Quantum-enhanced Federated Learning for Metaverse-empowered Vehicular Networks

Bishmita Hazarika, *Member, IEEE*, Keshav Singh, *Member, IEEE*,
Octavia A. Dobre, *Fellow, IEEE*, Chih-Peng Li, *Fellow, IEEE*, and Trung Q. Duong, *Fellow, IEEE*

***Abstract*—In the rapidly evolving domain of vehicular metaverse, this study introduces a cutting-edge quantum-based decentralized and heterogeneity-aware federated learning framework for vehicular metaverse named QV-FEDCOM, which stands as a testament to the innovative fusion of quantum computing principles with federated learning (FL). This framework is ingeniously tailored to address the challenges in a vehicular metaverse, offering a cost-efficient and adaptive solution for the dynamic vehicular landscape. QV-FEDCOM is strengthened by key components like quantum sequential-training-program, with reinforcement learning-based dynamic mode switching to reduce communication costs and manage vehicle states adaptively, and the quantum vehicle-context-grouping utilizing hierarchical clustering and simulated annealing for effective vehicle grouping based on contextual data similarity, addressing the complexities of data heterogeneity. Additionally, the integration of quantum-inspired principal component analysis (Q-PCA) enhances memory efficiency, further optimizing the framework. These elements converge in the QV-FEDCOM algorithm, establishing a decentralized, efficient, and context-aware quantum federated learning (QFL) process that redefines learning dynamics in the vehicular metaverse. Our study also introduces an innovative quantum trajectory loss (QTL) function, specifically designed for trajectory prediction tasks, which combines the Huber loss with an angular deviation penalty to robustly handle errors and penalize large deviations in the predicted trajectory angle. The effectiveness of the QV-FEDCOM framework is rigorously validated through comprehensive simulations, with its performance meticulously compared against various adaptations, showcasing its transformative capabilities within the vehicular metaverse ecosystem.**

***Index Terms*—Quantum Federated Learning (QFL), Quantum Neural Network (QNN), Vehicular Metaverse.**

I. Introduction

As the Internet-of-Vehicles (IoV) and automated driving technologies rapidly advance, we are moving towards a future where vehicles and their networks are fully autonomous. This evolution will bring about a significant increase in data processing demands, with vehicular networks potentially handling terabytes of data per minute. Such a high volume and speed of data processing, essential in dynamic vehicular networks, pose substantial challenges [1]. In this context, federated learning (FL) emerges as a significant advancement that eliminates the traditional requirement of gathering data centrally on a singular device, instead distributing the machine learning process across various clients [2]–[4]. FL enhances processing efficiency and personalizes learning by utilizing diverse data from each client, while also facilitating the secure sharing of insights across a broad network, thereby maintaining user data confidentiality [4]. Research in this domain has extensively explored various applications of FL, encompassing horizontal [5], vertical [6], and transfer learning methodologies [7]. These algorithms demonstrate the effectiveness of FL in improving collaborative intelligence and protecting user data privacy [8], [9], and for our discussion, we will refer to these traditional methods collectively as classical federated learning (CFL).

Despite its advantages in privacy and collaborative training, CFL faces limitations in addressing the challenges posed by dynamic vehicular networks [10]. These limitations include data inconsistency, model obsolescence, scalability issues, high communication costs, difficulty handling heterogeneous and non-IID data, and resource constraints at edge devices. Consequently, classical computation methods are unlikely to meet the processing and quality-of-service demands required in such networks [11]. Quantum processing emerges as a promising solution to these challenges [12], [13], addressing them in the following ways:

- **Quantum Speedup for Efficient Model Convergence:** Quantum parallelism enables simultaneous exploration of multiple states, greatly accelerating model training and convergence. This is crucial for real-time decision-making in vehicular networks, where traditional CFL struggles to process large, dynamic datasets within strict time constraints.

The work of K. Singh and C.-P. Li was supported in part by the National Science and Technology Council of Taiwan under Grant NSTC 113-2218-E-110 -008, Grant NSTC 112-2221-E-110-029-MY3, and in part by the Sixth Generation Communication and Sensing Research Center funded by the Higher Education SPROUT Project, the Ministry of Education of Taiwan. The work of O. A. Dobre was supported in part by the Canada Research Chairs Program-Canada Research Chair CRC-2022-00187, and the work of T. Q. Duong was supported by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109. (*Corresponding author: Keshav Singh*).

Bishmita Hazarika was with the Institute of Communications Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan. She is now with the Faculty of Engineering and Applied Science, Memorial University, St. John's, Canada. (email: bhazarika@mun.ca).

Keshav Singh, and Chih-Peng Li are with the Institute of Communications Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan. (e-mail: {keshav.singh, cpli}@mail.nsysu.edu.tw).

O. A. Dobre is with the Faculty of Engineering and Applied Science, Memorial University, St. John's, Canada. (email: odobre@mun.ca).

Trung Q. Duong is with the Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's, NL A1C 5S7, Canada, and also with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, BT7 1NN Belfast, U.K. (e-mail: tduong@mun.ca).

This paper has been presented in part at the IEEE International Conference on Communication (ICC), 9–13 June 2024, Denver, CO, USA.

- **Reduced Communication Overhead:** QFL leverages quantum communication techniques such as entanglement and teleportation, which allow faster, more efficient synchronization of model updates across nodes. This significantly reduces the communication burden compared to CFL, particularly in large-scale, decentralized networks.
- **Handling Data Heterogeneity:** QFL utilizes quantum-inspired optimization methods like Q-VCG (Quantum Vehicular-Context Grouping) to dynamically cluster vehicles based on contextual data. This helps to address the challenges posed by heterogeneous and non-IID data, where CFL often struggles to generalize effectively across different data distributions.
- **Memory and Resource Efficiency:** Techniques like Q-PCA (Quantum-Inspired Principal Component Analysis) compress quantum data, reducing the quantum memory footprint while preserving essential information. This optimizes resource utilization in edge environments where computational power and memory are limited.
- **Enhanced Privacy and Security:** QFL enhances privacy by employing Quantum Secure Multi-party Computation (SMC), enabling secure global model updates without exposing individual vehicle data. Unlike CFL, which relies on traditional privacy-preserving techniques, QFL offers a more efficient and secure approach with lower overhead.

However, equipping every vehicle and infrastructure component within the IoV with quantum processors is impractical. Instead, the concept of the vehicular metaverse offers a strategic alternative, where quantum computations are managed at the edge or within virtual environments using quantum servers [14], [15]. This approach merges quantum computing advancements with practical implementation strategies, potentially revolutionizing the operation of vehicular networks. Thus, integrating quantum computing with FL is necessary to evolve and meet the challenges of future autonomous vehicular networks. Classical FL, while effective today, is not well-suited to handle the communication, scalability, and heterogeneity issues that will arise in next-generation vehicular systems. Quantum-based FL, with its advanced processing, communication, and privacy-preserving capabilities, is, therefore, the logical next step in this evolution.

Building on this foundation, quantum federated learning (QFL) presents itself as an innovative solution to overcome the inherent limitations of CFL [16], [17]. Utilizing quantum neural networks (QNN) [18], QFL offers an advanced data processing capability, enhanced model precision, and improved security, aligning well with the dynamic requirements of the vehicular metaverse. The transition to QFL is not just an incremental step but a necessity for keeping pace with rapid technological advancements and avoiding obsolescence. Nevertheless, the practical application of QFL in vehicular networks encounters its own set of challenges [19], including the integration of diverse data types, the need for heightened computational power, and addressing quantum-specific complexities. Several recent articles have addressed various challenges in quantum communications. For instance, [20] proposed a novel quantum communication scheme that improves quantum bit error ratio, yield, and goodput, circumventing delays associated with conventional approaches by utilizing realistic noisy pre-shared entanglement. In [21], the authors study a quantum secure direct communication protocol to address challenges including reliance on quantum memory, susceptibility to eavesdropping, and low transmission reliability, significantly improving robustness and secure information rate compared to existing protocols. The authors in [22] explored using the quantum approximation optimization algorithm to solve the maximum likelihood detection problem.

