsymbol{u}^{(\tau-1)} - \boldsymbol{v}^{(\tau-1)}\right\| + \eta\beta\sum_n \theta_n^{(\tau-1)}\left\|\boldsymbol{u}_n^{(\tau-1)} - \boldsymbol{v}^{(\tau-1)}\right\|, & \\ \qquad\qquad \tau_l \mid \tau-1, \tau_l\tau_e \nmid \tau-1 & \\ 0, \qquad\qquad \tau_l\tau_e \mid \tau-1, & \end{cases} \quad (1)$$

where $\boldsymbol{u}_n^{(\tau)}$ denotes the virtual edge model satisfying

$$\boldsymbol{u}_n^{(\tau)} = \sum_{m\in\mathcal{E}_n^{(\tau)}} \alpha_{m,n}^{(\tau)} \boldsymbol{w}_m^{(\tau)}.$$

Proof. *For the sake of simplicity, we assume $\boldsymbol{\xi}_m^{(\tau)} = \mathcal{D}_m$, since stochastic gradient descent has been proven to be an approximation to deterministic gradient descent [18]. So we substitute $\nabla f_m\left(\boldsymbol{w}_m^{(\tau)}, \boldsymbol{\xi}_m^{(\tau)}\right)$ by $\nabla f_m\left(\boldsymbol{w}_m^{(\tau)}\right)$, and obtain*

$$\begin{aligned} \left\|\boldsymbol{u}^{(\tau)} - \tilde{\boldsymbol{v}}^{(\tau)}\right\| &= \left\|\left[\boldsymbol{u}^{(\tau-1)} - \eta\sum_m \alpha_m \nabla f_m\left(\boldsymbol{w}_m^{(\tau-1)}\right)\right] \right. \\ &\qquad \left. - \left[\boldsymbol{v}^{(\tau-1)} - \eta\nabla F\left(\boldsymbol{v}^{(\tau-1)}\right)\right]\right\| \\ &= \left\|\left[\boldsymbol{u}^{(\tau-1)} - \boldsymbol{v}^{(\tau-1)}\right]\right. \\ &\qquad \left. - \eta\sum_m \alpha_m\left[\nabla f_m\left(\boldsymbol{w}_m^{(\tau-1)}\right) - \nabla f_m\left(\boldsymbol{v}^{(\tau-1)}\right)\right]\right\|. \end{aligned} \quad (2)$$

When $\tau_l \nmid \tau$–1, we derive the first formula in (1) *by the triangle inequality and β-smoothness. When $\tau_l \mid \tau-1$ and $\tau_l\tau_e \nmid \tau-1$, we have $\boldsymbol{w}_m^{(\tau-1)} = \boldsymbol{w}_{m'}^{(\tau-1)}$ if $m, m' \in \mathcal{E}_n^{(\tau-1)}$, and thus*

$$\begin{aligned} (2) = &\left\|\left[\boldsymbol{u}^{(\tau-1)} - \boldsymbol{v}^{(\tau-1)}\right]\right. \\ &\left. - \eta\sum_n \theta_n^{(\tau-1)}\left[\nabla F_n\left(\boldsymbol{u}_n^{(\tau-1)}\right) - \nabla F_n\left(\boldsymbol{v}^{(\tau-1)}\right)\right]\right\|, \end{aligned}$$

so the second formula in (1) *can be similarly derived. When* $\tau_l\tau_e \mid \tau-1$*, we have*

$$(2)=\left\|\left[\boldsymbol{u}^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right]-\eta\left[\nabla F\left(\boldsymbol{u}^{(\tau-1)}\right)-\nabla F\left(\boldsymbol{v}^{(\tau-1)}\right)\right]\right\|=0,$$

where the second equality holds because $\boldsymbol{u}^{(\tau-1)}=\boldsymbol{v}^{(\tau-1)}$ *when* $\tau_l\tau_e \mid \tau-1$ *according to the definition.*

Lemma 2. *Let* $\tau=k\tau_l\tau_e+\tau_0$ *for some* $\tau_0\in(0,\tau_l\tau_e]$*. We have*

$$\left\|\boldsymbol{w}_m^{(\tau)}-\tilde{\boldsymbol{v}}^{(\tau)}\right\|\leq\frac{\delta_m}{\beta}[(1+\eta\beta)^{\tau_0}-1]. \tag{3}$$

Proof. *This is follows from Lemma 3 in [17].*

Lemma 3. *Let* $\tau=k\tau_l\tau_e+\tau_0$ *for some* $\tau_0\in(0,\tau_l\tau_e]$*. We have*

$$\left\|\boldsymbol{u}_n^{(\tau)}-\tilde{\boldsymbol{v}}^{(\tau)}\right\|\leq\frac{\delta_n^{(\tau)}}{\beta}[(1+\eta\beta)^{\tau_0}-1]-\eta\tau_0\left(\delta_n^{(\tau)}-\Delta_n^{(\tau)}\right). \tag{4}$$

