Mobility Resilient Vehicular Federated Learning: Enhancing Training Efficiency in Dynamic Environments

Tianao Xiang, Yuanguo Bi, *Member, IEEE,* Lin Cai, *Fellow, IEEE,* Mingjian Zhi

***Abstract*—The vehicular environment presents unique challenges, including massive data generation, stringent latency requirements for safety-critical applications, bandwidth limitations, and intermittent connectivity, which make centralized learning approaches impractical. Vehicular Federated Learning (VFL) enables distributed model training by leveraging local data from connected vehicles, while preserving data privacy and reducing network overhead. However, the dynamic nature of VFL presents several additional challenges. High vehicle mobility and unstable channels lead to inconsistent client participation, while heterogeneous vehicle capabilities result in unbalanced training workloads and competitive resource allocation. These challenges significantly degrade VFL model performance and prolong training periods. In this paper, we propose a Mobility Resilient Vehicular Federated Learning (MR-VFL) scheme, which comprises two key components: an amplification-based adaptive vehicular FL (AVFL) training scheme and a dual-timescale FL scheduler. Specifically, AVFL adapts local training epochs to vehicle capabilities to improve scheduling flexibility and alleviate the impact of insufficient local epochs on model updates, which enhances training efficiency and reduces communication competition. The dual-timescale FL scheduler includes a macro scheduling strategy that optimizes long-term VFL performance based on the correlation between convergence speed and model accuracy, and a Mamba-based real-time scheduler that enhances training efficiency and reduces decision latency in massive vehicles scenarios. Extensive simulations show that MR-VFL effectively mitigates performance degradation due to complex vehicle mobility and heterogeneity, and improves training efficiency.**

***Index Terms*—Federated learning, Internet of Vehicles, client scheduling, reinforcement learning.**

I. Introduction

THE global autonomous driving market will reach $172.15 billion by 2030 [1], driving rapid development of intelligent data-driven applications in transportation systems, such as road safety, driving efficiency, and vehicle infotainment [2]–[4]. These applications generate massive amounts of data on each vehicle, creating valuable opportunities to train and improve machine learning models for enhanced system performance. However, traditional centralized learning approaches exist significant shortcomings when they are applied to vehicular environments.

First, the massive data generation of modern vehicles (25GB per hour per vehicle) makes centralized collection impractical, creating prohibitive communication overhead in bandwidth-constrained networks [5]. Second, transmitting raw data also raises privacy concerns, as vehicular data contains sensitive travel patterns and personal information [6]. Finally, despite the powerful computation and communication capabilities of modern vehicles, centralized learning systems underutilize these distributed resources, overloading the central nodes and creating single points of failures that result in poor robustness for dynamic vehicular environments.

Federated Learning (FL) directly addresses these vehicular-specific challenges by enabling collaborative model training while keeping data local [7], [8]. By transmitting compact model updates rather than raw data, FL reduces communication overhead and preserves privacy by keeping sensitive data local, thereby improving training efficiency and robustness compared to traditional centralized learning systems. Additionally, FL leverages the distributed computational resources of onboard vehicles, which improves training efficiency and distributes the substantial computational workload across the system. These advantages make FL suitable for supporting applications in Vehicular Edge Networks (VENs), including traffic prediction, content caching, platooning, multimodal learning, and security guarantees [9]–[13].

However, implementing FL in vehicular environments presents unique challenges beyond standard FL settings. High vehicle mobility leads to frequent handoffs and limited connection duration, resulting in transmission errors and delays that may prolong FL training time and degrade efficiency [14], [15]. In addition, due to unstable wireless links and limited network resources, vehicular FL (VFL) faces unpredictable client participation patterns and frequent vehicle dropouts, which significantly impact model performance [16]. Meanwhile, vehicles typically have heterogeneous capabilities, including computational power, communication capacity, data distribution, and data quantity [10], [17], leading to substantial variations in training time and model quality. These factors severely degrade FL training efficiency in dynamic vehicular environments.

To address these VFL-specific challenges, we propose a Mobility Resilient Vehicular Federated Learning (MR-VFL)

Tianao Xiang, Mingjian Zhi are with the School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China, and also with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada (e-mail: 2010652@stu.neu.edu.cn; 2110657@stu.neu.edu.cn).

Yuanguo Bi is with the School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China, and also with the Engineering Research Center of Security Technology of Complex Network System, Ministry of Education, Shenyang 110169, China (e-mail: biyuanguo@mail.neu.edu.cn)

Lin Cai is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada (e-mail: cai@ece.uvic.ca).

scheme to explicitly address the inherent heterogeneity of vehicles in terms of computational capabilities, communication resources, and data characteristics. MR-VFL includes two key components: (1) an amplification-based adaptive vehicular FL (AVFL) training scheme that dynamically adjusts local epochs based on vehicle capabilities and sojourn time, enhancing training flexibility while maintaining convergence guarantees; and (2) a dual-timescale scheduling mechanism that optimizes both long-term model performance and immediate training efficiency by intelligently anticipating vehicle mobility patterns. The scheduler formulates client selection as a joint optimization problem solved through reinforcement learning, deriving optimal strategies across macro and micro timescales. The main contributions of this paper are summarized as follows.

1) First, we identify the issues that significantly affect FL performance and then present MR-VFL, which captures dynamic and comprehensive VEN information to flexiblely schedule vehicles and improve VFL training efficiency.
2) Second, we analyze the effect of local epochs on FL convergence and provide a theoretical range for local epoch adjustments that ensures FL convergence. Then, we design an AVFL method, which incorporates an amplification factor to achieve an efficient and flexible training process while alleviating communication competitiveness.
3) Third, we formulate the vehicle scheduling as a long-term optimization problem to maximize VFL training efficiency, which is solved by transforming it into an Markov Decision Process (MDP). We design a Mamba-based Actor-Critic scheduler that implements dual-timescale optimization, balancing immediate scheduling decisions with long-term model performance objectives.
4) Finally, we conduct a comprehensive numerical analysis and extensive experiments. The results demonstrate that MR-VFL alleviates the degradation of the global model in VFL caused by complex and dynamic VENs. Furthermore, the results indicate that MR-VFL outperforms the existing VFL schemes in training efficiency.

The remainder of this paper is organized as follows. Section II reviews the related literature. Section IV presents the settings of VFL and the corresponding models, including the VEN model, vehicle communication, and computation cost models. Section V demonstrates the proposed AVFL scheme and the corresponding convergence analysis. Section VI demonstrates and analyzes the numerical results. Finally, the conclusion is drawn in Section VII.

II. Related Work

While FL has emerged as a promising solution for distributed learning in various domains, its application in vehicular environments presents unique challenges that require specialized solutions. Existing research has explored various aspects of FL in vehicular networks, including client selection, scheduling, and participation patterns.

In recent years, FL has been introduced into VENs to support applications like safety data sharing [18], content caching services [19], autonomous controllers [20], and trajectory prediction [21]. However, the high mobility and heterogeneity of vehicles usually limit the performance of VFL applications. To address these challenges, client selection schemes are designed, and the client participation patterns in FL are studied.

To improve the VFL training efficiency, various client selection schemes are designed. Saputra *et al.* [22] proposed a dynamic economic framework for FL in Internet of Vehicle (IoV) to optimize the selection and payment contracts of smart vehicles, achieving faster convergence and better profits. Samarakoon *et al.* [23] proposed a novel allocation approach jointly considering transmit power and resource for Ultra-Reliable Low-Latency Communication (URLLC) in VENs, which achieves significant reductions in power consumption and extreme events in queues. Xiao *et al.* [24] proposed an FL approach in VENs, considering vehicle position and velocity. It formulates a Min-Max problem to jointly optimize onboard computation, transmission power, and local model accuracy, achieving minimum cost in the worst case. Prathiba *et al.* [25] proposed an FL-empowered computation Offloading and Resource Management (FLOR) framework that optimizes resource allocation and computation offloading across heterogeneous networks in VENs. Hu *et al.* [26] proposed an AutoFL framework to optimize autonomous client participation in federated edge learning, achieving higher model accuracy, significant reduction of energy cost, and enhancing long-term fairness compared to state-of-the-art algorithms.

Furthermore, client scheduling schemes are also proposed in VFL to improve resource utilization and decrease transmission delay. Zhang *et al.* [27] analyzed a communication minimization problem in FL and proposed an optimized probabilistic device scheduling policy is derived in closed-form by solving the approximate communication time minimization problem. Yin *et al.* [28] formulated the client scheduling problem as a Stackelberg leader-follower game and proposed a joint client scheduling and wireless resource allocation algorithm, named SCSBA, to solve this urgent and challenging problem in 6G IoT networks. Hu *et al.* [29] proposed a scheduling policy that jointly considers the channel quality and training data representation of user devices, reducing the bias and variance of the aggregated model updates. Saputra *et al.* [22] proposed a novel dynamic FL-based economic framework for an IoV network to improve the VFL global model accuracy and accelerate the convergence speed. Zhang *et al.* [30] investigated the device scheduling problem in relay-assisted FL systems under the constraints of power consumption and mean squared error (MSE), and then proposed a relay-assisted large-scale FL framework.

In addition to client selection schemes, the impact of different client participation patterns on FL convergence has also been studied. Li *et al.* [26] analyzed the convergence rate of FedAvg on non-iid data under convex and strongly convex conditions and discussed the impact of different client participation schemes on FL. Wang *et al.* [31] presented the unified convergence analysis for FL with arbitrary client participation patterns and designed a generalized FedAvg with amplifies parameter updates. Ruan *et al.* [32] extended the FL paradigm by accommodating devices with flexible participation, presented convergence on non-iid data, and explored the impact of early departures or late arrivals of devices. Liu *et al.* [33] proposed an efficient communication approach in VENs, FedCPF, to improve accuracy and communication efficiency by providing customized local training, partial client participation, and flexible aggregation. Cho *et al.* [34] analyzed FL convergence that client availability follows a natural cyclic pattern and discovered that cyclic client participation can achieve a faster asymptotic convergence rate than vanilla FedAvg, providing valuable insights into the design of client sampling protocols.

However, most approaches fail to comprehensively address the combination of high mobility, intermittent connectivity, heterogeneous capabilities, and resource constraints that characterize vehicular environments. Existing client selection schemes face fundamental limitations in vehicular environments: (1) they assume semi-static participation patterns and focus on resource optimization while ignoring sporadic participation caused by high mobility and ultra-short sojourn time; (2) they reactively handle dropouts rather than proactively leveraging predictable mobility patterns; and (3) they attempt to mitigate heterogeneity through selective participation instead of embracing it as a resource. Meanwhile, existing flexible participation research lacks theoretical foundations for vehicular scenarios, ignoring critical characteristics such as fragile wireless links, mid-training departures, and systematic violations of traditional FL stability assumptions. Therefore, we propose MR-VFL, a mobility-resilient framework that fundamentally differs by introducing amplification-based adaptive training with convergence guarantees for ultra-short participation windows, proactive dual-timescale scheduling that captures spatiotemporal mobility patterns, and transforming vehicular heterogeneity into opportunities for enhanced training efficiency.

III. Observations and Features in VFL

This section presents key observations about vehicular federated learning systems that highlight why specialized approaches are needed beyond standard FL techniques, further motivating the mobility-resilient design for dynamic and complex VENs.

A. Unstable Client Set

Unlike traditional FL where client availability is relatively stable, vehicular networks present unique challenges due to high mobility (vehicles routinely traveling at 30-120 km/h), which significantly impacts both training convergence speed and model performance. When implementing FL systems in VENs, highly mobile vehicles typically exhibit limited sojourn time within the coverage area of a Road Side Unit (RSU), often as short as 10-30 seconds in urban environments and 5-10 seconds on highways. Consequently, selected vehicles may drop from the VFL system before completing their assigned training and communication tasks, leading to intermittent and unreliable participation in the FL process. The VFL system may therefore lose valuable data and degrade model performance, especially when data distributions are heterogeneous across vehicles. Furthermore, the dynamic and unpredictable nature of VENs conditions compounds these challenges, making it difficult to maintain a stable client set for long-term training—a fundamental assumption in most existing FL frameworks.

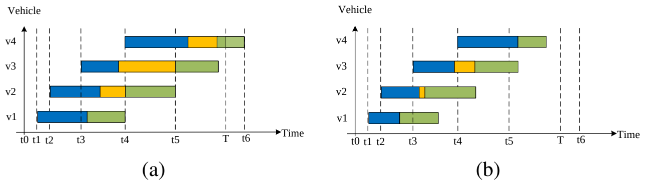


Fig. 1. Impact of competitive model transmissions in VFL. (a) Without scheduling: Multiple vehicles (v1-v4) complete local training (blue) at different times, leading to transmission competition (yellow) and unpredictable waiting times before successful model uploads (green). (b) With intelligent scheduling: Training completion and transmission times are coordinated to minimize competition, reducing overall round time and maximizing resource utilization.