In recent literature, the exploration of QFL within classical network contexts has gained significant attention, as evidenced by studies such as those cited in [19], [23]–[27]. These studies explored a variety of aspects related to QFL, employing both quantum-based and classical data models to enhance our understanding of its practical applications. The study in [24] specifically examines the influence of data non-uniformity on the performance of QFL, shedding light on the challenges posed by varied data sets. Meanwhile, [25] takes a closer look at the difficulties arising from the data that does not exhibit independent and identical distribution (non-iid), proposing a novel framework that decomposes a global quantum channel into localized channels. This approach significantly improves QFL performance in non-iid settings, marking a substantial advancement in the field. Further, [26] introduces an innovative federated QNN that incorporates quantum teleportation. This network particularly focuses on optimizing resource allocation in wireless communications and demonstrates enhanced efficiency and comparable performance in power allocation for non-orthogonal multiple access (NOMA)-based systems. However, the practical application of QFL is not without challenges. As highlighted in [19], issues such as the integration of heterogeneous data, increased computational demands, and quantum-specific complexities remain significant hurdles. In a similar vein, [28] investigates a QFL-based approach that prioritizes the safe and effective combination of local model parameters using quantum bits, applicable across various model types in both centralized and decentralized settings. Notably, recent research efforts, particularly those documented in [29] and [30], have contributed to the development of efficient FL systems and decentralized, secure QFL frameworks. Yet, these studies tend to concentrate on individual components of QFL and lack a comprehensive analysis of its broader applications, particularly over classical networks. This observation underscores the need for more extensive research into QFL, with a specific focus on its potential within the context of quantum communication networks (QCNs) [31].

Our study aims to bridge this gap by proposing a novel framework: a decentralized, context-aware, QFL-based model, specifically engineered for scalability and apt for large-scale implementation. We evaluate the effectiveness of our framework through a simulated vehicle trajectory

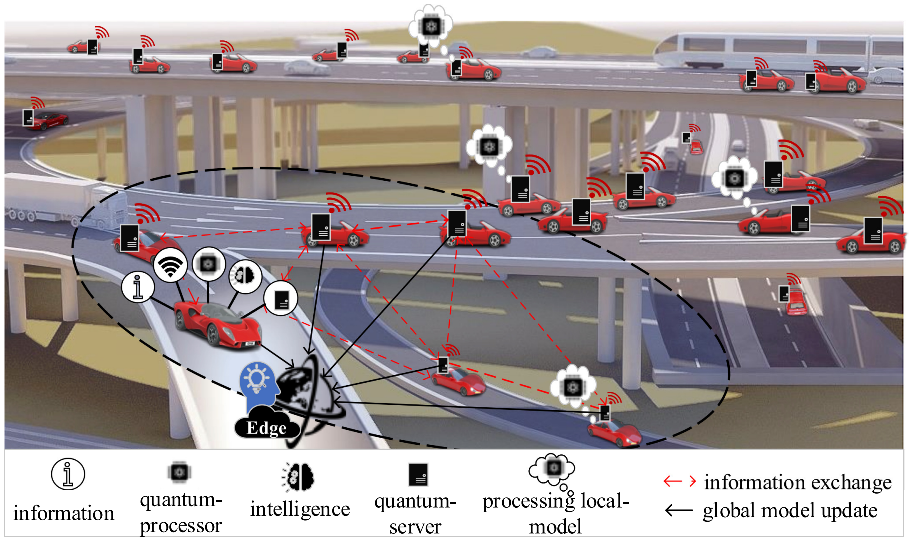

Fig. 1: An illustration of QV-FEDCOM based vehicular metaverse framework.

prediction problem, which serves as a comprehensive test case for the proposed model. At its core, this framework efficiently tackles several key challenges inherent in the dynamic environment of the next-generation vehicular QCNs: *a. Reducing communication costs, b. Handling data heterogeneity*, and *c. Optimizing memory consumption.*

Our research's key contributions are summarized as follows:

1) We introduce the quantum-based decentralized and heterogeneity-aware FL framework for vehicular metaverse (QV-FEDCOM framework) which innovatively merges quantum computing principles with FL, a novel approach that harnesses the strengths of both domains to optimize learning processes in a distributed vehicular metaverse.
2) We develop three key components within the QV-FEDCOM framework: Q-STP for reducing communication costs and ensuring robust model training by dynamically managing vehicle states; Q-VCG for scalable, context-based vehicle grouping to address data heterogeneity; and Q-PCA to compress memory and reduce the quantum memory footprint.
3) Subsequently, we integrate Q-STP, Q-VCG, and Q-PCA into the QV-FEDCOM framework, creating the QV-FEDCOM algorithm, which facilitates a decentralized, memory-efficient, and context-aware QFL process.
4) Additionally, incorporate a loss function specifically designed for trajectory prediction tasks named QTL which combines the Huber loss with an angular deviation penalty to handle both small and large errors robustly and penalize large deviations in the predicted trajectory angle.
5) Finally, we assess the effectiveness of our framework through simulations for vehicle trajectory prediction. The comparative performance analysis of the QV-FEDCOM algorithm with its various adaptations offers insightful benchmarks and highlights its capabilities within the framework's ecosystem.

Structure of the paper: The paper is structured as follows. Section II presents the system model. Section III discusses the proposed optimization method of the communication cost, followed by the description of handling contextual data in Section IV, and the Q-PCA program in Section V. Section VI outlines the design of the decentralized quantum-based federated learning algorithm, QV-FEDCOM. Section VII details the experimental setup, while Section VIII discusses the simulation results. Section IX discusses the complexity analysis and Section X concludes the paper.

II. System Model

In this study, we explore a suburban vehicular metaverse environment comprising K vehicles, as depicted in Fig. 1, equipped with intelligent agents, each outfitted with quantum processors, real-world data collection sensors, sophisticated AI functionalities, and state-of-the-art connectivity modules. Each of these vehicles are engaged in an intricate interplay of both learning and sharing. They consistently enhance their localized QNN models and occasionally synchronize with a comprehensive global model hosted on an edge server. Multiple edge servers, serving as aggregators, enable independent and concurrent global model updates. Our QV-FEDCOM framework operates over t rounds, each divided into r time slots. During these slots, vehicles refine models, apply the quantum parameter shift rule for gradient calculation, and communicate updates to the edge server. Communication decisions are optimized through mode indicators for adaptability and efficiency. The central server models are updated based on all vehicle inputs. Scheduled synchronizations, governed by R_{sync}, which denotes the synchronization interval or the frequency at which these synchronizations occur, guarantees consistent alignment with the global model for all vehicles, irrespective of their communication frequencies.

In QFL-based vehicular environments, communication cost is a critical factor. We focus on costs related to both local vehicle-to-vehicle communication and communication with the server. The communication cost depends on the distance between vehicle i and the edge server (d_i), the size of the transmitted quantum data (s_i), the available bandwidth (b), and the data transmission rate (e). The size of the transmitted data (s_i) directly impacts communication overhead, representing the data exchanged between vehicles and the server in each iteration. For a vehicle's quantum state, represented by qubits Q, each qubit is encoded using b_q bits, leading to a total size of $Q \times b_q$. Additionally, the quantum neural network (QNN) parameters P, each using b_p bits, contribute $P \times b_p$ to the update size, along with ancillary data A represented as b_a bits. Hence, the total quantum update size for the i^{th} vehicle is:

$$s_i = (Q \times b_q) + (P \times b_p) + b_a \, . \tag{1}$$

Next, the estimation of the transmission rate involves considering both local data processing within each vehicle and data transmission to the edge server. Following the same method used in (1), we define $(s_{i,local})$ and $(s_{i,server})$ as the size of the quantum data update required for local transmission and transfer to the server, respectively. The local transmission speed is denoted as v_{local}. Thus, the local transmission time and server transmission time are calculated, respectively, as

$$t_{i,local} = \frac{s_{i,local}}{v_{local}} \, , \tag{2}$$

$$t_{i,server} = \frac{s_{i,server}}{b} \, . \tag{3}$$

In this study, we mainly focus on the communication cost involved in the local communication among the vehicles and the communication cost incurred in server communication. Thereby, we estimate the overall communication cost as

$$CC_i = \texttt{LocalCC}_i + \texttt{ServerCC}_i \, . \tag{4}$$

Here, local and server communication costs are calculated, respectively, as

$$\texttt{LocalCC}_i = t_{i,local} \times C_{local} \, , \tag{5}$$

$$\texttt{ServerCC}_i = t_{i,local} \times C_{server} \times f(d_i) \, , \tag{6}$$

where C_{local} and C_{server} are constants that represent the cost per unit time for local and server communications, respectively. $f(d_i) = 1 + \alpha \times d_i$ is a linear function that represents the additional cost incurred due to the distance and α is a constant that represents the rate at which cost increases with distance.