Proof. *Similarly to the proof of Lemma 1, we can write the virtual edge model in the recursive form:*

$$\left\|\boldsymbol{u}_n^{(\tau)}-\tilde{\boldsymbol{v}}^{(\tau)}\right\|\leq\left\|\boldsymbol{u}_n^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\|+\eta\beta\sum_{m\in\mathcal{E}_n}\alpha_{m,n}^{(\tau-1)}\left\|\boldsymbol{w}_m^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\|+\eta\Delta_n^{(\tau-1)}.$$

Since $\boldsymbol{v}^{(\tau-1)}=\tilde{\boldsymbol{v}}^{(\tau-1)}$ *when* $\tau_l\tau_e\nmid\tau$*, we substitute* $\left\|\boldsymbol{w}_m^{(\tau-1)}-\boldsymbol{v}^{(\tau-1)}\right\|$ *by* (3)*. Then we can prove Lemma 3 by mathematical induction.*

Theorem 1. *For data-heterogeneous HFL with mobility, the central-cloud difference* $\left\|\boldsymbol{u}^{(k\tau_l\tau_e)}-\tilde{\boldsymbol{v}}^{(k\tau_l\tau_e)}\right\|$ *has upper bound*

$$U_k=r(\tau_l\tau_e,\eta,\delta)-\eta\tau_l[\frac{1}{2}\tau_e\left(\tau_e-1\right)\delta-\sum_{j=1}^{\tau_e-1}j\Delta^{[k\tau_e+j]}], \tag{5}$$

where $r(\tau,\eta,\delta)=\frac{\delta}{\beta}\left[\left(1+\eta\beta\right)^{\tau}-1\right]-\tau\eta\delta$*, and* $\Delta^{[j]}=\Delta^{(j\tau_l)}$*.*

Proof. *The proof follows by substituting* (3) *and* (4) *into* (1)*, and summing up from* $\tau=k\tau_l\tau_e+1$ *to* $\tau=(k+1)\tau_l\tau_e$*.*

This theorem indicates that mobility impacts the convergence upper bound of HFL by changing the edge gradient difference $\Delta^{[j]}$. If $\Delta^{[j]}=\Delta^{[0]}$ for all j, then the theorem degenerates into the case without mobility.

Specifically, considering the data distribution mentioned in Section II, we have $\Delta^{[j]}\approx 0$ for all j in the i.i.d. case, since the mobility of vehicles does not impact the statistics of the datasets within the coverage area of an edge server. However, for the edge non-i.i.d. case, we have $\Delta^{[0]}\gg 0$, since the initial datasets of each edge group is different. As vehicles move during training, the data across edge groups gradually mix up, so $\Delta^{[j]}$ may decrease over iterations. From (1) and (5), $\Delta^{[j]}$ is positively correlated with the convergence upper bound, so mobility may increase the convergence speed in the edge non-i.i.d. case by decreasing the degree of data heterogeneity.

IV. Simulation Results

A. Settings

We consider a city road system on a square grid, where each side of the square is covered by an edge server, e.g., a base station or a road side unit. 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 randomly distributed. We use the Simulation of Urban MObility (SUMO) simulation platform with the Manhattan model. Each side of the road is a meters long, and vehicles travel on the road at a maximum speed of v meters per second (m/s) and slow down when crossing the intersection. The interval between each edge aggregation is assumed to be 1 second. We set the default speed to $v=30$.

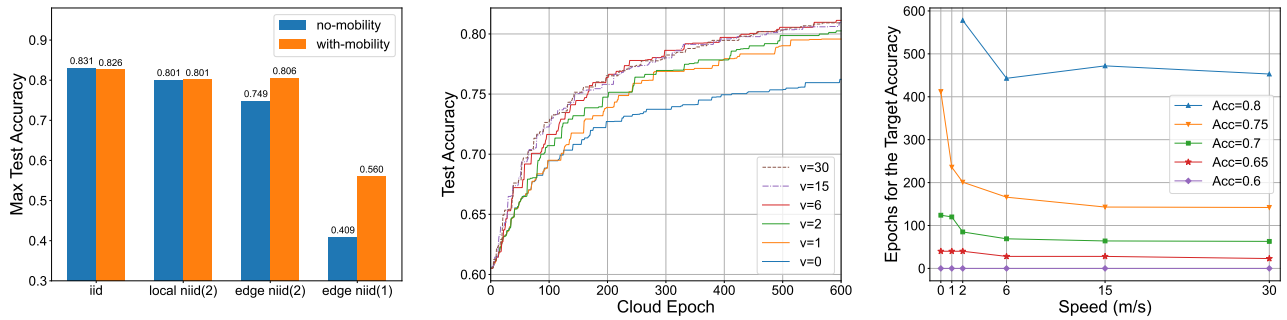
We conduct experiments on the CIFAR-10 dataset, because it is widely used in related works [7], [9], [16], which makes it convenient to compare our performance with other works. To create non-i.i.d. data by partitioning the labels, we choose 40000 training and 8000 test samples in total belonging to 8 classes. For the i.i.d. case, we uniformly divide the training samples into M disjoint subsets, and allocate each subset to one vehicle. For the local non-i.i.d. case, we sort all the data by labels and allocate l classes to each vehicle (denoted by 'local niid(l)' in the following), with all the vehicles holding the same number of samples. For the edge non-i.i.d. case, we first allocate the sorted data to edge servers so that each edge server owns l classes of samples, and then uniformly allocate these samples to the vehicles within their coverage (denoted by 'edge niid(l)' in the following).