B. Competitive Model Transmissions Among Vehicles

Vehicle-to-infrastructure communications in FL present unique challenges due to complex message structures and unstable wireless links. In practical VENs, communication resources are limited and shared among multiple vehicles. As illustrated in Fig. 1, competition for transmission resources creates significant challenges:

1) Unpredictable Waiting Time: As shown in Fig. 1 (a), uncoordinated training completion leads to simultaneous transmission attempts. Limited bandwidth forces some vehicles (e.g., v2 and v3) to wait (yellow blocks) before transmission (green blocks), unnecessarily extending the training time per VFL round.

2) Bandwidth Contention: When multiple vehicles compete for bandwidth, the network must divide available resources, leading to slower transmission rates for all participants. This bandwidth contention problem worsens in dense urban environments where many vehicles may complete training within short time intervals.

3) Inefficient Resource Utilization: These waiting times and reduced transmission rates extend overall FL round time toward synchronization barrier. This reduces the number of completed rounds and may cause vehicles to depart before transmission, destabilizing the training process. Fig. 1 (b) demonstrates how intelligent scheduling mitigates these issues by coordinating both training and transmission phases. By staggering completion times and optimizing resource allocation, the proposed approach aims to minimize waiting periods and bandwidth competition, enabling more efficient resource utilization and shorter round times.

C. Imbalanced Local Training Efficiency

Vehicles exhibit significant disparities in local training efficiency across data characteristic, computational capability, and network sojourn time. Vehicles with low-quality data may compete with those with high-quality data, increasing average latency and reducing model convergence speed. For example, vehicles with high quality data and high computational power can execute the local epochs quickly while providing valuable model updates. Conversely, vehicles with low-quality data and limited computational power require more time to complete local training and provide low-quality model updates. MR-VFL specifically aims to accommodate these varying capabilities through adaptive training parameters, amplification mechanisms, and heterogeneity-aware scheduling, enabling effective participation across the full spectrum of vehicle resources.

D. Key Ideas of MR-VFL

Based on the observations of the unique challenges in VFL, we propose several novel techniques specifically designed for high mobility vehicular environments.

1. Amplification-based Adaptive Local Training for Mobility-Aware Efficiency. Local training time is the majority of each FL round's duration and is influenced by three key factors: the number of local training epochs, computational capabilities, and potential waiting time. Ideally, a selected vehicle should complete its local epochs and transmit its model update to the aggregator within its sojourn time in the coverage area. However, due to unpredictable departure time and limited resources, if local training takes too long, the selected vehicle may not complete its training process before leaving the coverage area. Unlike existing approaches that use fixed local epoch settings, MR-VFL presents a novel amplification-based adaptive training scheme, which can dynamically adjust local training epochs according to vehicle-specific factors: data quality, predicted sojourn time, and heterogeneous computational capabilities. This approach allows vehicles with limited sojourn time to contribute their updates despite performing fewer local epochs, significantly improving participation rates and training efficiency in highly mobile scenarios.

2. Actor-Critic Based Dynamic Real-Time Scheduling for Extended Client Availability. Beyond optimizing individual vehicle workloads, we address the critical challenge of unstable client availability in vehicular networks. The theoretical analysis shows how this instability severely degrades FL performance. MR-VFL implements a dual-timescale approach: at the macro level, we employ a participation strategy that balances fast initial convergence with final model accuracy; at the micro level, we formulate client selection as a dynamic bin-packing problem solved via actor-critic reinforcement learning. The proposed Mamba-based scheduler captures spatioal and temporal patterns in vehicle mobility, strategically allocating time slots to reduce communication competition while ensuring fair vehicle selection through an exponential penalty mechanism. This approach maintains higher vehicle availability by anticipating mobility patterns rather than reacting to dropouts, improving model performance while reducing training time.

IV. System Model

In this section, we demonstrate the settings of VFL and present the corresponding models, including the VEN, vehicle communication, and computation cost models.

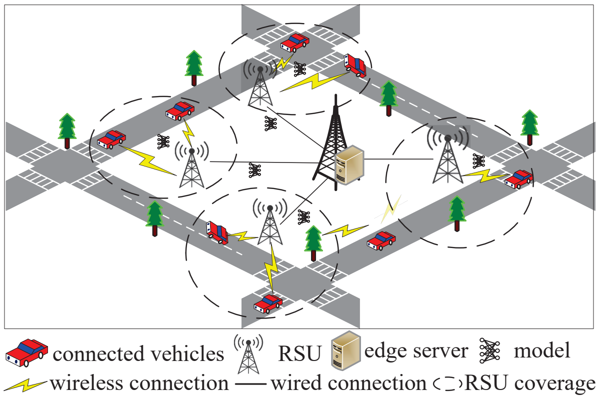


Fig. 2. Architecture of the VFL system.

A. Vehicular Edge Network Model

A typical VEN consists of an edge server, multiple RSUs, and onboard vehicles, as shown in Fig. 2. The edge server performs a distributed ML task by leveraging on-board CPUs and local datasets of moving vehicles. Each vehicle executes a local training process and transmits the model updates to the edge server. RSUs maintain stable connections between the edge server and the vehicles, transmitting models between both sides. Vehicles within this coverage experience stable and low-latency communication links, enabling efficient transmissions of model updates. In contrast, vehicles outside the coverage face multi-hop connections and frequent handoffs, which results in increased packet loss and communication delay. Therefore, the edge server tends to discard vehicles that are far away from it to reduce the training time of per FL round and enhance communication efficiency.

B. Preliminary of VFL

Let $\mathbf{w}_{v,k} \in \mathbb{R}^d$ denote the model parameters for vehicle v at communication round k, where d represents the model dimension. Each vehicle maintains a local dataset $\mathcal{D}_v$. The learning rate $\eta > 0$ controls the step size of model updates. Let E denote the number of local training epochs, where $e \in 1, ..., E$ is the index of local epoch, and $k = 1, \cdots, K$ denotes the FL round. The stochastic gradient estimator $g_n(\mathbf{x})$ provides an unbiased estimate of the true gradient $\nabla F_n(\mathbf{x})$ of the local objective function. The aggregate dataset at iteration t is therefore $\mathcal{D}_t = \mathcal{D}vv \in \mathcal{V}_t$. VFL aims to optimize

$$\min_{\mathbf{w}} f(\mathbf{w}) = \sum_{n=1}^{|\mathcal{V}_t|} q_n F_n(\mathbf{w}), \tag{1}$$

where $q_v \in [0, 1]$ is the aggregation weight for vehicle v with $\sum_{v \in \mathcal{V}_t} q_v = 1$. $F_v(\mathbf{w})$ represents the local empirical loss function of vehicle v. The FL server distributes the global model at each round. Each vehicle $v \in \mathcal{V}_t$ trains its local model E epochs on its local dataset $\mathcal{D}_v$ and obtains the updated model as

$$\mathbf{w}_{v,k+1} = \mathbf{w}_{v,k} - \eta \sum_{e=1}^{E} \nabla F_v(\mathbf{w}_{v,k}), \tag{2}$$

where η is the learning rate. After local training, v sends $\mathbf{w}_{v,k+1}$ to the server, and then the server performs model aggregation to obtain the global model $\mathbf{w}_{k+1}$ in the next global round. The above process repeats until the globally trained model reaches an expected level of precision.

C. Time Consumption Model

1) Sojourn time prediction model: To evaluate the availability of vehicles within the coverage, the edge server needs to predict the sojourn time of vehicles. The sojourn time of vehicle $v \in \mathcal{V}_t$ is represented by T_v^{soj}. The sojourn time is influenced by multiple factors, such as temporal, traffic, spatial, augmented, and personalized aspects, as discussed in [35]. The sojourn time estimation can be modeled by extracting features from these factors as a regression task in the edge server. Each sample's ground truth is denoted by $\mathbf{a} = [a_1, a_2, \ldots, a_N] \in \mathbb{R}_+$, where $a_i = e_i - s_i$ represents the sojourn time equals to the time gap between the arrival and departure time. The dataset consists of samples $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_N]$, with each sample $\mathbf{b}_i \in \mathbb{R}^d$ encoding the route path p_i's trajectory in the road network. We aim to train a model to accurately predict the sojourn time for future unseen data $\mathbf{b}_q \notin \mathbf{B}$.

In sojourn time estimation, the tolerance of the estimation gap varies according to the total sojourn time. Thus, the mean absolute percentage error (MAPE) is a more reasonable metric in this problem. Our target is to minimize MAPE as

$$\min_R \sum_{i=1}^{N} \frac{|a_i - R(\mathbf{b}_i)|}{a_i} + \Omega(R), \tag{3}$$

where $R(\mathbf{b}_i)$ is the sojourn time estimation of the route $\mathbf{b}_i$, the function R is a regression model, and $\Omega(R)$ is the regularization term that controls the complexity of the model R. It can be learned by using proper machine learning models. The problem in Eq. (3) can be addressed by the wide-deep-recurrent learning-based solution to obtain T_v^{soj} [36].

2) Computation time model: Denote vehicle v's computation cycle frequency as $\delta_v(k) \in [\delta_{\min}, \delta_{\max}]$, and the CPU cycles to process per data sample as c_v. The local computation time of v performing local training E epochs in the k^{th} communication round is

$$T_v^{cmp} = \frac{E c_v D_v}{\delta_v(k)}, \tag{4}$$

where D_v is the dataset size of v in bits.

3) Communication time model: In VFL, transmission competition occurs when multiple vehicles simultaneously upload their model updates, leading to bandwidth contention and increased communication delay. For a given vehicle v, the basic transmission rate without competition can be represented as $r_v = B_t \log_2(1 + \frac{p_v h_v}{N_0})$, where B_t denotes the allocated transmission bandwidth at the t-th iteration, p_v represents the transmission power, h_v denotes the channel gain between vehicle v and the server, and N_0 represents the background noise. Given a constant model size ε, the communication time without competition is

$$T_{v,\mathrm{wc}}^{\mathrm{com}} = \frac{\varepsilon}{r_v} = \frac{\varepsilon}{B_t \log_2\left(1 + \frac{p_v h_v}{N_0}\right)}. \tag{5}$$

The unstable nature of wireless links introduces packet loss, characterized by the packet loss ratio λ_{pl}, which represents the probability of a packet being lost during transmissions. Let ε_u denote the size of each packet, with $\frac{\varepsilon}{\varepsilon_u}$ packets required for a complete model transmission. For successful transmissions, each packet must be received correctly with probability $(1 - \lambda_{pl})$. The number of transmission attempts X required for successful model upload follows a negative binomial distribution, representing the number of trials needed until achieving the required number of successes. Therefore, the expected number of packet transmissions is

$$N_v = \frac{\varepsilon}{\varepsilon_u(1 - \lambda_{pl})}. \tag{6}$$

Considering the high vehicle mobility in VFL, the channel gain of vehicle v on the wireless channel can be decomposed as

$$h_v(t) = h_v^{PL} \cdot h_v^{SS}(t) \cdot h_v^{LS}(t), \tag{7}$$

where h_v^{PL} represents path loss, $h_v^{SS}(t)$ represents small-scale fading, and $h_v^{LS}(t)$ represents large-scale shadowing effects. The competition-adjusted transmission rate incorporating both bandwidth sharing and interference effects is

$$r_v^{\mathrm{c}} = \frac{B_t}{|\mathcal{C}_v|} \log_2\left(1 + \frac{p_v h_v \alpha_v}{N_0 + \sum_{u \in \mathcal{C}_v} p_u h_u \alpha_u}\right), \tag{8}$$

where α_v represents the fading coefficient that follows either Rayleigh or Rician distribution depending on the vehicle's environment, and the denominator $|\mathcal{C}_v|$ accounts for the number of transmitters sharing the bandwidth with v, while the sum term in the logarithm represents the interference from the concurrent transmissions.

The total time for vehicle v to participate in one FL communication round, including both computation time T_v^{cmp} and competition-adjusted communication time, is $T_v^{ttl} = T_v^{cmp} + T_v^{com}$. To ensure successful participation, vehicle v needs to complete its training process within its sojourn time T_v^{soj}, and we have

$$T_v^{soj} \geq (1 + \xi) T_v^{ttl}, \tag{9}$$

where $\xi > 0$ is a small safety margin constant that extends the FL training time duration to ensure that vehicle v has sufficient time to complete training even when encountering minor delays or latency fluctuations during the process.

D. Traffic Flow Model

In VFL, the available set of vehicles is usually not constant and stable. Informally, we can divide the vehicles in FL into two categories, which are stable vehicles and temporary vehicles. The stable vehicles means that the vehicles can stay in the edge server coverage and participate in the FL process for multiple rounds. The temporal vehicles are that the vehicles with high mobility leave the area during the current FL round and give up the FL training. We denote the stable vehicle set and temporary vehicle set as $\mathcal{V}_t^{st}$ and $\mathcal{V}_t^{tp}$, and the entire available vehicle set during round t can be denoted by $\mathcal{V}_t = \mathcal{V}_t^{st} \cup \mathcal{V}_t^{tp}$. To capture the dynamic of VENs and quantify

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3607138

the number of $\mathcal{V}_t^{tp}$, a Poisson distribution-based traffic flow model is presented as

$$P(|\mathcal{V}_t^{\rm tp}| = m) = \frac{e^{-\Lambda}\Lambda^m}{m!},, \tag{10}$$

where Λ is a pre-detemined parameter.