III. Optimizing Communication Cost

In QFL, the primary challenge is the high communication cost due to the transmission of quantum data. This is exacerbated in vehicular networks where bandwidth is limited and connectivity can be intermittent. Efficient communication is essential to maintain model integrity while ensuring that the system remains scalable and robust.

To address these challenges, we propose the Quantum Sequential-Training-Program (Q-STP), which introduces two operational modes for vehicles within the FL process. These modes are designed to balance communication efficiency with model integrity dynamically:

1. Streaming Mode: In this mode, a vehicle actively participates in quantum streaming for FL. Quantum streaming involves continuously transmitting quantum-encoded data updates from the vehicle to the edge server in real time. This mode is beneficial when the communication cost is low, and the network conditions are favorable. By frequently sending the latest quantum data to the edge server, the global model can be updated more often, thereby maintaining high model accuracy and ensuring that the system promptly incorporates new information from the vehicles.

2. Calibration Mode: In this mode, a vehicle focuses on stabilizing its local model using quantum techniques without actively participating in quantum streaming. Stabilizing the local model involves optimizing and refining the model's parameters locally using the vehicle's data. This process enhances the model's performance and accuracy on the local dataset. Calibration mode is useful when communication costs are high or network conditions are poor. By allowing vehicles to independently improve their local models, this mode reduces the need for constant data transmission. Even though the local models are not frequently synchronized with the global model, they remain well-calibrated and effective due to ongoing local optimization efforts.

The proposed Q-STP optimizes communication decisions by dynamically adjusting these modes based on current network conditions and vehicle states. This approach provides several benefits: adaptive communication by reducing unnecessary overhead when conditions are poor, efficient resource utilization by allowing local model refinement when communication is expensive, and a balanced trade-off between model accuracy and communication costs by maintaining high model integrity while optimizing efficiency. The implementation of Q-STP in the QV-FEDCOM framework ensures that vehicles refine their models and communicate updates to the edge server based on the selected mode, with scheduled synchronizations to align with the global model for all vehicles. The optimization problem for minimizing communication costs in Q-STP is formulated to adjust the operational mode dynamically, ensuring efficient communication and robust model performance.

A. Problem Formulation for CC_i minimization

Next, we formulate the optimization problem, aiming to minimize CC_i, as:

$$
\begin{aligned}
(\mathcal{P}_1): \quad & \min_{M_i, C_i, W_i^{(t)}, U_i^{(t)}} \quad \sum_{i=1}^{N} CC_i \\
\text{s.t.} (C.1) \quad & M_i \in \{0,1\} \quad \forall i \, , \\
(C.2) \quad & CC_i \geq 0 \quad \forall i \\
(C.3) \quad & L_i(t) = U(L_i(t-1), D_i(t)) \text{ if } M_i = 1, \forall i \, ,
\end{aligned}
$$

$$
\begin{aligned}
&(C.4) \quad L_i(t) = S(L_i(t-1), D_i(t)) \text{ if } M_i = 0, \forall i\,, \\
&(C.5) \quad W_i^{(t)}, \text{ if } M_i = 1, \quad \forall i\,, \\
&(C.6) \quad U_i^{(t)}, \text{ if } M_i = 1 \text{ or } M_i = 0, \quad \forall i\,. \qquad (7)
\end{aligned}
$$

Here, in constraint $C.1$, M_i serves as a binary indicator for the quantum mode:

$$
M_i = \begin{cases} 1 & \text{if } d_i > \mathbb{T} \text{ and } CC_i \leq \text{Limits}_i \\ 0 & \text{otherwise} \end{cases} \tag{8}
$$

Here, $d_i > \mathbb{T}$ ascertains if the vehicle i is sufficiently distant from the server, surpassing threshold $\mathbb{T}$, to engage the quantum streaming mode. Concurrently, $CC_i \leq \text{Limits}_i$ ensures that vehicle i's communication cost adheres to the limit, $\text{Limits}_i = m$ mbps per vehicle × vehicle to bandwidth ratio × OHF, where OHF stands for the overhead factor. Both conditions must concur for M_i to be 1, indicating the vehicle's operation in the quantum streaming mode. Conversely, M_i defaults to 0, denoting calibration mode. Additionally, $C.2$ mandates non-negative communication costs. $C.3$ postulates that quantum updates through operation U are applied to the local model $L_i(t-1)$ of vehicle i at round t using data $D_i(t)$ solely when $M_i = 1$. Further, $C.4$ underscores that operation S is triggered when $M_i = 0$ (representing the calibration mode), $C.5$ and $C.6$ respectively elaborate on the nuances of quantum transmissions $W_i^{(t)}$ and quantum update exchanges $U_i^{(t)}$ contingent on the quantum mode.

B. RL-based Quantum Sequential-Training-Program (Q-STP)

To enhance the decision-making process in Q-STP, we integrate a decentralized reinforcement learning (RL) approach, where each vehicle acts as its own RL agent. This decentralized approach allows each vehicle to make real-time mode-switching decisions (between streaming and calibration modes) based on its unique communication conditions and network state. Using Deep Q-Networks (DQN), each vehicle learns the optimal mode-switching policy by minimizing communication costs while maintaining model accuracy.

1) State: The state space includes features such as communication cost, distance from the server, vehicle speed, and the current operational mode.

2) Action: The action space comprises three actions: switching to streaming mode, switching to calibration mode, or staying in the current mode.

3) Reward: The reward function is designed to balance communication costs and model accuracy, defined as:

$$
r_t = -\text{norm}(CC_i) + \text{norm}(\Delta\texttt{Acc})\,, \tag{9}
$$

where $\Delta\texttt{Acc}$ is the change in model accuracy and CC_i is the communication cost. Both terms are normalized to ensure comparability.

This decentralized setup ensures that the computational load remains minimal, as each vehicle processes its state-action pairs. Additionally, the decentralized RL setup minimizes communication overhead by allowing vehicles to make localized decisions, thus avoiding the need for frequent communication with the edge server. This mechanism effectively reduces quantum data transmission, which is particularly resource-intensive. The Q-values are updated using the DQN update rule:

$$
Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \nu \left(r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right) \tag{10}
$$

where ν is the learning rate, and γ is the discount factor. This method ensures that mode-switching decisions are made efficiently while reducing unnecessary communication. The Q-STP is detailed in **Algorithm 1**.

Algorithm 1 Q-STP for Communication Cost Minimization

```
Initialize parameters and hyperparameters.
Initialize $CC_i \leftarrow 0$ for each vehicle $i$.
Initialize mode indicators $M_i \leftarrow 0$ for each vehicle $i$.
Initialize agent with state, action, and reward function.
for each training round do
    for each vehicle $i$ do
        Calculate the state : $CC_i$, $d_i$, $s_i$, $M_i$.
        $s_i \leftarrow$ state_features
    end for
    for each vehicle $i$ do
        Select $\omega_i$ based on the policy derived for state $s_i$.
        If action $\omega_i$ is to switch mode:
        if $\omega_i =$ Streaming Mode then
            $M_i \leftarrow 1$
        else if $\omega_i =$ Calibration Mode then
            $M_i \leftarrow 0$
        end if
    end for
    for each vehicle $i$ do
        if $M_i = 1$ then
            $L_i(t) \leftarrow U(L_i(t-1), D_i(t))$.
        else
            $L_i(t) \leftarrow S(L_i(t-1), D_i(t))$.
        end if
    end for
    Reward Calculation:
    for each vehicle $i$ do
        Calculate change in accuracy $\Delta\texttt{Acc}$.
        Normalize $CC_i$ and $\Delta\texttt{Acc}$ using (??).
        Calculate reward $r_t$ using (9).
    end for
    Policy Update:
    for each vehicle $i$ do
        Update Q-values using (10).
    end for
end for
```

IV. Handling Contextual Data Heterogeneity

Although the Q-STP process helps reduce communication costs, data heterogeneity in QFL remains a critical challenge, particularly in dynamic vehicular networks where vehicles generate diverse data based on factors like speed, location, and bandwidth. This diversity leads to inefficient learning processes, as models trained on highly varied data might not generalize well across the entire network [4]. To address this, grouping vehicles with similar contextual data is essential. By creating groups with more homogeneous data, the learning process becomes more efficient, improving the quality of the trained models and enhancing the performance of the aggregated global model. This method effectively mitigates the negative impact of data heterogeneity, ensuring better overall learning outcomes.