We train a 3-layer CNN with a batch size of 20. The learning rate is set to $\eta=0.1$ for all the vehicles with no momentum and learning rate decay. The local and edge epochs are set to $\tau_e=10$ and $\tau_l=6$. We run the Mob-HierFAVG algorithm for 600 cloud epochs.

B. Performance

The performance of HFL under different initial data distributions is shown in Fig 2(a). To address the impact of mobility, we consider two typical speeds for each distribution, $v=0$ (denoted by 'no-mobility') and $v=30$ (denoted by 'with-mobility'). For the i.i.d. case and the local non-i.i.d. case, the maximum test accuracies achieved in the no-mobility and with-mobility scenarios are approximately equal. On the other hand, in the edge non-i.i.d. case, the mobility clearly increases the performance. When $l=2$, mobility improves the accuracy by 5.7% (74.9% to 80.6%); and when $l=1$, mobility improves the accuracy by 15.1% (40.9% to 56.0%). These results are aligned with our conjecture in Section III. Furthermore, they show that for the edge non-i.i.d. case, as the number of classes an edge server holds decreases, the accuracy of HFL decreases, while the accuracy improvement brought by mobility increases.

We then take a further look at the edge non-i.i.d. case. Assume that we start training from a pre-trained model with a test accuracy of 60%. The results in Fig. 2(b) demonstrate that the test accuracy of the scenarios with mobility, i.e., $v>0$, is obviously higher than the one achieved when $v=0$. Besides, Fig. 2(c) illustrates that a higher speed generally results in faster convergence. In particular, it takes only 142 cloud epochs in the $v=30$ case to reach an accuracy of 75%, reduced by 65.5% compared with the $v=0$ case



(a) Comparison of the max test accuracy within 600 cloud epochs for different initial data distribution.

(b) Accuracy on the test dataset for the edge non-i.i.d.(2) case with different vehicle speeds, resuming from the model of 60% test accuracy.

(c) Rounds to reach the target accuracies for the edge non-i.i.d.(2) case, resuming from the model of 60% test accuracy.

Fig. 2. Performance of HFL with different initial data distributions and vehicle speeds.

(412 epochs), and 39.8% compared with the $v = 1$ case (236 epochs). These results indicate that mobility can indeed increase the convergence speed and the final test accuracy of HFL. However, the improvement brought by mobility is also limited. In our experiments, both the final accuracy and the convergence speed typically saturate beyond $v = 6$. This is because the datasets are already sufficiently mixed at this speed, and no further gains can be achieved with even higher mobility.

We also extend the experiments to the scenario with up to 16 edge servers and 80 vehicles. The results show that the conclusions above still hold. Besides, since our convergence analysis is based on a general objective $F(\boldsymbol{w})$, it's promising to conduct similar experiments on the learning tasks in vehicular networks, such as beam selection and channel estimation.

V. Conclusion

In this article, we have investigated the impact of mobility on the convergence of data-heterogeneous HFL. Based on the HFL model with mobility, the convergence analysis of HFL has been conducted, exposing the influence of mobility on the data heterogeneity, and hence, the convergence speed of HFL. Experiments carried out on the CIFAR-10 dataset and the SUMO platform demonstrate the benefits of mobility in the case of edge non-i.i.d. initial data distribution. Specifically, mobility enables an increase in test accuracy by up to 15.1%. Also, the results show that a higher speed results in faster convergence up to a certain value beyond which the achieved accuracy saturates.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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, 2020.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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 *IEEE 41st Int. Conf. on Distributed Computing Systems (ICDCS)*, DC, USA, Jul. 2021, pp. 24–34.

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

[16] 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.

[17] 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.

[18] T. Tuor, S. Wang, K. K. Leung, and K. Chan, "Distributed machine learning in coalition environments: overview of techniques," in *21st Int. Conf. on Information Fusion (FUSION)*, Cambridge, UK, Jul. 2018, pp. 814–821.