E. Problem Formulation

While the macro-scheduling strategy is presented, we can quantify the MR-VFL goal and formulate it as a optimization problem. The goal of the scheduling problem is

$$\max_{\alpha_v, x_v, T_v^s} \sum_{r=1}^{R} \theta_{acc}^r + (T_{syn}^R - T_{syn,a}^R) + \sum_{v\in\mathcal{V}_t} x_v, \tag{11}$$

where θ_{acc}^r is the increase of accuracy after round r and $x_v \in \{0,1\}$ is an indicator that denotes whether v is successfully scheduled. T_{syn}^R is the predetermined maximum time duration for VFL training per round and $T_{syn,a}^R$ is the actual time of communication round R. The goal in Eq. (11) is to maximize the accuracy improvement θ_{acc}^r, the time gap for per FL round $(T_{syn}^R - T_{syn,a}^R)$, and the number of the selected vehicles $\sum_{v\in\mathcal{V}_t} x_v$. In particular, the improvement in accuracy θ_{acc}^r is a short-term metric to capture the performance of the current round. In addition, according to Theorem V.4, the number of successfully scheduled vehicles in the current round $\sum_{v\in\mathcal{V}_t} x_v$ can affect the long-term FL convergence speed and upper bound. Specifically, with a large $\sum_{v\in\mathcal{V}_t} x_v$ in each round, VFL can converge quickly. Furthermore, the time saving of the current round $(T_{syn}^R - T_{syn,a}^R)$ can capture the training efficiency that uses short time to complete the current round.

The first constraint is the sojourn time constraint based on the proposed AVFL scheme, which is formulated as

$$x_v(\alpha_v T_v^{com} + T_v^{cmp} + T_v^{wt}) \le T_v^{soj}, \tag{12}$$

where T_v^{wt} denotes the waiting time of vehicle v due to the competitive communications, α_v is the amplification factor in the AVFL of vehicle v, and $x_v \in \{0,1\}$ is an indicator that denotes whether v is successfully scheduled. Compared to Eq. (9), the waiting time is considered in Eq. (12), which is more rationale because in practical a vehicle is hard to occupy a dedicated communication resource to transmit its model update. Therefore, when two or more vehicles have conflicts on communications, it may prolong the transmission time and degrade the FL efficiency.

Furthermore, the range of amplification α_v in Eq. (12) needs to be determined, and we have

$$\alpha_v \in [1, E/E_v^{low}], \tag{13}$$

where E_v^{low} is the least local epochs for vehicle v and $E_v^{low} \ge 1$. E_v^{low} is determined by the amounts of local data, the distribution of local data, and the round index, which can be formulated as

$$E_v^{\rm low} = G(\text{data distribution}, \text{data amounts}, \text{round index}), \tag{14}$$

where $G(\cdot)$ is a function to quantify the relation between E_v^{low} and the input variables, which can be considered as embedding layers of neural network in practical design. Specifically, when v has a large data amounts, a comprehensive data distribution, or an early period round index, E_v^{low} can be decreased far from the required local epochs E.

The other constraint is about the FL synchronous time, which can be formulated as

$$\begin{cases} T_{\rm syn,a}^R = T_{\rm syn}^R & \text{if } \sum_{v\in\mathcal{V}_t} x_v \le N_R \text{ and } T_{\rm syn,a}^R = T_{\rm syn}^R, \\ T_{\rm syn,a}^R = T_{\rm syn,a}^R & \text{if } \sum_{v\in\mathcal{V}_t} x_v = N_R \text{ at } T_{\rm syn,a}^R. \end{cases} \tag{15}$$

where N_R is the required number of vehicles in round R before synchronizing. Eq. (15) shows a soft synchronous barrier that if the number of the successfully scheduled vehicles is larger than N_R, the synchronous event can be triggered at $T_{syn,a}^R$, otherwise the synchronous event is triggered at the hard synchronous barrier T_{syn}^R.

Let us first define the key variables for the OFDMA resource allocation model. The indicator variable $x_v \in \{0,1\}$ denotes whether vehicle v is selected to transmit its model update, where $x_v = 1$ indicates the vehicle is selected and $x_v = 0$ indicates the vehicle is not selected. The bandwidth allocation is represented by $bd_i \in \{0,1\}$, where $bd_i = 1$ indicates that bandwidth unit i is assigned. According to the OFDMA settings, the communication resource constraints can be expressed as

$$\sum_{b_i\in\mathcal{B}} \sum_{v\in\mathcal{V}_t} x_v b_i T_v^{\rm com} \le T_{\rm syn}^R |\mathcal{B}|, \tag{16}$$

where $\mathcal{B}$ is the entire bandwidth capacity satisfying $\sum_{bd_i\in\mathcal{B}} \sum_{v\in\mathcal{V}_t} x_v bd_i = \mathcal{B}$. Eq. (16) shows the physical communication resource limitation for vehicles scheduled in a single round.

To ensure fair vehicle selection across multiple rounds, we introduce a logarithmic fairness constraint that limits the disparity in selection frequency. This constraint is formulated as

$$\sum_{v\in\mathcal{V}_r} \exp\left(n_v^r\right) \le \Gamma_r, \tag{17}$$

where n_v^r is the cumulative number of times vehicle v has been selected up to round r and Γ_r is a predetermined threshold to control fairness. This exponential formulation creates a progressively stronger penalty for frequently selected vehicles, encouraging the scheduler to distribute participation opportunities more equitably across the available vehicle population while still maintaining system performance.

Besides the above constraints, there is another constraint to guarantee the performance of the proposed FL process, which can be formulated as

$$f(\mathbf{w}_{r-1}) - f(\mathbf{w}_r) \le \phi, \tag{18}$$

where ϕ is a pre-determined parameter to denote the performance threshold to terminate the current FL process.

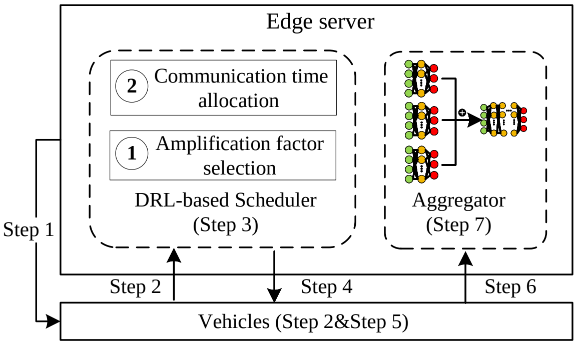

Edge server
2 Communication time allocation
1 Amplification factor selection
DRL-based Scheduler (Step 3)
Aggregator (Step 7)
Step 1
Step 2
Step 4
Step 6
Vehicles (Step 2&Step 5)

Fig. 3. The system overview of MR-VFL.

A joint optimization problem for accuracy improvement and training time is formulated, which can be expressed as

$$\max_{\alpha_v, x_v, T_v^s} \sum_{r=1}^{R} \theta^r_{acc} + (T^R_{syn} - T^R_{syn,a}) + \sum_{v \in \mathcal{V}_r} x_v \tag{19a}$$

$$s.t.\ Eqs.\ (12),\ (13),\ (14),\ (15),\ (16),\ (17),\ \text{and}\ (18),$$

$$T_v^{\text{com}} \leq T^R_{\text{syn}} - T_v^s, \quad \forall v \in \mathcal{V}_r \text{ with } x_v = 1, \tag{19b}$$

where $\mathcal{V}_r$ is the vehicle set at round r, θ^r_{acc} is the increase of accuracy after round r and T_v^s is the scheduled communication time of v. The constraint in Eq. (19b) means that the scheduled time slot of v can make v transmit the model update before the synchronous barrier T^R_{syn}.

V. System design of Mobility Resilient Vehicular Federated Learning Scheme

A. Overview of MR-VFL

In this section, inspired by the observations of VFL, we propose MR-VFL to improve FL performance in highly dynamic vehicular networks. MR-VFL consists of two main components: an amplification based vehicular federated learning method AVFL and a dynamic vehicle scheduler based on deep reinforcement learning. As shown in Fig. 3, the entire MR-VFL process consists of seven steps, as follows.

Step 1:At the beginning of each round, an edge server sends an information collection request to the vehicles within its coverage.

Step 2:Vehicles that receive the request estimate their computation time, communication time, and data quality, and send these information to the edge server along with other vehicle details, such as departure time and current speed.

Step 3:The edge server predicts the sojourn time of each vehicle and determines the available vehicle set based on the collected information.

Step 4:The edge server selects vehicles to participate in the FL process and sends the control information including amplification factors to them based on the decisions of the designed Mamba-based scheduler.

Step 5:The selected vehicles perform local training according to AVFL.

Step 6:The selected vehicles transmit their model updates to the edge server.

Algorithm 1 The procedures of amplification-based VFL

```
Input: learning rate $\eta$, required local epochs $E$, communication round $R$.
Initialize $r_0 \to 0$, $\mathbf{w} \to 0$.
for $r = 0, \cdots, R-1$ do
    $\mathbf{y}_n^r \leftarrow \mathbf{w}_r$
    for all $v \in \mathcal{V}_r$ in parallel do
        Compute $E_v^r$ according to Eq. (21).
        for $e = 0, \cdots, E_v^r$ do
            $\mathbf{y}^r_{v,e+1} \leftarrow \mathbf{y}^r_{v,e} - \eta \nabla g_n^e(\mathbf{w}^e_{v,r})$.
        end for
        $\Delta_v^r \leftarrow \mathbf{y}^r_{v,E_v^r} - \mathbf{w}_r$.
        if $E_v^r < E$ then
            $\beta_v^r = E/E_v^r$.
            Choose amplification factor $\alpha_v^r \leq \beta_v^r$.
            $\Delta_v^r \leftarrow \alpha_v^r \Delta_v^r$.
        end if
        Send $\Delta_v^r$ to Server.
    end for
    $\mathbf{w}_{r+1} = \mathbf{w}_r + \sum_{v=1}^{\mathcal{V}_r} q_v^r \Delta_v^r$.
end for
```

Step 7:When the required number of model updates is reached, or the synchronous limit of the round is reached, the edge server aggregates the received model updates and initiates the next round training.

Steps 1-4 comprise the dynamic vehicle scheduling process, while the remaining steps are the AVFL process. In the proposed MR-VFL scheme, the scheduled vehicles are not required to complete training or upload model updates successfully. Instead, the design prioritizes maximal participation by scheduling the largest feasible number of vehicles for VFL. Note that the amplification factor also determined by the designed scheduler, which can dynamiclly adapt the error caused by the decision-making process.

B. Amplification based Vehicular Federated Learning

In this section, we first analyze the impact of local epochs on FL convergence, and then design AVFL by using an amplification factor to adjust local epochs. Furthermore, we present a theoretical range of local epoch adjustments without degrading the VFL convergence.

1) Amplification-based training method: AVFL aims to make all vehicles complete their FL process within their sojourn time and contribute to the global model, thus alleviating the impact of unstable and unpredictable traffic flow on FL process. We assume that the gradient $g_v^e(\mathbf{x})$ of vehicle v at epoch e obtained by the stochastic gradient descent (SGD) method is an unbiased estimate of the true gradient $\nabla F_v^e(\mathbf{x})$. Thus, the local training process in Eq. (2) can also be expressed as

$$\mathbf{w}_{v,r+1} = \mathbf{w}_{v,r} - \eta \sum_{e=1}^{E} g_v^e(\mathbf{w}_{v,r}). \tag{20}$$

As shown in Eq. (20), the local training is equivalent to estimate $\nabla F_v^e(\mathbf{x})$ by computing $g_v^e(\mathbf{x})$ for E times. Theoretically,

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3607138

even if the number of local epochs $\mathcal{E} \neq E$, the expectation of $\nabla F_n(\mathbf{x})$ during E and $\mathcal{E}$ epochs is equivalent, which is $\frac{1}{E}\sum_{e=1}^{E} g_v^e(\mathbf{w}_{v,r}) \approx \frac{1}{\mathcal{E}}\sum_{e=1}^{\mathcal{E}} g_v^e(\mathbf{w}_{v,r})$ [37].

A vehicle with limited computation capability is difficult to complete the required local training epochs. To address this issue, the vehicle can decrease their local epochs and still obtain a good enough local gradient. We denote the shortened local epochs as $E_v^r \leq E$, where E_v^r can be formulated as

$$E_v^r = \lfloor \frac{(T_v^{soj} - T_v^{com})\delta_v(k)}{c_v D_v} \rfloor, \tag{21}$$

where $T_v^{soj} - T_v^{com}$ denotes the maximizing remained computation time and $\frac{\delta_v(k)}{c_v D_v}$ is the computation cost per epoch.