A. Problem Formulation for Maximizing Data Similarity

The objective is to minimize data heterogeneity by optimally grouping vehicles based on their contextual data similarity. We define an optimization problem where the goal is to maximize the intra-cluster similarity by focusing on training vehicles based on data context similarity. Let us consider a binary variable Ω_{ij} to indicate indicates whether vehicles i and j are grouped. Then the optimization problem is given as

$$\begin{aligned}(\mathcal{P}_2):\quad &\max_{\Omega_{ij}} \sum_{i=1}^{V}\sum_{\substack{j=1\\ j\neq i}}^{V} \texttt{similarity\_score}_{(i,j)}\cdot\Omega_{ij}\\ \text{s.t.}(C.7)\quad &\Omega_{ij}\in\{0,1\}\quad \forall i\neq j,\forall(i,j)\in\{1,\ldots,K\},\\ (C.8)\quad &\Omega_{ij}=\Omega_{ji}\quad \forall(i,j)\in\{1,\ldots,K\},\\ (C.9)\quad &\sum_{\substack{j=1\\ j\neq i}}^{K}\Omega_{ij}=1,\forall i\in\{1,\ldots,K\},\\ (C.10)\quad &\Omega_{ii}=0,\forall i\in\{1,\ldots,K\},\end{aligned}\tag{11}$$

where the contextual data $=\{x-$coordinare, $y-$coordinate, d_i, s_i, e, $b\}$. Constraint $C.7$ ensures that Ω_{ij} is a binary variable. $C.8$ enforces symmetry in the clustering. This means if vehicle i is grouped with vehicle j (i.e., $\Omega_{ij}=1$), then vehicle j must also be grouped with vehicle i ($\Omega_{ji}=1$). This is necessary because the similarity relationship is mutual. $C.9$ guarantees that each vehicle is assigned to exactly one cluster. This constraint ensures that each vehicle is grouped with exactly one set of other vehicles, making the clustering process exhaustive. $C.10$ ensures that a vehicle is not grouped with itself. This prevents trivial solutions where each vehicle is considered its cluster.

B. Quantum Vehicular-Context-Grouping (Q-VCG)

To address vehicular data heterogeneity, Q-VCG optimizes groupings of vehicles based on contextual data similarity. These groupings allow for independent quantum training on different edge servers, improving the efficiency and scalability of the federated learning process. In the Q-VCG framework, edge servers handle the computation of vehicle groupings. Each edge server has access to the global contextual data of vehicles within its region, such as coordinates, speed, and bandwidth. This global view, combined with the server's computational resources, enables it to perform the complex tasks of hierarchical clustering and simulated annealing. The edge server calculates the cosine similarity between vehicles using min-max normalized contextual data and then groups vehicles accordingly. The Q-VCG utilizes edge servers to compute the grouping instead of considering vehicles as agents mainly because:

- Edge servers are equipped with greater computational resources compared to vehicles, enabling them to efficiently compute vehicle similarities and optimize group formations.
- Vehicles only have access to their local data, while the edge server has a complete view of all vehicles in its region, allowing it to make informed decisions about grouping.

After computing the optimal groups, the edge server distributes lightweight control messages to the vehicles, containing their group assignments for the next federated learning round. To gauge data homogeneity between vehicles i and j, we use the cosine similarity metric based on contextual data factors x-y-coordinates, d_i, s_i, e, b [32], [33]. However, these factors have different units and scales, which can skew the similarity calculation if not properly addressed. To ensure that all features contribute equally to the similarity measure, we apply min-max normalization to scale these features to a common range $[0,1]$. Min-max normalization is applied as:

$$V_n \leftarrow \frac{V_n-\min(V_n)}{\max(V_n)-\min(V_n)},\tag{12}$$

where V_n is the value of the feature for a given vehicle, such that x-y-coordinates, d_i, s_i, e, $b\in V_n$, $\min(V_n)$ is the minimum value of that feature across all vehicles, and $\max(V_n)$ is the maximum value of that feature across all vehicles. This transformation scales each feature such that the minimum value maps to 0 and the maximum value maps to 1, thereby ensuring that features with different units and scales contribute equally to the similarity calculation. After normalization, we calculate the importance of each feature using a random forest model. This process is summarized in **Algorithm 2**.

Algorithm 2 Random Forest-based Feature Importance

```
Input: Normalized dataset $\mathbf{X}$ with features $V_n$.
Output: Feature importance weights $w_n$.
Train a Random Forest model using $\mathbf{X}$.
Extract feature importance scores from the trained model.
Normalize the feature importance scores to sum up to 1.
Assign these normalized scores as weights $w_n$ for each feature $V_n$.
```

These weights adjust the similarity calculation to reflect each feature's importance. The weighted cosine similarity between vehicle i and j is computed as:

$$\texttt{cos\_sim}_{(i,j)} = \frac{\sum_{n=1}^{6} w_n V_{i,n} \cdot V_{j,n}}{\sqrt{\sum_{n=1}^{6} (w_n V_{i,n})^2} \cdot \sqrt{\sum_{n=1}^{6} (w_n V_{j,n})^2}}, \tag{13}$$

where w_n are the feature importance weights derived from the Random Forest model, and $V_{i,n}$ and $V_{j,n}$ represent the normalized n-th component of the feature vectors for vehicles i and j, respectively. This weighted cosine similarity effectively captures the similarity between the contextual data of two vehicles, taking into account the relative importance of each feature.

To ensure privacy, each vehicle computes a locally transformed version of its feature vector using the secure multi-party computation (SMC) protocol described in [34]. These transformed vectors are shared with a central aggregator, which computes the similarity scores without accessing the raw data. This approach ensures that the QV-FEDCOM framework adheres to federated learning's fundamental goal of data privacy while effectively managing data heterogeneity and optimizing the learning process.

Next, to achieve optimal clustering, we employ a two-step approach combining hierarchical clustering with an optimization step. This ensures the resulting clusters are as homogeneous as possible, reducing data heterogeneity within each cluster and improving the efficiency of the federated learning process.

1) Step 1: Hierarchical Clustering: We begin by normalizing the feature vectors and calculating the cosine similarity between each vehicle pair. Hierarchical clustering is then applied to create an initial set of clusters using Ward's method [35], which minimizes the variance within each cluster during the clustering process.

Hierarchical clustering begins with each vehicle in its cluster and iteratively merges the two clusters resulting in the smallest increase in within-cluster variance. The process uses squared Euclidean distance as the metric and continues until a specified number of clusters or a distance threshold is reached. At each step, the distance matrix is updated after merging the most similar clusters. The objective function in Ward's method is given by:

$$D(A, B) = \frac{|A||B|}{|A| + |B|} \|\mathbf{c}_A - \mathbf{c}_B\|^2 \tag{14}$$

where $D(A, B)$ is the distance between clusters A and B, $|A|$ and $|B|$ are the number of vehicles in clusters A and B respectively, and $\mathbf{c}_A$ and $\mathbf{c}_B$ are the centroids of clusters A and B.

2) Step 2: Optimization Step: After obtaining the initial clusters through hierarchical clustering, an optimization step is applied to refine these clusters. The goal is to maximize intra-cluster similarity while ensuring the clusters meet predefined constraints such as minimum cluster size and maximum number of clusters.

We employ simulated annealing (SA) to perform this optimization. SA is chosen for its ability to effectively search large solution spaces and avoid local optima. SA algorithm calculates the objective function as

$$OF = \sum_{\substack{i,j \in \text{same cluster} \\ i \neq j}} \text{cos\_sim}_{(i,j)}, \tag{15}$$

such that, the difference/change in OF over different time stamps is denoted as ΔE. Next, the acceptance probability of the SA algorithm is calculated as

$$P(\Delta E) = \begin{cases} 1 & \text{if } \Delta E > 0 \\ \exp\left(\frac{\Delta E}{G}\right) & \text{if } \Delta E \leq 0 \end{cases} \tag{16}$$

where G is a parameter analogous to temperature in traditional simulated annealing, which controls the likelihood of accepting worse solutions, such that $G_{t+1} = \zeta G_t$. Initial G starts as value 1 and the final value is achieved upon completion of the maximum iterations.

The optimization process involves the following steps:

1) The objective function to be maximized during optimization is calculated as (15).
2) The initial solution for the optimization algorithm is the set of clusters obtained from hierarchical clustering.
3) The SA algorithm iteratively modifies the cluster assignments to improve the objective function. The algorithm considers possible moves such as reassigning vehicles to different clusters or swapping vehicles between clusters. At each iteration, the algorithm evaluates the new configuration based on the objective function and decides whether to accept the new configuration based on a probabilistic acceptance criterion, which helps avoid local optima.
4) During the optimization process, constraints are enforced to ensure that the resulting clusters are valid. These constraints include:
    - Each cluster must contain at least K_{min} vehicles. If a cluster falls below this size, the vehicles are reassigned.
    - Each group should encompass at least K_{min} vehicles for efficient QFL training. If a vehicle i does not qualify to be in any group with more than K_{min} vehicles, vehicle i will not take part in the training process.
    - A vehicle can only belong to one cluster at a time.
    - For all the vehicles, vehicle i and vehicle j can only be in the same group if the value of $\texttt{cos\_sim}_{(i,j)}$ is greater than the similarity threshold ($\texttt{Sim\_Thres}$). This value is determined based on a specific percentile ($\mathbb{P}$) of all calculated cosine similarity values within that round. This helps the system to be more resilient against drastic changes in data distribution.

The Q-VCG algorithm is detailed in **Algorithm 3**.

V. Quantum-Inspired Principal Component Analysis (Q-PCA)

The Q-STP and Q-VCG significantly optimize communication efficiency and manage data heterogeneity. However, a critical challenge that remains unaddressed is

Algorithm 3 Q-VCG for Optimal Vehicular Grouping

```
Input: Normalized data matrix, similarity matrix, $K_{\text{min}}$, $\beta$, Sim_Thres
Output: Optimized clusters
Calculate cosine similarities using (13).
Perform hierarchical clustering using Ward's method.
Obtain initial clusters when the clusters are most homogeneous.
Initialize (15) as the sum of intra-cluster similarities.
Set the initial solution to the initial clusters.
Set initial parameters for the SA process.
while stopping criterion not met do
    Generate a new set of clusters.
    Evaluate the new set of clusters using (14).
    Calculate the change in the objective function $\Delta E$.
    if $\Delta E > 0$ then
        Accept the new set of clusters.
    else
        Accept the new cluster-set with probability
          proportional to $\Delta E$.
    end if
    Ensure all clusters meet the constraints:
    for each cluster do
        if cluster size $< K_{\text{min}}$ then
            Reassign vehicles to meet $K_{\text{min}}$.
        end if
    end for
    Ensure the total number of clusters $\mathcal{G}$ within limit.
    Ensure each vehicle belongs to only one cluster.
    Ensure vehicles in the same cluster have a cosine
      similarity $\geq$ Sim_Thres.
end while
return Optimized clusters
```

the quantum memory footprint. Quantum states, even when efficiently utilized, can quickly consume extensive memory resources, as each additional qubit doubles the state space. To sustain the scalability of the QV-FEDCOM framework and ensure its practical deployment, it is crucial to reduce quantum memory usage without compromising the integrity of the quantum data [36]. To address this limitation, we find inspiration in the classical computational method known as principal component analysis (PCA) [37]. PCA is typically used in classical computing to transform a set of possibly correlated variables into a set of linearly uncorrelated variables called "principal components."

Emulating this approach in the quantum computing domain, we conceptualize an analogous process for quantum datasets and denote it as Q-PCA [38]–[40]. The integration of Q-PCA occurs before the Q-STP and Q-VCG since it is designed to optimize the quantum data representation. This ensures that the data is already in an optimized form when it goes through the sequential training and vehicle-context grouping processes, thereby reducing the quantum memory load. Q-PCA technique was introduced in [41] as an effective technique for reducing the quantum memory footprint. Thus, the integration of Q-PCA in our framework helps reduce the quantum memory footprint, a critical factor for efficient QFL. By combining Q-PCA with components like Q-STP and Q-VCG, we address the challenges of data heterogeneity, communication costs, and memory optimization within QFL. This cohesive framework, QV-FEDCOM, showcases the innovative application of Q-PCA in a new context, enhancing overall system efficiency and scalability. We will ensure the manuscript is updated to reflect this clarity and emphasis on the integrated framework's contributions.

In the future, we aim to refine our approach to further reduce the quantum memory footprint and enhance the QV-FEDCOM framework in its next version with a more robust integration and optimization of Q-PCA.

To compress quantum data, the primary goal of integrating Q-PCA is to select a subset of principal quantum components that maximize the cumulative quantum variance [41], [42]. The quantum covariance matrix $\hat{\rho}$ is constructed from the quantum states representing the data, such that for N number of quantum states, the set of quantum states that encode the data can be denoted as $\{|\psi_i\rangle\}_{i=1}^{N}$. Thus, this matrix is constructed from the outer products of the quantum states $|\psi_i\rangle$, mathematically, given as

$$\hat{\rho} = \frac{1}{N} \sum_{i=1}^{N} |\psi_i\rangle\langle\psi_i| \,. \tag{17}$$

This matrix captures the statistical properties of the quantum states representing the data. Next, we find the eigenstates $|\phi_q\rangle$, which are the principal components in Q-PCA. These eigenstates are the key to reducing the dimensionality of the quantum data, and they represent the most significant components of the dataset in terms of quantum variance. For each $|\phi_q\rangle$ of $\hat{\rho}$, its quantum variance is calculated as

$$\text{Var}(|\phi_q\rangle) = \langle\phi_q|\hat{\rho}^2|\phi_q\rangle - (\langle\phi_q|\hat{\rho}|\phi_q\rangle)^2 \,. \tag{18}$$

The process of estimating the variance involves applying $\hat{\rho}$ and $\hat{\rho}^2$ to each $|\phi_q\rangle$ and then measuring the expectation values. Therefore, the optimization problem aims to maximize:

$$\begin{aligned}
(\mathcal{P}_3): \quad & \max_{\{|\phi_q\rangle\}} \quad \sum_{q=1}^{Q'} \text{Var}(|\phi_q\rangle) \\
\text{s.t.}\ & (C.11) \quad Q' < Q, \\
& (C.12) \quad \text{The eigenstates } \{|\phi_q\rangle\} \text{ are orthogonal}, \\
& (C.13) \quad F(\rho, \sigma) \geq \xi,
\end{aligned}$$

where Q is the original number of qubits and Q' represents the number of principal components selected. This ensures that the resulting quantum state is a compressed version of the original. $F(\rho, \sigma)$ is the fidelity between the original quantum state ρ and the compressed quantum state σ. ξ is the predefined threshold for fidelity. This constraint ensures that the fidelity between the original and compressed states is not below this threshold, indicating that the compressed state retains a high degree of similarity to the original state.

Thus, the Q-PCA algorithm is designed to optimize and solve $\mathcal{P}_3$. The Q-PCA algorithm is described in detail in **Algorithm 4.**

Algorithm 4 Q-PCA for Quantum Data Compression

```
Prepare quantum states $\{|\psi_i\rangle\}$ from classical data.
Initialize the quantum covariance matrix $\hat{\rho}$.
for each quantum state $|\psi_i\rangle$ do
    Update $\hat{\rho}$ by adding $|\psi_i\rangle\langle\psi_i|$.
end for
Normalize $\hat{\rho}$ by the number of states $N$.
Compute eigenstates $\{|\phi_q\rangle\}$ of $\hat{\rho}$.
for each eigenstate $|\phi_q\rangle$ do
    Calculate Var($|\phi_q\rangle$).
end for
Select a subset of eigenstates $\{|\phi_q\rangle\}$ that maximize cumulative variance.
Transform quantum data to the basis of selected principal components.
```

VI. Decentralized QFL For Vehicular Metaverse

We incorporate the Q-STP, Q-VCG, and Q-PCA algorithms and design a quantum-based decentralized and heterogeneity-aware FL framework for vehicular metaverse named QV-FEDCOM framework. In this framework, the QV-FEDCOM algorithm is designed to handle data heterogeneity in a cost-efficient way while maintaining a robust quantum environment.