However, the local update with E_v^r (i.e., $\eta \sum_{e=1}^{E_v^r} g_v^e(\mathbf{w}_{v,r})$) is smaller than that with E (i.e., $\eta \sum_{e=1}^{E} g_v^e(\mathbf{w}_{v,r})$), which may cause insufficient model update and potentially decrease the FL efficiency. To this end, the amplification factor is introduced to compensate the insufficient model update and make it similar to the one with sufficient model update. Thus, the proposed amplification-based training method is

$$\begin{aligned}\mathbf{w}_{v,r+1} &= \mathbf{w}_{v,r} - \alpha_{v,r}\eta \sum_{e=1}^{E_v^r} g_v^e(\mathbf{w}_{v,r}) \\ &\approx \mathbf{w}_{v,r} - \alpha_{v,r}\eta \sum_{e=1}^{E_v^r} \nabla F_v^e(\mathbf{w}_{v,r}),\end{aligned} \tag{22}$$

where $\alpha_{v,r} \geq 1$ is an amplification factor of vehicle v in round r to achieve the sufficient model update.

We summarize the above designs in the Algorithm 1. The FL server distributes the initial model and the required number of local epochs E to the set of vehicles $\mathcal{V}_t$. For those vehicles with enough sojourn time, they can send the model updates to the server after completing E local epochs. For vehicles without enough sojourn time, they compute their own local epochs E_v^t according to Eq. (21) and proceeds with local training. The vehicle then chooses an amplification factor α_v^t to amplify insufficient model updates and further mitigate the effects of insufficient local training. After the amplification process in line 12 - line 14, the vehicles send the model updates to the server and wait for the next round. Finally, the server aggregates the revised model updates to obtain a global model. AVFL dynamically adapts to each vehicle's unique computational resources through Eq. (21), ensuring effective participation regardless of individual limitations.

Computational Complexity Analysis: The computational complexity of Algorithm 1 can be analyzed as follows. For each vehicle v, the local training has complexity $O(E_v^r \cdot |D_v| \cdot d)$, where E_v^r is the number of local epochs, $|D_v|$ is the local dataset size, and d is the model dimension. The model amplification has complexity $O(d)$ per vehicle. On the server side, the model aggregation has complexity $O(|\mathcal{V}_r| \cdot d)$.

Since local training and amplification occur in parallel across vehicles, the training phase complexity is $O(\max_v(E_v^r \cdot |D_v|) \cdot d)$. Combined with sequential aggregation, the overall per-round complexity is $O(\max_v(E_v^r \cdot |D_v|) \cdot d + |\mathcal{V}_r| \cdot d)$. For typical scenarios where the training cost dominates the aggregation cost, this simplifies to $O(\max_v(E_v^r \cdot |D_v|) \cdot d)$. The communication complexity is $O(|\mathcal{V}_r| \cdot d)$ for transmitting model updates.

Importantly, the adaptive approach potentially reduces the effective E_v^t for computation-constrained vehicles, thereby improving efficiency without sacrificing performance.

2) The convergence analysis of AVFL: In this part, we present a convergence analysis of AVFL. We make the following assumptions that are commonly used in the existing works [31].

Assumption 1 (Lipschitz gradient)**.** *For all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and $v \in \{1, \ldots, |\mathcal{V}|\}$*

$$\|\nabla F_v(\mathbf{x}) - \nabla F_v(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y}, v. \tag{23}$$

Assumption 2 (Unbiased stochastic gradient with bounded variance)**.** *For all $\mathbf{x} \in \mathbb{R}^d$ and $v \in \{1, \ldots, |\mathcal{V}|\}$*

$$\mathbb{E}\left[\mathbf{g}_v(\mathbf{x}) | \mathbf{x}\right] = \nabla F_v(\mathbf{x}) \tag{24}$$

$$\mathbb{E}\left[\|\mathbf{g}_v(\mathbf{x}) - \nabla F_v(\mathbf{x})\|^2 \middle| \mathbf{x}\right] \leq \sigma^2, \forall \mathbf{x}, v. \tag{25}$$

Assumption 3 (Bounded gradient divergence)**.** *For all $\mathbf{x} \in \mathbb{R}^d$ and $v \in \{1, \ldots, |\mathcal{V}|\}$*

$$\|\nabla F_v(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq d^2, \forall \mathbf{x}, v. \tag{26}$$

Based on the above assumptions, the convergence theorem of AVFL is presented as follows.

Theorem V.1. *Let $v \in \{1, \ldots, |\mathcal{V}|\}$ denote the vehicle index, and $q_v^r \in [0, 1]$ denote the aggregation weight of vehicle v in the r-th round. We have $\sum_{v=1}^{|\mathcal{V}|} q_v^r = 1$ and $\sum_{v=1}^{|\mathcal{V}|} (q_v^r)^2 \leq \rho^2 \leq 1$. Combining with Assumptions 1, 2, and 3 hold, we have*

$$\begin{aligned}&\frac{1}{R}\sum_{r=1}^{R} \|\nabla f(\mathbf{w}_r)\|^2 \\ &\leq \frac{2\tau}{\eta R E}\mathbb{E}[f(\mathbf{w}_r) - f(\mathbf{w}_0)] + \frac{2\tau d^2}{RE}\sum_{r=1}^{R}\sum_{v=1}^{|\mathcal{V}|}\beta_v^r (q_v^r)^2 \\ &\quad + \frac{2\tau\sigma^2}{RE}\left(5.5 + \frac{\eta L E}{2} + 2b + \frac{2}{b}\right)\sum_{r=1}^{R}\sum_{v=1}^{|\mathcal{V}|}\frac{(q_v^r)^2(\alpha_v^r)^2}{\beta_v^r} \\ &\quad + \frac{\tau d^2}{RE}(\eta L E + 3 + \frac{4}{b})(1 + c)\sum_{r=1}^{R}\sum_{v=1}^{|\mathcal{V}|}\frac{q_v^r \alpha_v^r}{\beta_v^r},\end{aligned}$$

where b and c are constants, η is the learning rate, and β_v^e denotes E/E_v^e, i.e., the ratio of required and actual local training epochs. q_v^r and α_v^r denote the aggregation weight and the amplification factor, respectively.

Proof. The proof of Theorem V.1 is provided in Appendix A. □

Theorem V.1 can be considered as the main part and the error bound term. The main part is the first term in Eq. (27), which controls the convergence speed of AVFL. This part is depended on the learning rate η and not affected by α_n^r and β_n^r. The other terms can be considered as the error bound part affected by α_n^r and β_n^r. The detailed discussion is presented as follows.

For the main part, different learning rate strategies significantly affect AVFL convergence speed. With a dismishing learning rate strategy $\eta = \frac{1}{\sqrt{RE}}$, AVFL can achieve a sub-linear convergence speed, which is shown in following corollary.

Corollary V.2. *When Assumptions 1, 2, and 3 hold, with a diminishing learning rate* $\eta = \frac{1}{\sqrt{RE}}$, $b = 1$, *and* $c = E - 1$, *we have*

$$
\begin{aligned}
\frac{1}{R}\sum_{r=1}^{R}\|\nabla f(\mathbf{w}_r)\|^2 &\leq \frac{2\tau}{\sqrt{RE}}\mathbb{E}[f(\mathbf{w}_r) - f(\mathbf{w}_0)] \\
&+ \frac{2\tau\sigma^2}{RE}\left(10 + \frac{L}{2}\sqrt{\frac{E}{R}}\right)\sum_{t=1}^{R}\sum_{v=1}^{|\mathcal{V}|}\frac{(q_v^r)^2(\alpha_v^r)^2}{\beta_v^r} \\
&+ \frac{\tau d^2}{R}(L\sqrt{\frac{E}{R}} + 7)\sum_{r=1}^{R}\sum_{v=1}^{|\mathcal{V}|}\frac{q_v^r\alpha_v^r}{\beta_v^r} + \frac{2\tau d^2}{RE}\sum_{r=1}^{R}\sum_{v=1}^{|\mathcal{V}|}\beta_v^r(q_v^r)^2.
\end{aligned}
\tag{27}
$$

In addition, the error bound part is controlled by the amplification factor α_v^t and the ratio of the required and actual local training epochs β_v^t. To decrease the error bound of AVFL, the amplification factor and aggregation weight need to satisfy the following conditions.

Corollary V.3. *If* $\alpha_v^r < \beta_v^t$ *and* $(q_v^r)^2 \leq \frac{1}{\beta_v^r}$, *then Theorem V.1 guarantees that* $\frac{1}{R}\sum_{r=1}^{R}\|\nabla f(\mathbf{w}_r)\|^2 \to 0$ *as* $R \to \infty$.

Proof. The proof of Corollary V.3 is provided in Appendix B. □

3) Robustness Analysis of Amplification Factor: To evaluate the robustness of the amplification factor in heterogeneous vehicular environments, we analyze the stability of the approach under parameter variations and selection uncertainties. From Theorem V.1, convergence is guaranteed when $\alpha_v^r < \beta_v^r$ where $\beta_v^r = E/E_v^r$. The convergence bound shows that the error term scales linearly with $\sum_{r,v}\frac{q_v^r\alpha_v^r}{\beta_v^r}$. For any amplification factor perturbation $\alpha_v^r \to \alpha_v^r + \epsilon_v$ where $|\epsilon_v| \leq \epsilon_{\max}$, the additional error contribution is bounded by

$$
\Delta_\epsilon \leq \frac{\tau d^2}{RE}(\eta LE + 3 + \frac{4}{b})(1+c)\sum_{r,v}\frac{q_v^r|\epsilon_v|}{\beta_v^r} \leq C\cdot\epsilon_{\max}, \tag{28}
$$

where C is a problem-dependent constant. This linear relationship ensures graceful degradation under amplification errors.

C. Mobility-Aware Dual-Timescale Scheduling with Reinforcement Learning

We propose a mobility-aware scheduler to address unstable vehicle sets and competitive communications through a dual-timescale approach. We first analyze synchronized VFL to design a long-term macro scheduling strategy, then formulate the scheduling issue as a dynamic bin-packing problem solved via Actor-Critic reinforcement learning. The training efficiency balance involves a fundamental tradeoff - while more vehicles improve model performance, they can slow convergence speed if the slowest vehicle creates bottlenecks. Theorem V.4 quantifies this relationship.

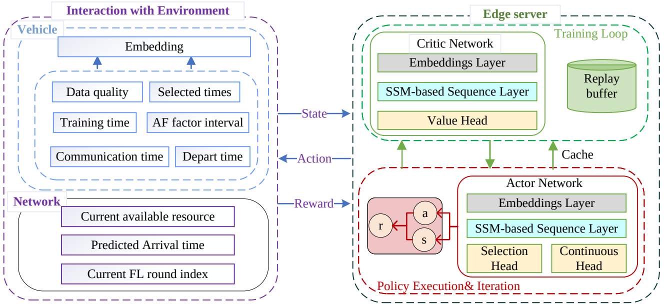


Fig. 4. The Actor-Critic-based scheduler design.

Theorem V.4. *Let* $|\mathcal{V}_r|$ *be the number of total available vehicles,* K_r *be the number of selected vehicles for FL, and* $\|\nabla F_v(\mathbf{w}_r)\|^2 \leq c^2$. *Define* $h_r(K_r) = 1 - \sum_{v=K_r+1}^{|\mathcal{V}_r|} q_v^r$ *as the proportion of selected vehicles' weights.*

Under Assumptions 1–3, the following bounds hold:

(a) Convergence upper bound is

$$
\begin{aligned}
&\frac{1}{R}\sum_{r=1}^{R}\|\nabla f(\mathbf{w}_r)\|^2 \\
&\quad\leq \frac{2(f(\mathbf{w}_1) - f(\mathbf{w}^*))}{\eta R} + \frac{d^2 + \eta(L-1)c^2}{R}\sum_{t=1}^{R}\frac{1}{h_r(K_r)}.
\end{aligned}
\tag{29}
$$

(b) Per-round loss reduction is

$$
f(\mathbf{w}_r) - f(\mathbf{w}_{r+1}) \geq \frac{\eta}{2}\|\nabla f(\mathbf{w}_r)\|^2 - \frac{\eta d^2 + \eta^2(L-1)c^2}{2h_r(K_r)}. \tag{30}
$$

Proof. The proof of Theorem V.4 is provided in Appendix C. □

Theorem V.4 shows that progressively increasing participant numbers can optimize both convergence speed and model performance. While larger $h_r(K_r)$ tightens the bound in Eq. (29), it also reduces the loss gap in Eq. (30), slowing convergence.

Let K_0 and $K_{\max}$ denote the initial and the maximum numbers of selected vehicles, respectively. The proposed macro scheduling policy implements this insight using an adaptive sigmoid function that is expressed as

$$
K_r = K_0 + \frac{K_{\max} - K_0}{1 + e^{-\kappa_1(r-\kappa_2)}}, \tag{31}
$$

where r is the index of round, κ_2 is a hyperparameter to control the sensitivity of α, and κ_1 is a constant. Eq. (31) balances rapid early convergence (fewer vehicles) with comprehensive final model performance (more vehicles).