In the QV-FEDCOM framework, each vehicle i has a local model (L_i) initialized. This model is represented by the parameterized quantum circuit $L_i(\boldsymbol{\theta}_i) = U_{\text{QNN}}(\boldsymbol{\theta}_i)$ in the context of QNN, where $\boldsymbol{\theta}_i$ represents the set of trainable parameters for vehicle i and U_{QNN} denotes the parameterized quantum unitary (circuit) for the QNN of vehicle i. Next, updating the local model involves quantum circuits with parameter shifts for gradient calculations given as

$$\Delta\boldsymbol{\theta}_i = -\eta\nabla\mathcal{L}_{\text{local}}(L_i)\,, \tag{19}$$

$$\boldsymbol{\theta}_{i\text{new}} = \boldsymbol{\theta}_i + \Delta\boldsymbol{\theta}_i\,, \tag{20}$$

where $\nabla\mathcal{L}_{\text{local}}(L_i)$ is the gradient of the local loss concerning the parameters of the QNN, η is the learning rate and $\boldsymbol{\theta}_{i\text{new}}$ are the updated parameters after a given update step. Next, we derive the local gradients by applying (24), which is derived using *Theorem* 1 provided below:

Theorem 1 (Quantum Parameter Shift Rule): When dealing with a quantum circuit that employs a loss function $\mathcal{L}_{\text{local}}(\theta)$, we can approximate the gradient concerning the parameter θ_{in} as:

$$\frac{\partial\mathcal{L}_{\text{local}}}{\partial\theta_{in}} \approx \frac{\mathcal{L}_{\text{local}}(\theta_{in}+\frac{\pi}{2}) - \mathcal{L}_{\text{local}}(\theta_{in}-\frac{\pi}{2})}{2}. \tag{21}$$

Proof: Starting with the general concept of a derivative:

$$\lim_{h\to 0}\frac{f(x+h)-f(x)}{h} = f'(x). \tag{22}$$

Algorithm 5 QV-FEDCOM framework

```
Initialization:
Initialize global parameters and hyperparameters.
Initialize $\mathcal{G}$ empty lists for grouped vehicles.
Initialize global models $M_G$ for each group.
Initialize local models $L_i$ for each vehicle $i$.
Initialize $CC_i$ and mode indicators $M_i$ for each vehicle $i$.
Perform Q-PCA to compress quantum data.
Main Loop:
for each round do
    Calculate vehicle dynamics using Q-STP.
    Form vehicle groups using Q-VCG.
    for each group $G_g$ do
        Initialize the edge server for group $G_g$.
        Sync the global model $M_{G_g}$ with the edge server.
        for each time slot $r$ do
            for each vehicle $i$ in group $G_g$ do
                Update the local model $L_i$.
                Calculate local gradients.
                if $M_i = 1$ then
                    Send gradients to the edge server.
                    Update $M_{G_g}$ with received gradients.
                    Send model update to the vehicle $i$.
                end if
            end for
            if $r \mod R_{\text{sync}} = 0$ then
                for each vehicle $i$ in group $G_g$ do
                    if $M_i = 0$ then
                        Send gradients to the edge server.
                        Update global model $M_{G_g}$.
                        Send model update to entire group.
                    end if
                end for
            end if
        end for
    end for
    function CONVERGENCE($G_g$, $\mathcal{L}_p$, counter)
        Calculate $\mathcal{L}_c$ for group $G_g$.
        $\Delta\mathcal{L} = (\mathcal{L}_p - \mathcal{L}_c)/\mathcal{L}_p$
        if $\Delta\mathcal{L} < \mathcal{W}$ then
            counter ← counter + 1
            if counter $\geq$ Y then
                return True
            else
                return False
            end if
        else
            counter ← 0
            return False
        end if
    end function
end for
```

From the central finite difference approximation:

$$\frac{\partial f(x)}{\partial x} \approx \frac{f(x+h)-f(x-h)}{2h}. \tag{23}$$

By applying (23) with $h = \frac{\pi}{2}$, specifically valid for parameterized quantum circuits due to their unitary nature [43]:

$$\frac{\partial \mathcal{L}_{\text{local}}}{\partial \theta_{in}} \approx \frac{\mathcal{L}_{\text{local}}(\theta_{in} + \frac{\pi}{2}) - \mathcal{L}_{\text{local}}(\theta_{in} - \frac{\pi}{2})}{2}. \tag{24}$$

■

Next, the global model is aggregated using the gradient:

$$\Delta \boldsymbol{\theta}_{G_g} = -\eta \nabla \mathcal{L}_{\text{global}}(M_{G_g})\,, \tag{25}$$

$$\boldsymbol{\theta}_{G_g\text{new}} = \boldsymbol{\theta}_{G_g} + \Delta \boldsymbol{\theta}_{G_g}\,, \tag{26}$$

where $\boldsymbol{\theta}_{G_g}$ is the set of trainable parameters for the global model of the group G_g. The step $r \bmod R_{sync} = 0$ ensures that vehicles that do not send updates in every time slot can still sync with the global model periodically, where R_{sync} is the synchronization time.

For this study, we define the loss function as quantum trajectory loss (QTL) to enhance the prediction accuracy of vehicle trajectories. The QTL comprises two components: the Huber loss [44] and the angular deviation penalty. The Huber loss is defined as:

$$\mathcal{L}_{Hub} = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2 & \text{for } |y_i - \hat{y}_i| \leq \delta \\ \delta\left(|y_i - \hat{y}_i| - \frac{\delta}{2}\right) & \text{otherwise}\,, \end{cases} \tag{27}$$

where δ is a hyperparameter that determines the threshold at which the loss transitions from a quadratic function to a linear function. x_i and y_i are the actual coordinates of the vehicle i, while $\hat{x}_i$ and $\hat{y}_i$ are the predicted coordinates at the same time.

To address the directionality aspect of vehicle trajectories, we introduce the *angular deviation penalty*. Similar techniques to minimize angular deviations have been explored in fields like robotics [45] and autonomous driving systems [46], [47], where trajectory alignment is crucial. In our framework this term penalizes deviations in the predicted trajectory's direction, ensuring the predicted trajectory aligns not only with the position but also the direction of the actual trajectory. This penalty is especially important in vehicular applications where directional accuracy is critical for tasks like navigation and collision avoidance. Mathematically, the angular deviation penalty is expressed as:

$$\mathcal{L}_{\text{Ang}} = \sum_{t=1}^{T} \theta(t)\,, \tag{28}$$

where angular deviation $\theta(t)$ represents the difference in trajectory direction between the predicted and true paths, as:

$$\theta(t) = \left| \tan^{-1}\left(\frac{\hat{y}_i - y_i}{\hat{x}_i - x_i}\right) - \tan^{-1}\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right) \right|. \tag{29}$$

The combination of Huber loss and angular deviation penalty is key in optimizing both positional accuracy and trajectory directionality. Finally, the QTL is estimated as:

$$\mathcal{L}_{QTL} = \mathcal{L}_{Hub} + \mathcal{L}_{Ang}\,. \tag{30}$$

In **Algorithm 5**, the QV-FEDCOM algorithm determines convergence dynamically through the reduction rate of the loss value. Let $\mathcal{L}_p$ and $\mathcal{L}_c$ represent the previous and current loss values, respectively, calculated from (30). The percentage decrease, $\Delta\mathcal{L}$, is defined as $\Delta\mathcal{L} = \frac{\mathcal{L}_p - \mathcal{L}_c}{\mathcal{L}_p}$. Convergence is deemed achieved when $\Delta\mathcal{L}$ falls below a set threshold $\mathcal{W}$ for Y consecutive rounds, employing a counter mechanism for tracking.