1) DRL-based Problem Transformation: The problem formulated in Eq. (19) is similar to a 2D bin-pack problem but limited by unpredictable future vehicle traffic, making it difficult to solve with the existing linear programming techniques. To this end, we adopt DRL to solve Eq. (19) as a long-term optimization problem.

Based on the main idea of DRL, the above optimization problem can be transformed into an MDP, where the following components are included.

System State: From the beginning of the VFL communication round R_0 to the end of round R, the edge server in the area starts to observe the following state of networks $s^r(t)$. Consequently, the observation state $s^r(t)$ is characterized by

$$s^r(t) = \{\mathcal{V}_t^r, \mathcal{ST}_t^r, \mathbf{w}(r), B_t^r, \mathcal{T}_t^r, r\}, \tag{32}$$

where $\mathcal{V}_t^r$ denotes the current scheduled vehicle set, $\mathcal{ST}_t^r$ is the corresponding vehicle status of $\mathcal{V}_t^r$, $\mathbf{w}(r)$ is the FL workload, B_t^r is the available bandwidth, and $\mathcal{T}_t^r$ is the current remained time slot set. Specifically, the vehicle status $\mathcal{ST}_t^r$ includes the departure time τ_v^{dep}, the arrival time predicted by edge server τ_v^{arr}, the cumulative number of vehicle v selected to participate into training, the data quality $\mathcal{D}_v$, the times that v selected by edge server sl_v, the communication time cost $T_{v,pl}^{com}$ and training time cost T_v^{cmp}.

Action State: Based on the observed state, the edge server makes decision for the current arrival vehicle v, including the selection indicator x_v, the scheduled transmission time slot T_v^s, the utilized bandwidth b_v and the amplification factor α_v according to Eqs. (13) and (14). Thus, the action is expressed as

$$a^r(t) = \{x_v, T_v^s, b_v, \alpha_v\}, \tag{33}$$

Reward Function: To balance immediate model performance with long-term training efficiency and fairness, we design a multi-objective reward function. Let n_v^r denote the cumulative number of times vehicle v has been selected for participation up to round r. Based on the optimization objectives in Eq. (19), the reward function is defined as

$$r(s_t, a_t) = x_v \left(c_1 \frac{|\mathcal{V}_t^r|}{N^r}(T_{syn}^r - T_v^s) + c_2\theta_{acc,v} - c_3(e^{n_v^r} - 1) \right), \tag{34}$$

where c_1, c_2, and c_3 are positive weighting coefficients that balance the three objectives. The reward function comprises three components: (1) **Time Efficiency:** $c_1 \frac{|\mathcal{V}_t^r|}{N^r}(T_{syn}^r - T_v^s)$ rewards early task completion relative to the synchronization barrier, with $\frac{|\mathcal{V}_t^r|}{N^r}$ normalizing by the scheduling completion rate; (2) **Accuracy Improvement:** $c_2\theta_{acc,v}$ captures the immediate model performance gain from vehicle v's update; (3) **Fairness Penalty:** $c_3(e^{n_v^t} - 1)$ applies exponentially increasing penalties to frequently selected vehicles, ensuring equitable participation while deterring systematic favoritism.

Algorithm 2 The procedures of AC-based FL scheduler

```
1:  Input: Model performance threshold $\phi$, local epochs $E$,
    max rounds $R_{\max}$, max time $T_{syn}^R$
2:  Initialize actor-network $\Theta$ and critic-network $\Phi$
3:  for each episode do
4:      Initialize environment, set $t = 0$
5:      for round $r = 1 : R_{\max}$ do
6:          Initialize vehicle set $\mathcal{V}_a^r$ and use actor-network for
            initial scheduling
7:          while $t \leq T_{syn}^R$ do
8:              if new vehicle $v'$ arrives then
9:                  Add $v'$ to $\mathcal{V}_a^r$, $\mathcal{ST}_t^r$ and schedule using actor-
                    network
10:             end if
11:             Execute $a^r(t)$, obtain reward $r(t)$ and next state
                $s^r(t+1)$
12:             Update $\mathcal{V}_t^r$ and set $\mathcal{V}_a^r \leftarrow \mathcal{V}_a^r \setminus \{v'\}$
13:             Store experience tuple to replay-memory
14:             if $|\mathcal{V}_t^r| \geq N_r$ then
15:                 Store $T_{syn,a}^R$ and break
16:             end if
17:             $t \leftarrow t + 1$
18:         end while
19:         if Global model meets $\phi$ then
20:             Terminate FL task and break
21:         end if
22:     end for
23:     Update networks $\Theta$ and $\Phi$ using replay-memory
24: end for
```

2) AC-Based Solution for VFL: The AC-based deep reinforcement learning algorithm is one of the policy gradient methods that can address the MDP problems with continuous and high-dimension state and action spaces.

As shown in Fig. 4, AC uses an actor network to learn the policy that dynamically adjusts the action according to the system state, and uses a critic network to evaluate the performance of the learned policy. In addition, we use the replay memory to enhance the AC training stability and efficiency. Specifically, the actor and critic networks leverage a Mamba architecture to effectively model temporal dependency in the dynamic vehicular environment. [38] The Actor Network processes embedded vehicle and network state features through a Mamba sequence layer, outputting vehicle selection probabilities and continuous action parameters (amplification factor, transmission schedule, bandwidth). Concurrently, the Critic Network feeds the state and chosen action through its own Mamba sequence layer to produce a state-action value estimate ($V(s,a)$). Experiences (s_t, a_t, r_t, s_{t+1}) from the replay memory are used to train these Mamba-based networks, with the critic's evaluations guiding the actor's policy optimization. Specifically, to address the challenges of the high mobility of VENs and massive vehicles in VFL, we design a Mamba-based architecture to improve the decision efficiency.

As shown in Algorithm 2, lines 1–5 initialize the environment. For each FL round (lines 7–18), the server schedules vehicles from set $\mathcal{V}_a^r$, which denotes the available vehicles set in round r that can potentially be selected for participation. The scheduler activates when new vehicles enter, adding them to $\mathcal{V}_t^r$. It uses the actor network to generate action $a^r(t)$ based on state $s^r(t)$, transmits scheduling parameters, and stores decisions in replay memory. At line 14, when vehicle count reaches threshold N_r, remaining vehicles are deferred to the next round's set V_t^{r+1}. The process terminates (line 19) when the number of VFL round exceeds maximum rounds $R_{\max}$ or the global model meets performance requirement ϕ. Upon termination, both actor (Θ) and critic (Φ) networks are updated using replay memory.

Computational Complexity Analysis: For Algorithm 2, we analyze the complexity as follows. The complexity of state representation processing is $O(|\mathcal{V}_a^r| \cdot f)$, where f is the number of features per vehicle. The actor network forward pass for action selection requires $O(|\mathcal{V}_a^r|\cdot(d_s\cdot L_h+L_h^2\cdot L_{\text{actor}}))$ operations, where d_s is the state dimension, L_h is the hidden layer size, and L_{actor} is the number of actor layers. The complexity of critic network evaluation and experience storage is $O(|\mathcal{V}_a^r|\cdot((d_s+d_a)\cdot L_h+L_h^2\cdot L_{\text{critic}}))$, where d_a is the action dimension and L_{critic} is the number of critic layers. The complexity of network update operation is $O(B\cdot L_h^2\cdot(L_{\text{actor}}+L_{\text{critic}}))$, where B is the batch size from the replay memory.

Therefore, the overall per-round complexity is $O(|\mathcal{V}_a^r|\cdot L_h\cdot(d_s+d_a+L_h\cdot(L_{\text{actor}}+L_{\text{critic}}))+B\cdot L_h^2\cdot L_{\text{total}})$, where $L_{\text{total}}=L_{\text{actor}}+L_{\text{critic}}$ represents the total number of layers across both actor and critic networks.

In practical implementations, we use efficient mini-batch processing and GPU acceleration to handle large vehicle sets, as demonstrated in the experimental results.

D. Theoretical Guarantee

The vehicular scheduling problem is formulated as a Constrained Markov Decision Process (CMDP) with the primary objective in Eq. (11) subject to multiple constraints including fairness (Eq. (17)), resource allocation (Eq. (16)), and timing constraints. We now establish theoretical guarantees for our Mamba-based Actor-Critic scheduler in this constrained setting.

Theorem V.5 (Constrained Policy Convergence)**.** *Consider the VFL scheduling CMDP with bounded state space $|\mathcal{S}|$, action space $|\mathcal{A}|$, and constraint set $\mathcal{C}=\{c_1,...,c_m\}$ where c_i represents the constraints from Eqs.* (12)-(17)*. Let $\pi^\star$ denote the optimal constrained policy and π_θ denote the policy parameterized by the proposed Mamba-based actor network. Under standard regularity conditions (bounded rewards, Lipschitz continuity) and Slater's condition (feasible interior point exists), the primal-dual Actor-Critic algorithm converges such that*

$$\|V^{\pi_\theta}-V^{\pi^\star}\|_\infty\leq\varepsilon_{\textit{approx}}+\varepsilon_{\textit{stat}}+\varepsilon_{\textit{vehicular}},\tag{35}$$

where $\varepsilon_{\textit{approx}}$ represents the neural network approximation error, $\varepsilon_{\textit{stat}}$ represents the finite sample error, $\varepsilon_{\textit{vehicular}}=O(\lambda_{\max}/t_{\min})$ captures the impact of vehicle mobility.

Proof. The proof of Theorem V.5 is provided in Appendix D-A. □

Theorem V.6 (Constrained Optimality)**.** *For the vehicular scheduling CMDP with multiple constraints including the fairness constraint $\sum_{v\in\mathcal{V}_t}\exp(n_v^t)\leq\Gamma_t$ (where n_v^t denotes the cumulative selection count for vehicle v up to round t), the primal-dual Actor-Critic algorithm converges to a policy $\pi_{\theta^\star}$ that satisfies:*

Near-optimal performance: *The policy achieves near-optimal value with respect to the Lagrangian relaxation:*

$$\mathcal{L}(\pi,\boldsymbol{\lambda})=\mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right]-\sum_{i=1}^{m}\lambda_i\mathbb{E}_\pi[c_i(\pi)],\tag{36}$$

where $c_i(\pi)$ represents the expected violation of constraint i, including the fairness constraint $c_{\textit{fairness}}(\pi)=\mathbb{E}_\pi\left[\max\left(0,\sum_{v\in\mathcal{V}_t}\exp(n_v^t)-\Gamma_t\right)\right]$.

Constraint satisfaction guarantee: *After sufficient training iterations $T>T_0(\delta)$, the learned policy satisfies all the constraints with high probability for all the constraints in $\mathcal{C}$.*

Optimality gap: *The performance gap between the learned policy and the optimal constrained policy is bounded:*

$$V_c^{\pi^\star}(s)-V^{\pi_{\theta^\star}}(s)\leq\frac{1}{1-\gamma}\left(\varepsilon_{\textit{approx}}+\varepsilon_{\textit{dual}}+O(1/\sqrt{T})\right),\tag{37}$$

where $\varepsilon_{\textit{dual}}$ is the duality gap.

Proof. The proof of Theorem V.6 is provided in Appendix D-B. □

Theorem V.7 (Sample Efficiency)**.** *To achieve an ε-optimal constrained policy with the probability at least $1-\delta$ for the vehicular scheduling CMDP with m constraints, the required number of samples for the primal-dual Actor-Critic algorithm is*

$$N=\mathcal{O}\left(\frac{m^2}{(1-\gamma)^4\varepsilon^2}\cdot\log\frac{|\mathcal{S}||\mathcal{A}|}{\delta}\right),\tag{38}$$

where $|\mathcal{S}|$ and $|\mathcal{A}|$ represent the effective dimensions of the state and action spaces, and m is the number of constraints. For the vehicular federated learning scheduler with V vehicles, T time slots per round, B bandwidth levels, m constraints, and we have

$$N=\mathcal{O}\left(\frac{m^2}{(1-\gamma)^4\varepsilon^2}\cdot\log\frac{V^2T^2BH}{\delta}\right),\tag{39}$$

where H captures vehicle heterogeneity features.

Proof. The proof of Theorem V.7 is provided in Appendix D-C. □

Overall, the proposed MR-VFL maintains robustness in highly heterogeneous and uncertain VFL environments through: (1) a swift Mamba-based scheduler with theoretical performance guarantees (Theorems V.5–V.7), and (2) an adaptive macro-scheduling strategy based on Theorem V.4, which ensures graceful degradation with convergence rate $O(1/\sqrt{h_r(K_r)})$ as client availability $h_r(K_r)$ decreases, where $h_r(K_r)$ represents the proportion of the aggregation weights of the selected vehicles.