VII. Experiment

This section outlines our experimental setup.

A. Problem Description

To evaluate the effectiveness of our QV-FEDCOM framework, we focus on a vehicular trajectory prediction problem, which is fundamentally a regression task. The goal is to predict continuous-valued outputs, specifically the future x and y coordinates of each vehicle. These predictions are real numbers, distinguishing this problem from classification tasks that predict discrete categories. The prediction is modeled as:

$$(\hat{x}(t), \hat{y}(t)) = f(\text{contextual\_data}), \tag{31}$$

where $\hat{x}(t)$ and $\hat{y}(t)$ are the predicted coordinates at time t, and $f(\cdot)$ represents the learned model that maps the vehicle's contextual data—including x and y coordinates, speed, distance traveled, transmission rate, and available bandwidth—into the predicted trajectory. The continuous nature of these predictions, which smoothly vary over time, aligns the task with regression rather than classification.

The model learns a mapping from the contextual features to the continuous output space of vehicle positions, generating predictions over a continuous domain. This process of predicting real-valued coordinates is inherently incompatible with classification approaches, further establishing that the task falls under the category of regression. For evaluating the performance of the quantum-based regression model, we adopt a *tolerance-based accuracy metric*, which assesses how close the predicted x and y coordinates are to the true values within a predefined error margin. This metric is suitable for continuous predictions, as it measures the proportion of predictions that fall within an acceptable tolerance range of the actual coordinates. Let the predicted position of the vehicle at time t be $(\hat{x}(t), \hat{y}(t))$, and the true position be $(x(t), y(t))$. The error between the predicted and true coordinates is computed using the Euclidean distance:

$$e(t) = \sqrt{(\hat{x}(t) - x(t))^2 + (\hat{y}(t) - y(t))^2}. \tag{32}$$

We define a tolerance threshold ϵ to determine how much deviation is acceptable. For instance, $\epsilon = 0.10$ (10% tolerance) defines the range of acceptable prediction errors. A prediction is considered accurate if the error $e(t)$ is within the tolerance threshold $e(t) \leq \epsilon$. If this condition is satisfied, the prediction is counted as accurate. The accuracy is then calculated as the proportion of predictions that meet the accuracy condition over the training rounds:

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} \mathbb{1}(e_i(t) \leq \epsilon)}{N}, \tag{33}$$

where $\mathbb{1}(\cdot)$ is an indicator function that equals 1 if $e_i(t) \leq \epsilon$, and 0 otherwise, and N is the total number of predictions made during a given training round.

B. Dataset Preparation

For our experiments, we utilize the Didi Chuxing GAIA Initiative Dataset[1], which provides comprehensive and high-quality vehicular trajectory data. This dataset includes information on vehicle coordinates, speed, and timestamps, making it suitable for our trajectory prediction task. The dataset contains ride-hailing orders of Didi Chuxing in Haikou, China, from May 1st to October 31st, with a total number of $12,185,427$ orders [48]–[50]. To enhance our dataset for the QV-FEDCOM framework, we simulate additional vehicular attributes such as transmission rate and available bandwidth. We use the OpenAI Gym library [51] to simulate these attributes, ensuring realistic and varied data for our experiments:

- **Transmission Rate:** Simulated based on typical vehicular communication scenarios, assuming a normal distribution with a mean of 10 Mbps and a standard deviation of 2 Mbps.
- **Available Bandwidth:** Simulated using common bandwidth availability patterns in vehicular networks, assuming a normal distribution with a mean of 50 Mbps and a standard deviation of 10 Mbps.

The preprocessing steps involve the following steps:

1) **Normalization:** All features are normalized to ensure they are on a comparable scale. This is crucial for calculating similarity scores accurately.
2) **Quantum Encoding:** The normalized data is encoded into quantum states using a sequence of Ry rotation gates. Each vehicular attribute undergoes encoding into a quantum state via a rotation angle, θ, set relative to the attribute value.

C. Experimental Setup

We set up a quantum environment using Quantum Toolbox [52], TFQ, and an FL setup using TensorFlow Federated (TFF) in Python, utilizing the GPU runtime on Google Colab Pro [19], [30]. Given the complexity of quantum computations, we use a modest dataset of 100 vehicles with 6 features, encoded into 100 quantum circuits, each operating on 6 qubits.

Each vehicular attribute is represented by a qubit, and R_y rotation gates are used to encode the normalized feature values into quantum states. Specifically, the state for each vehicle is initialized to $|0\rangle^{\otimes 6}$ and transformed using R_y gates: $|\psi\rangle = \bigotimes_{i=1}^{6} R_y(\theta_i)|0\rangle$ where $\theta_i = \kappa \times \text{FeatureValue}_i$ and κ is a normalization constant. κ is used to scale the vehicular attributes when encoding them into quantum states and is determined by the maximum value among all vehicular attributes, such that the largest feature value corresponds to a rotation angle of π. This is calculated as: $\kappa = \frac{\pi}{\text{max\_value}}$ where max_value is the maximum value of the vehicular attributes across all vehicles. CNOT gates are used to entangle qubits and capture correlations between features, and measurement gates are applied to extract classical information from the quantum states.

TABLE I: Simulation Parameters.

Param.	Value	Param.	Value
$\mathbb{T}$	0.75 km	t	50
$\mathbb{H}$	5	r	20
S_Epochs	5	R_{sync}	10
ν	0.01	$\mathcal{W}$	0.1
γ	0.99	η	0.01
δ	1	$\mathbb{P}$	80
C_Epoch	2	α	0.25
β	0.2	Y	5
$K_{\min}$	5	κ	$[0, 1]$
OHF	1.2	Q	6
v_{local}	100Mbps	b_a	100bits
K	100	m	0.5
N	100	ξ	0.95
d_i	$[50 - 90]$m	b	$[20 - 50]$Mbps
b_q	64bits	b_p	64bits
C_{local}	0.05	C_{server}	0.10
G	1	ζ	0.95

Given that our experiments occur in a simulated environment on a conventional computer, and the computational complexity can rise exponentially, we opt to train a compact QNN architecture. For each vehicle, the QNN architecture consists of an input layer of 6 qubits, two hidden layers with 3 qubits each, and an output layer designed to provide continuous values corresponding to the predicted future coordinates of the vehicle. This setup ensures that our quantum computations are feasible within the constraints of the Google Colab Pro environment while providing sufficient complexity to accurately model the vehicular trajectory prediction task. The other parameters used in this study are given in **Table I**.

D. Benchmark Comparisons

*1) **QFL**:* In benchmark comparisons, QFL represents a fundamental QFL model that operates with a single central server. Unlike the QV-FEDCOM framework, it lacks enhancements such as Q-STP, Q-VCG, and Q-PCA, and does not utilize a composite loss function, making it a baseline model for assessing the advanced features and benefits of the QV-FEDCOM framework.

2) QV-FEDCOM (without QTL): This benchmark variant of QV-FEDCOM excludes the QTL function and utilizes only Huber loss instead. It serves to demonstrate the impact of the QTL on the performance and accuracy of trajectory predictions within the framework.

[1] https://gaia.didichuxing.com/

3) QV-FEDCOM (without Q-VCG): This version of QV-FEDCOM operates without the Q-VCG mechanism, highlighting the role of Q-VCG in managing data heterogeneity and optimizing the federated learning process.

4) QV-FEDCOM (without Q-STP): In this configuration, the QV-FEDCOM framework is tested without the Q-STP. This benchmark helps to evaluate the significance of Q-STP in improving communication efficiency and overall training performance.

5) QV-MetaFL: QV-MetaFL is a previous version of QV-FEDCOM that also incorporates Q-STP and Q-VCG described in [53]. However, in QV-MetaFL, the Q-STP employs a heuristic approach for mode-switching rather than reinforcement learning, relying on predefined rules to manage the training sequence. Similarly, the Q-VCG in QV-MetaFL uses a heuristic hierarchical clustering approach to group vehicles, as opposed to the more advanced methods used in QV-FEDCOM. This comparison showcases the advancements and optimality of each component in the QV-FEDCOM framework.