VI. Performance Evaluation

In this section, we perform extensive simulations to evaluate the performance of MR-VFL. We first select an area of Victoria, British Columbia, Canada, from OpenStreetMap [39], which has a radius of 2.5 km and can cover most of the Saanich core area in Victoria. Using SUMO [40], vehicles are simulated to enter this predetermined area with random initial speed, time, and position over 1000 seconds using a micro-movement model based on the Intelligent Driver Model (IDM) and the macro-movement model of Manhattan. The traffic flows are controlled by the traffic density factors, which we simulate in busy and sparse road conditions. In addition, to enhance the realism of the VFL simulations, the computational capacities of the vehicles are drawn from a Gaussian distribution whose mean is measured on a laptop

TABLE I
SIMULATION PARAMETERS

Parameter	Variable
Batch size	64
Devices per round	20
Learning rate	0.01
Data categories	10
Synchronous limit	1000s
Radius of selected area	2.5KM
Total road types	8
Traffic density	10 and 20
Bandwidth of vehicle i	sampled from $[10, 30]MHz$
Transmission power of vehicle i	sampled from $[23, 30]dBm$
Path loss constant	-40dB
Small-scale fading channel power gain	1
Reference distance	1m
Large-scale path loss factor	4
Noise power spectral density	-174dBm/Hz
Bandwidth of the server	50MHz
Transmission power of the server	30dBm

equipped with an Intel i7 CPU and a NVIDIA 4060 GPU, and the variance is set to 20.

We consider an FL scenario consisting of a server and $|\mathcal{V}|$ vehicles. The training tasks consist of the classification tasks on the FashionMNIST and EMNIST datasets, where a two-layered convolutional neural network (CNN) and LENET are adopted, respectively. In addition, we consider a non-i.i.d. data distribution according to a Dirichlet distribution, where the concentration parameter is set to 0.1. Therefore, the training data of each vehicle in the settings are different in both amounts and types. The detailed settings of the simulations are demonstrated in Table I.

A. Implementation Setup

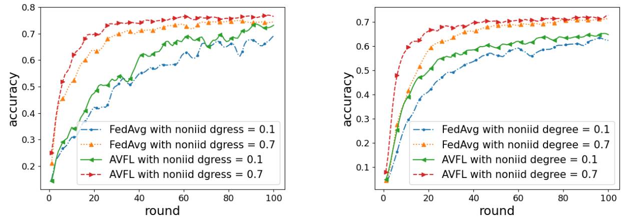
The proposed Mamba-based scheduler employs a 4-layer Mamba sequence model (256 units) for the actor network and a 3-layer Mamba model (128 units) for the critic network. We use learning rates of 3×10^{-4} (actor) and 1×10^{-3} (critic), with $\gamma = 0.92$. The model processes 32-dimensional embeddings of vehicle, resource, and temporal features. Computationally, the implementation requires 2.4 GFLOPS for training and 0.8 GFLOPS for inference, with a 12MB memory footprint. Benchmarking on an edge server (Intel Core i5, 16GB RAM) demonstrates 120 vehicles/second throughput with 8.2ms mean latency, sufficient for real-time VFL coordination in urban environments.

B. Benchmarks

To comprehensively explore the performance of MR-VFL, several state-of-the-art methods are selected as benchmarks.

FedAvg: A classical FL algorithm in which the server randomly selects a subset of vehicles to participate in the training process. Each selected vehicle trains its model locally and sends the updates to the server [41].

FDL-FL: It is a flexible FL participation scheme that considers the impacts of different vehicle participation patterns on FL convergence. It allows vehicles that cannot complete the training to update their results and scale them up according to the number of epoch rounds performed [32].



(a) Simulation results of CNN under FashionMNIST dataset

(b) Simulation results of LENET under EMNIST dataset

Fig. 5. Simulation results of different schemes under different non-i.i.d. degrees.

FEDDATE-CS: It presents a surplus vehicle scheduling policy considering the uncertainties of vehicles. The server in FEDDATE-CS can select a surplus vehicle set according to the observable information to ensure enough vehicles to complete the current FL training round [42].

FedBuff: It is a classic asynchronous scheme that can flexibly schedule vehicles into the training process. It can efficiently utilize communication resources and improve vehicle availability [43].

C. Impact of Non-i.i.d. Degree

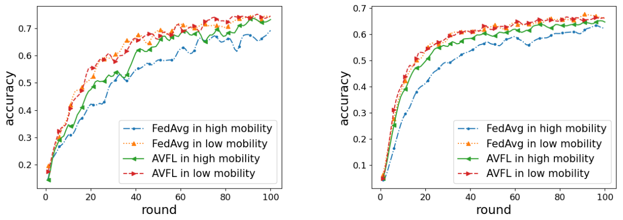
Fig. 5 shows the performance of different schemes with various degrees of non-i.i.d. data. As illustrated in Fig. 5, AVFL exhibits superior model accuracy compared to FedAvg, when the non-i.i.d. degree is 0.1 and 0.7. Notably, AVFL significantly outperforms FedAvg with a non-i.i.d. degree of 0.1. Under the EMNIST dataset and the LeNet model in Fig. 5 (b), AVFL's accuracy is approximately 9% higher than that of FedAvg over 20 rounds. This improvement can be attributed to AVFL allowing more vehicles to complete the training process, which increases the diversity of data and mitigates the impact of non-i.i.d. data, especially in scenarios with highly heterogeneous data.

D. Impact of Mobility Degree

Fig. 6 shows the performance of FedAvg and AVFL under different degrees of mobility, where high mobility means that vehicles have less sojourn time and experience higher probabilities of being unavailable than vehicles with low mobility. As shown in Fig. 6, AVFL can significantly mitigate the impact of high mobility in VFL. For example, AVFL with high and low mobility can achieve an accuracy similar to that of FedAvg with low mobility in Fig. 6 (a), which is 10% higher than the performance of FedAvg under high mobility. This observation indicates that AVFL improves the availability of vehicles by allowing them to train flexibly based on their sojourn time, thus mitigating the impact of high mobility.

E. Effectiveness of Amplification in AVFL

To further explore the impact of the amplification factor in AVFL, we introduce AVFL-A which denotes AVFL without



(a) Simulation results of CNN under FashionMNIST dataset

(b) Simulation results of LENET under EMNIST dataset

Fig. 6. Simulation results of different mobility degrees under different tasks.

the use of the amplification factor α to help eliminate the effects of inadequate training. The ablation experiments are conducted on FashionMNIST and EMNIST. As shown in Fig. 7, the proposed AVFL achieves the best accuracy in both tasks, while AVFL-A has similar performance compared to FedAvg and FDL-FL. In particular, in Fig. 7(b), AVFL achieves 60% accuracy in 32 rounds, which is 10% higher than that of other methods. As shown in Fig. 7(c), AVFL achieves 50% accuracy in 150 rounds, which is 10% higher than the other three methods. This observation indicates that the reinforcement process can effectively reduce the effects of insufficient training. Additionally, compared to FDL-FL with a constant amplification factor, AVFL adopts a dynamic amplification factor and outperforms FDL-FL.

F. Impact of AC-based Scheduler in MR-VFL

Fig. 9 illustrates the effectiveness of the proposed AC-based scheduler in MR-VFL. We use three metrics for performance evaluation: average training time of an FL round, longest scheduling time, and the number of vehicles successfully scheduled. The time limit of the VFL per round is set to 1000 seconds, and we test two traffic flow densities.

To comprehensively evaluate scheduling performance, we implement two distinct client selection approaches: constant selection number and greedy selection. The constant selection approach (Figs. 9(a), 9(b), and 9(c)) requires each algorithm to select exactly 20 vehicles per round, representing standard FL implementations that ensure consistent aggregation behavior. In contrast, the greedy selection approach in Figs. 9(d), 9(e), and 9(f) aims to maximize the number of participating vehicles in each round, potentially improving model quality while increasing resource utilization. In addition, we introduce a filter mechanism in AVFL as a new benchmark, which excludes vehicles without sufficient sojourn time for model training and transmission.

a) Constant selection number: As shown in Figs. 9(a) and 9(b), MR-VFL achieves the shortest average training time for each FL round, being 30% faster in high-density traffic flow and 3% faster in low-density traffic flow than FedAvg. Meanwhile, MR-VFL achieves a relatively fast scheduling speed compared to other SOTA methods under high and low traffic densities. In particular, in the low-density scenario, MR-VFL is nearly 20% faster than FEDDATE-CS and AVFL with a single filter. As illustrated in Fig. 9(c), MR-VFL is also effective in ensuring an adequate number of attendees. Although FEDDATE-CS outperforms other methods in the high-density scenario, it is not stable in the low-density scenario. In contrast, MR-VFL can reliably achieve the required number of participants in both scenarios.

b) Greedy selection: As shown in Figs. 9(d) and 9(e), the proposed scheduler achieves the shortest round time in both high and low density scenarios and is the fastest to complete the client selection process in each round. In the greedy manner, all benchmarks achieve the synchronous time limit of VFL, but the proposed scheduler can reduce nearly 10% time per round, leading to an improvement in training efficiency. Furthermore, in Fig. 9(f), the proposed scheduler can schedule the highest number of the selected vehicles.

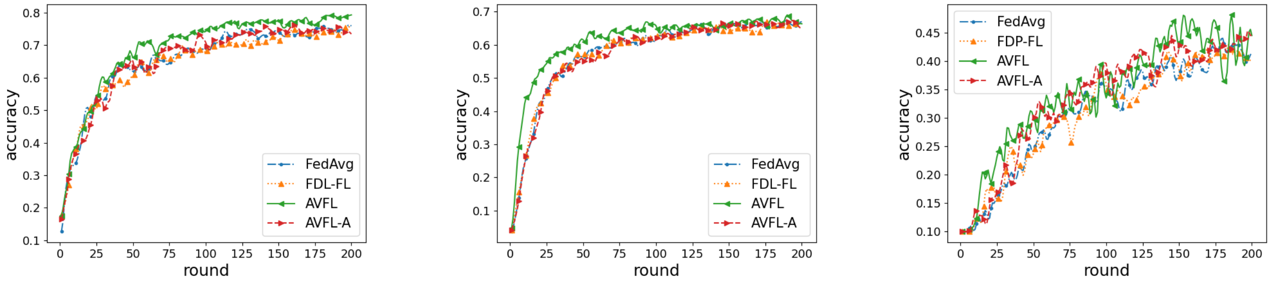
Fig. 10 highlights the AC-based scheduler's efficiency through experiments. Fig. 10 (a) and (b) evaluate the resilience using an adaptability score that weights the performance gains (20%/30%/50%) across escalating challenges—mobility, connectivity, and resource constraints. The metric normalizes growth to [0,1], rewards stability under stress, and defaults to 0.5 for incomplete phase data. Fig.10(c) demonstrates the efficiency of the proposed Mamba-based scheduler in large-scale vehicular scenarios compared to the alternative decision models. As the number of participating vehicles increases from 50 to 1000, the Mamba-based scheduler maintains the near-linear decision time scaling, significantly outperforming transformer-based and LSTM-based approaches that exhibit exponential growth in computational requirements.

The simulation results in Fig. 10 demonstrate the effectiveness of MR-VFL, which can be attributed to the ability of the AC-based scheduler to capture the dynamics of VENs and schedule the vehicles quickly.

G. Impact of Macro Scheduling Strategy in MR-VFL

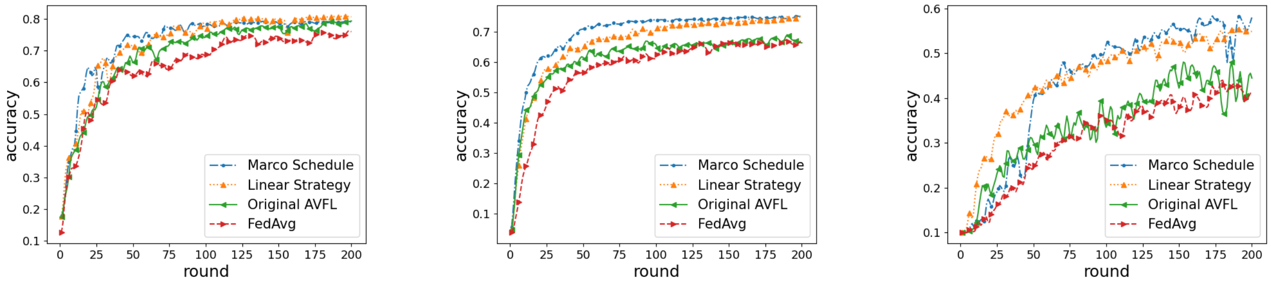
Fig. 11 illustrates the effect of macro scheduling on AVFL convergence, demonstrating significant performance improvements across all datasets, with Fig. 11 (b) showing approximately 10% accuracy gain while accelerating convergence.

To evaluate the macro scheduling strategy in Eq. (31), we compare our macro scheduling strategy against the linear-increasing selection strategy, constant selection, and standard FedAvg across three datasets (Fig. 8). Results show that the method adapting the macro scheduling strategy achieves a superior balance between early-stage convergence and final model quality. For example, as shown in Fig. 8 (b), the proposed macro scheduling strategy demonstrates 10% higher accuracy within 20 rounds on EMNIST compared to the linear increasing strategy. By dynamically adjusting participant numbers, the proposed macro scheduling strategy selects fewer vehicles initially for faster convergence, progressively increasing them for better final performance. These findings align with Theorem V.4, which establishes that while more vehicles improve model performance (tightening the bound in Eq. (29)), they can slow convergence by reducing the loss gap in Eq. (30). The adaptive strategy effectively balances this trade-off throughout training.



(a) Simulation results of CNN under FashionMNIST dataset (b) Simulation results of LENET under EMNIST dataset (c) Simulation results of ResNet18 under CIFAR10 dataset

Fig. 7. Simulation results of different methods under different tasks.



(a) Simulation results of CNN under FashionMNIST dataset (b) Simulation results of LENET under EMNIST dataset (c) Simulation results of ResNet18 under CIFAR10 dataset

Fig. 8. Simulation results of different Marco strategies under different tasks.

H. Performance Comparison

To further explore the performance of MR-VFL, we compare it with different SOTA schemes, as shown in Fig. 12. In addition to the three synchronous SOTA schemes, we use FedBuff [43], Fadas [44], Fedasync [45] and port [46] as the benchmarks because the asynchronous scheme can select all available vehicles during the training process, significantly alleviating the impact of mobility in synchronous VFL.

As shown in Fig. 12, MR-VFL significantly outperforms all state-of-the-art methods, including asynchronous approaches such as FedBuff. Specifically, MR-VFL achieves approximately 5% and 10% higher accuracy than the compared methods on EMNIST and CIFAR-10 datasets, respectively, demonstrating substantial improvements across varying task complexities.

To verify the theoretical convergence guarantees from Theorem V.1, Fig. 13 presents the training loss evolution. MR-VFL demonstrates the fastest convergence on the EMNIST dataset while maintaining competitive convergence speed on CIFAR-10 compared to asynchronous methods, confirming the theoretical predictions.

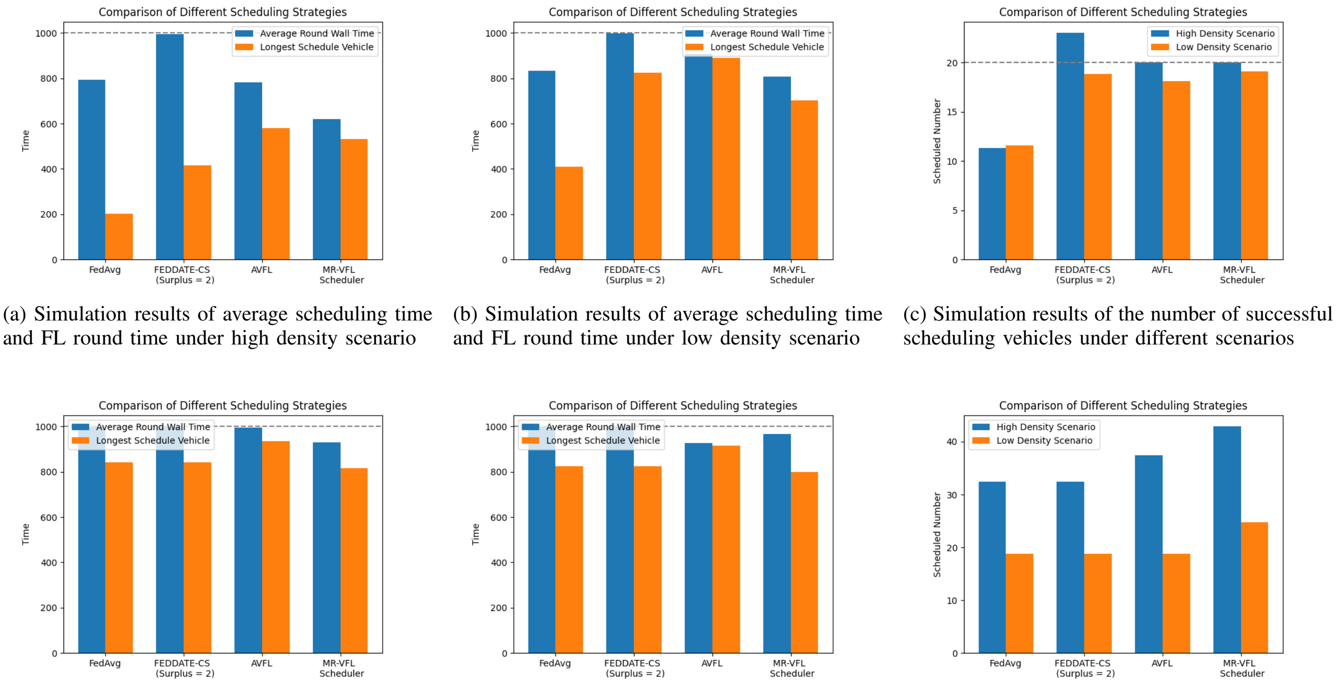
The combined results from Figs. 12 and 13 demonstrate that MR-VFL achieves both superior final accuracy and efficient convergence compared to existing benchmarks. This performance advantage is particularly pronounced on complex tasks like CIFAR-10, where vehicle unavailability due to high mobility has greater impact on model quality. MR-VFL effectively mitigates these mobility-induced challenges through adaptive training and intelligent scheduling, leading to consistent performance improvements.

These empirical observations align closely with the theoretical bounds in Theorem V.1, confirming that the approach successfully minimizes convergence degradation despite the inherent challenges of vehicle mobility and computational heterogeneity in dynamic vehicular environments.

VII. Conclusion

In this paper, we have proposed MR-VFL, a novel VFL scheme combining the AVFL training scheme with a dual-timescale scheduling mechanism. AVFL dynamically adjusts local training epochs based on heterogeneous vehicle capabilities while using amplification factors to mitigate insufficient local training effects. The scheduling mechanism that optimizes both long-term model performance and immediate training efficiency by intelligently anticipating vehicle mobility patterns, deriving optimal strategies across macro and micro timescales. Extensive simulations demonstrate that MR-VFL effectively alleviates performance degradation in dynamic vehicular environments while significantly improving training efficiency.

Despite these advances, MR-VFL faces limitations in sparse vehicle environments where statistical heterogeneity increases, which affects scheduling reliability. Additionally, the computational requirements of the Mamba-based scheduler may be challenged for the current RSU and vehicles, which can be further optimized.



(a) Simulation results of average scheduling time and FL round time under high density scenario

(b) Simulation results of average scheduling time and FL round time under low density scenario

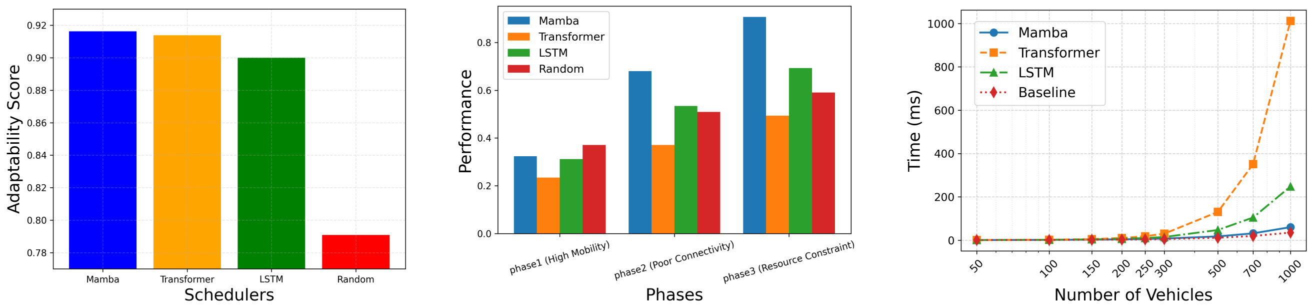
(c) Simulation results of the number of successful scheduling vehicles under different scenarios

(d) Simulation results of average scheduling time and FL round time under high density scenario in greedy manner

(e) Simulation results of average scheduling time and FL round time under low density scenario in greedy manner

(f) Simulation results of the number of successful scheduling vehicles in greedy manner

Fig. 9. Simulation results of different scheduling strategies under different scenarios.

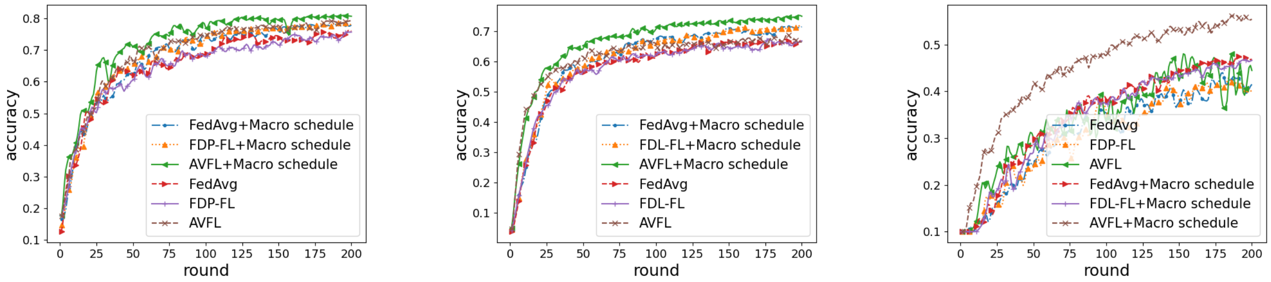


(a) Simulation results of average scheduling adaptability with different scheduler structure

(b) Simulation results of average performance with different scheduler structure under different phases

(c) Simulation results of the decision time scalability with increasing vehicles

Fig. 10. Simulation results of performance of AC-based scheduler with different decision model structures under different scenarios.

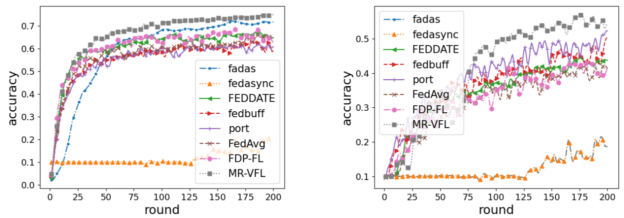


(a) Simulation results of LENET on FashionMNIST dataset

(b) Simulation results of LENET on EMNIST dataset

(c) Simulation results of ResNet18 on CIFAR10 dataset

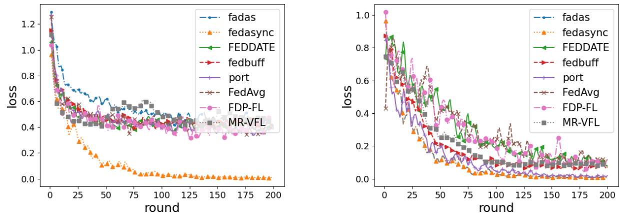
Fig. 11. Simulation results of the effect of the macro scheduling policy under different tasks.



(a) Simulation results of LENET under EMNIST dataset

(b) Simulation results of ResNet18 under CIFAR10 dataset

Fig. 12. Simulation results comparisons of model performance with other benchmarks under different models and datasets.



(a) Simulation results of LENET under EMNIST dataset

(b) Simulation results of ResNet18 under CIFAR10 dataset

Fig. 13. Simulation results comparisons of training loss with other benchmarks under different models and datasets.

Future research should explore several promising directions: (1) multi-modal learning integration to leverage diverse vehicular sensor data through modality-specific techniques; (2) hybrid approaches for sparse environments and predictive channel modeling for proactive scheduling; and (3) cross-RSU coordination with 6G network slicing for seamless coverage and optimized resource allocation across heterogeneous vehicles.

References

[1] N. or Next Move Strategy Consulting. (2023) Autonomous vehicle market by type (semi-autonomous and fully-autonomous(level 4/5), by propulsion type (internal combustion engine (ice), batteryelectric vehicles (bev), hybrid electric vehicles (hev), plug-in hybrid electricvehicle (phev), and fuel cell electric vehicle (fcev)). [Online]. Available: https://www.nextmsc.com/report/autonomous-vehicle-market

[2] D. Katare, D. Perino, J. Nurmi, M. Warnier, M. Janssen, and A. Y. Ding, "A Survey on Approximate Edge AI for Energy Efficient Autonomous Driving Services," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2714–2754, Aug. 2023.

[3] S. V. Balkus, H. Wang, B. D. Cornet, C. Mahabal, H. Ngo, and H. Fang, "A Survey of Collaborative Machine Learning Using 5G Vehicular Communications," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 2, pp. 1280–1303, 2022.

[4] M. P. Aparna, R. Gandhiraj, and M. Panda, "Steering Angle Prediction for Autonomous Driving Using Federated Learning: The Impact of Vehicle-to-Everything Communication," in *Proc. International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, 2021, pp. 1–7.

[5] Intel Corporation, "Data is the new oil in the future of automated driving," Intel Corporation, Tech. Rep., 2016, white Paper. [Online]. Available: https://www.intel.com/content/www/us/en/automotive/automotive-autonomous-driving-data-paper.html

[6] IHS Markit, "The connected car: Global forecast report," IHS Markit, Tech. Rep., 2022, market Research Report.

[7] D. Sirohi, N. Kumar, P. S. Rana, S. Tanwar, R. Iqbal, and M. Hijjii, "Federated Learning for 6G-enabled Secure Communication Systems: A Comprehensive Survey," *Artificial Intelligence Review*, vol. 56, no. 10, pp. 11 297–11 389, 2023.

[8] V. P. Chellapandi, L. Yuan, C. G. Brinton, S. H. .Zak, and Z. Wang, "Federated Learning for Connected and Automated Vehicles: A Survey of Existing Approaches and Challenges," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 119–137, Jan. 2024.

[9] B. Yang, X. Cao, K. Xiong, C. Yuen, Y. L. Guan, S. Leng, L. Qian, and Z. Han, "Edge Intelligence for Autonomous Driving in 6G Wireless System: Design Challenges and Solutions," *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 40–47, Apr. 2021.

[10] D. M. Manias and A. Shami, "Making a Case for Federated Learning in the Internet of Vehicles and Intelligent Transportation Systems," *IEEE Network*, vol. 35, no. 3, pp. 88–94, Jun. 2021.

[11] J. Thunberg, N. Lyamin, K. Sj"oberg, and A. Vinel, "Vehicle-to-Vehicle Communications for Platooning: Safety Analysis," *IEEE Netw. Lett.*, vol. 1, no. 4, pp. 168–172, Dec. 2019.

[12] C. He, T. H. Luan, N. Cheng, G. Wei, Z. Su, and Y. Liu, "Federated Learning Based Vehicular Threat Sharing: A Multi-Dimensional Contract Incentive Approach," in *Proc. IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, Hong Kong, China, 2023, pp. 1–5.

[13] J. Du, T. Lin, C. Jiang, Q. Yang, C. F. Bader, and Z. Han, "Distributed foundation models for multi-modal learning in 6g wireless networks," *IEEE Wireless Communications*, vol. 31, no. 3, pp. 20–30, 2024.

[14] M. H. C. Garcia, A. Molina-Galan, M. Boban, J. Gozalvez, B. Coll-Perales, T. Şahin, and A. Kousaridas, "A Tutorial on 5G NR V2X Communications," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1972–2026, Feb. 2021.

[15] X. Song, Y. Hua, Y. Yang, G. Xing, F. Liu, L. Xu, and T. Song, "Distributed Resource Allocation With Federated Learning for Delay-Sensitive IoV Services," *IEEE Trans. Veh. Technol.*, vol. 73, no. 3, pp. 4326–4336, Oct. 2024.

[16] H. Zhang, X. Zhang, and D. K. Sung, "An Efficient Cooperative Transmission Based Opportunistic Broadcast Scheme in VANETs," *IEEE Trans. Mobile Comput.*, vol. 22, no. 3, pp. 1327–1342, Aug. 2021.

[17] M. F. Pervej, R. Jin, and H. Dai, "Resource Constrained Vehicular Edge Federated Learning With Highly Mobile Connected Vehicles," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 6, pp. 1825–1844, May 2023.

[18] G. Wang *et al.*, "Efficient and Secure Pedestrian Detection in Intelligent Vehicles Based on Federated Learning," in *Proc. IEEE 95th Vehicular Technology Conference (VTC2022-Spring)*, Helsinki, Finland, 2022, pp. 1–5.

[19] J. Chi *et al.*, "Federated Learning Empowered Edge Collaborative Content Caching Mechanism for Internet of Vehicles," in *Proc. IEEE/IFIP Network Operations and Management Symposium*, Budapest, Hungary, 2022, pp. 1–5.

[20] T. Zeng *et al.*, "Federated Learning on the Road Autonomous Controller Design for Connected and Autonomous Vehicles," *IEEE Trans. Wireless Commun.*, vol. 21, no. 12, pp. 10 407–10 423, Dec. 2022.

[21] X. Zhou *et al.*, "STFL: Spatio-Temporal Federated Learning for Vehicle Trajectory Prediction," in *Proc. IEEE 2nd International Conference on Digital Twins and Parallel Intelligence (DTPI)*, Boston, MA, USA, 2022, pp. 1–6.

[22] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, L.-N. Tran, S. Gong, and E. Dutkiewicz, "Dynamic Federated Learning-Based Economic Framework for Internet-of-Vehicles," *IEEE Trans. Mobile Comput.*, vol. 22, no. 4, pp. 2100–2115, Oct. 2021.

[23] S. Samarakoon *et al.*, "Federated Learning for Ultra-Reliable Low-Latency V2V Communications," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, UAE, 2018, pp. 1–7.

[24] H. Xiao, X. Hu, H. Zhuang, and H. Sun, "Vehicle Selection and Resource Optimization for Federated Learning in Vehicular Edge Computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11 073–11 087, Aug. 2021.

[25] S. B. Prathiba, J. Ying, K. Liang, H. Wang, and Z. Yan, "Federated Learning Empowered Computation Offloading and Resource Management in 6G-V2X," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3234–3243, Oct. 2022.

[26] M. Hu, Y. Han, H. Wang, Q. Fan, J. Xiong, L. Yang, and Y. Zhao, "AutoFL: A Bayesian Game Approach for Autonomous Client Participation in Federated Edge Learning," *IEEE Trans. Mobile Comput.*, vol. 23, no. 1, pp. 194–208, Jan. 2024.

[27] M. Zhang, G. Zhu, S. Wang, J. Jiang, Q. Liao, C. Zhong, and S. Cui, "Communication-Efficient Federated Edge Learning via Optimal Probabilistic Device Scheduling," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 10, pp. 8536–8551, Apr. 2022.

[28] T. Yin, L. Li, W. Lin, T. Ni, Y. Liu, H. Xu, and Z. Han, "Joint Client Scheduling and Wireless Resource Allocation for Heterogeneous Federated Edge Learning With Non-IID Data," *IEEE Trans. Veh. Technol.*, vol. 73, no. 4, pp. 5742–5754, Nov. 2024.

[29] C.-H. Hu, Z. Chen, and E. G. Larsson, "Scheduling and Aggregation Design for Asynchronous Federated Learning over Wireless Networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 874–886, Feb. 2023.

[30] F. Zhang, J. Chen, K. Wang, and W. Chen, "Device Scheduling for Relay-Assisted Over-the-Air Aggregation in Federated Learning," *IEEE Trans. Veh. Technol.*, vol. 73, no. 5, pp. 7412–7417, Dec. 2024.

[31] S. Wang and M. Ji, "A Unified Analysis of Federated Learning with Arbitrary Client Participation," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, vol. 35, New Orleans, LA, USA, 2022, pp. 19 124–19 137.

[32] Y. Ruan, X. Zhang, S.-C. Liang, and C. Joe-Wong, "Towards Flexible Device Participation in Federated Learning," in *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, Virtual Event, 2021, pp. 3403–3411.

[33] S. Liu *et al.*, "FedCPF: An Efficient-Communication Federated Learning Approach for Vehicular Edge Computing in 6G Communication Networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1616–1629, Feb. 2022.

[34] Y. J. Cho, P. Sharma, G. Joshi, Z. Xu, S. Kale, and T. Zhang, "On the Convergence of Federated Averaging With Cyclic Client Participation," in *Proc. International Conference on Machine Learning*, Honolulu, HI, USA, 2023, pp. 5677–5721.

[35] A. Abdi, C. Amrit, R. Darvizeh, and P.-O. Siebers, "A review of travel and arrival-time prediction methods on road networks: Classification, challenges and opportunities," *PeerJ Computer Science*, vol. 7, p. e689, Oct. 2021.

[36] Z. Wang, K. Fu, and J. Ye, "Learning to estimate the travel time," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, London, U.K., Aug. 2018, pp. 858–866.

[37] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Science Business Media, 2013, vol. 87.

[38] T. Dao and A. Gu, "Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality," in *Proc. International Conference on Machine Learning (ICML)*, vol. 235, 21–27 Jul 2024, pp. 10 041–10 071.

[39] (2023) OpenStreetMap. [Online]. Available: https://www.openstreetmap.org

[40] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic Traffic Simulation Using SUMO," in *Proc. 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, USA, 2018, pp. 2575–2582.

[41] 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*, 2017, pp. 1273–1282.

[42] Y. Li, F. Li, L. Chen, L. Zhu, P. Zhou, and Y. Wang, "Power of Redundancy: Surplus Client Scheduling for Federated Learning Against User Uncertainties," *IEEE Trans. Mobile Comput.*, vol. 22, no. 9, pp. 5449–5462, May 2022.

[43] J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, M. Rabbat, M. Malek, and D. Huba, "Federated Learning With Buffered Asynchronous Aggregation," in *Proc. International Conference on Artificial Intelligence and Statistics*, 2022, pp. 3581–3607.

[44] Y. Wang, S. Wang, S. Lu, and J. Chen, "Fadas: Towards federated adaptive asynchronous optimization," *arXiv preprint arXiv:2407.18365*, 2024.

[45] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous federated optimization," *arXiv preprint arXiv:1903.03934*, 2019.

[46] N. Su and B. Li, "How asynchronous can federated learning be?" in *Proc. International Symposium on Quality of Service (IWQoS)*. IEEE, 2022, pp. 1–11.

[47] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, 2nd ed., ser. Applications of Mathematics. Springer Science & Business Media, 2003, vol. 35.

[48] Jalaj Bhandari and D. Russo, "Global optimality guarantees for policy gradient methods," 2022. [Online]. Available: https://arxiv.org/abs/1906.01786

[49] S. Kakade and J. Langford, "Approximately optimal approximate reinforcement learning," in *Proc. International Conference on Machine Learning (ICML)*, 2002, pp. 267–274.

[50] A. Agarwal, N. Jiang, S. M. Kakade, and W. Sun, *Reinforcement Learning: Theory and Algorithms*. CSRL Preprint, 2020.

Tianao Xiang received the B.E. degree in software engineering from Northeastern University, Shenyang, China, in 2018. He is currently pursuing the Ph.D. degree in computer science at Northeastern University, Shenyang, China. His research interests include mobile edge computing, Internet of vehicles and federated learning, etc.

Yuanguo Bi (M'11) received the Ph.D. degree in computer science from Northeastern University, Shenyang, China, in 2010. He was a Visiting Ph.D. Student with the BroadBand Communications Research (BBCR) lab, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada from 2007 to 2009. He is currently a Professor with the School of Computer Science and Engineering, Northeastern University. He has authored/coauthored more than 50 journal/conference papers, including high quality journal papers, such as IEEE JSAC, IEEE TWC, IEEE TITS, IEEE TVT, IEEE IoT Journal, IEEE Communications Magazine, IEEE Wireless Communications, IEEE Network, and mainstream conferences, such as IEEE Global Communications Conference, IEEE International Conference on Communications. His research interests include medium access control, QoS routing, multihop broadcast, and mobility management in vehicular networks, software-defined networking, and mobile edge computing. Dr. Bi has served as an Editor/Guest Editor for IEEE Communications Magazine, IEEE Wireless Communications, IEEE ACCESS. He has also served as the Technical Program Committee member for many IEEE conferences.

Lin Cai (S'00-M'06-SM'10-F'20) has been with the Department of Electrical & Computer Engineering at the University of Victoria since 2005 and is currently a Professor. She is an NSERC E.W.R. Steacie Memorial Fellow, a Canadian Academy of Engineering (CAE) Fellow, an Engineering Institute of Canada (EIC) Fellow, and an IEEE Fellow. In 2020, she was elected as a Member of the Royal Society of Canada's College of New Scholars, Artists and Scientists, and a 2020 "Star in Computer Networking and Communications" by N2Women. Her research interests span several areas in communications and networking, with a focus on network protocol and architecture design supporting ubiquitous intelligence. She received the NSERC Discovery Accelerator Supplement (DAS) Grants in 2010 and 2015, respectively. She co-founded and chaired the IEEE Victoria Section Vehicular Technology and Communications Joint Societies Chapter. She has been elected to serve the IEEE Vehicular Technology Society (VTS) Board of Governors, 2019 - 2024, and served as its VP Mobile Radio from 2023 to 2024. She served as a Board Member of IEEE Women in Engineering from 2022 to 2024, and a Board Member of IEEE Communications Society (ComSoc) from 2024 - 2026.

Mingjian Zhi received the B.S. degree in network engineering from the Hebei University of Technology, Tianjin, China, in 2018, and the M.S. degree in computer application technology in 2021 from Northeastern University, Shenyang, China, where she is currently working toward the Ph.D. degree in computer science and technology with the School of Computer Science and Engineering. Her current research interests include personalized federated learning and applications in mobile edge computing.