6) ***Data heterogeneity levels****:* For our experimentation, we manually curated three subsets from the main dataset of 100 vehicles, as detailed in Section VII-B. These subsets are designed to represent three distinct levels of data heterogeneity. heterogeneity level 1 is composed of data with largely similar vehicle properties, providing a more homogeneous dataset. In contrast, heterogeneity level 2 includes a broader diversity of vehicle characteristics, and heterogeneity level 3 encompasses a very diverse range of vehicle properties, representing the highest degree of heterogeneity. Unless specifically mentioned, most of our experiments default to using the dataset corresponding to heterogeneity level 2.

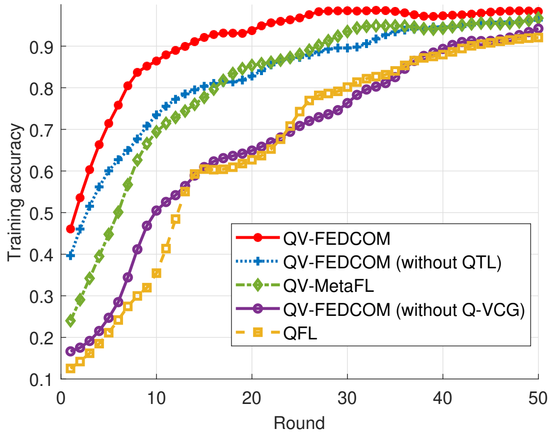

Fig. 2: Training accuracy vs. rounds.

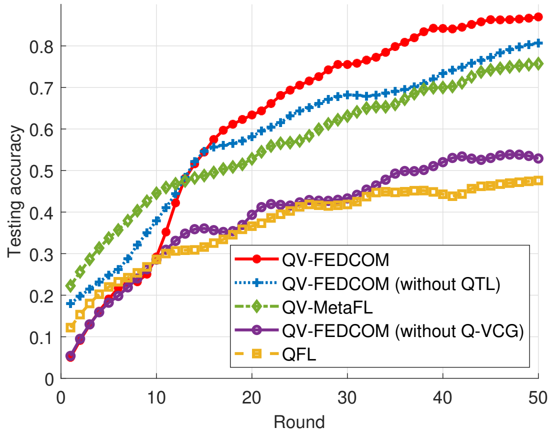

Fig. 3: Testing accuracy vs. rounds.

VIII. Simulation Results

This section presents the results of the performance evaluation of the proposed framework-

A. Training Accuracy

Fig. 2 illustrates the training accuracy of various frameworks across multiple training rounds. The QV-FEDCOM framework, which integrates DQN-based Q-STP, simulated annealing-based Q-VCG, Q-PCA, and QTL, demonstrates the highest training accuracy throughout the rounds. This performance underscores the effectiveness of these integrated components in addressing data heterogeneity and optimizing training processes. Initially, the QV-FEDCOM (without QTL) shows a comparable performance to the full QV-FEDCOM. However, as the training progresses, the gap widens, with the QV-FEDCOM maintaining a higher accuracy, emphasizing the role of the angular deviation penalty in the QTL loss function in refining the learning process and handling diverse data more robustly. The QV-FEDCOM (without Q-VCG) exhibits a slower start and lower overall accuracy, particularly in the early rounds. This can be attributed to the lack of vehicle-context grouping, which is crucial for reducing data heterogeneity and ensuring homogeneous training groups. This version's performance, although improved over time, remains below the full QV-FEDCOM. The QV-MetaFL, which employs heuristic methods for Q-STP and Q-VCG, starts moderately but quickly falls behind the dynamic and optimized QV-FEDCOM. This highlights the limitations of heuristic approaches compared to the more sophisticated and adaptive methods used in QV-FEDCOM. The baseline QFL model consistently shows the lowest accuracy, reflecting the absence of advanced mechanisms like DQN-based dynamic mode switching and context-aware clustering.

B. Testing Accuracy

Fig. 3 shows the testing accuracy of various frameworks over multiple rounds. QV-FEDCOM achieves the highest accuracy quickly, demonstrating its robustness. QV-FEDCOM (without QTL) has slower improvement and lower final accuracy, highlighting the impact of the QTL loss function. QV-FEDCOM (without Q-VCG) starts with lower accuracy and remains below the full version, emphasizing the importance of vehicle-context grouping. QV-MetaFL, using heuristic

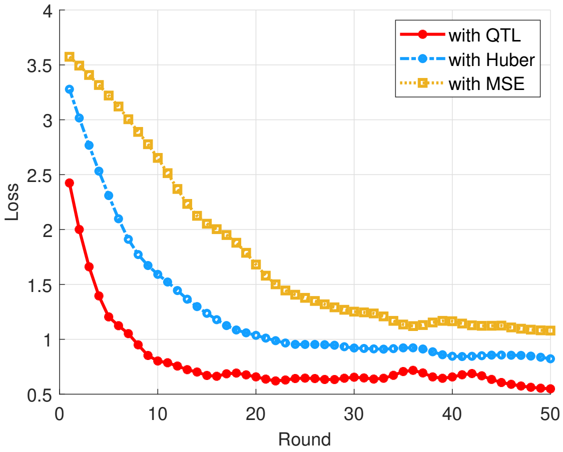

Fig. 4: Loss vs. rounds.

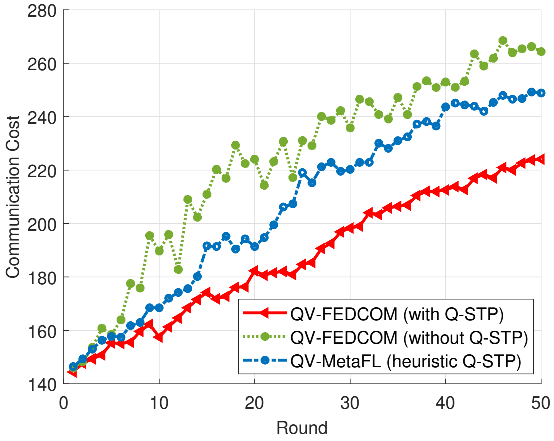

Fig. 5: Impact of Q-STP on communication cost.

methods, performs moderately but falls behind QV-FEDCOM in later rounds. The baseline QFL model lags throughout, showing significant improvements from Q-STP, Q-VCG, and QTL in QV-FEDCOM. This comparative analysis reinforces the superior performance and effectiveness of QV-FEDCOM in handling real-world vehicular data for federated learning, particularly in achieving high accuracy more quickly and maintaining robust performance across rounds.

C. *Loss*

Fig. 4 presents the loss values of different loss functions (QTL, Huber, and MSE) over multiple training rounds. The QTL, which combines the Huber loss with an angular deviation penalty, shows a steep decline in loss values early on and maintains the lowest loss throughout the training rounds. This behavior indicates that the QTL effectively handles both small and large errors while penalizing angular deviations, leading to more accurate trajectory predictions. The rapid convergence to lower loss values demonstrates the robustness and efficiency of the QTL in optimizing the model's performance. In contrast, the Huber loss function, while performing better than MSE, does not achieve the same level of loss reduction as QTL. The Huber loss is designed to handle outliers more effectively than MSE, resulting in a more stable decrease in loss values. However, without the angular deviation penalty, it cannot match the precision offered by QTL for trajectory predictions, as evident from its higher final loss values compared to QTL. The MSE loss function exhibits the highest loss values throughout the training rounds. MSE is highly sensitive to outliers, which can lead to significant errors in trajectory predictions. This sensitivity results in slower convergence and higher overall loss, making it less effective for the complex task of trajectory prediction in the vehicular metaverse. The comparative analysis demonstrates the superior performance of the QTL in minimizing loss values and improving model accuracy. The integration of angular deviation penalty within the QTL framework significantly enhances its ability to handle diverse data, leading to better overall performance.

It is essential to mention that the presented testing accuracy and loss metrics are relatively conservative. Such results stem from our experimental constraints, where a confined part of the dataset was employed alongside a limited number of rounds. These constraints arose since our conventional processors face challenges in managing the extensive computations tied to quantum procedures. Despite these limitations, the main goal was to gauge the comparative efficiency of our model. The primary QV-FEDCOM model's standout performance, even within these confines, affirms its promise and establishes its proficiency in quantum-centric learning scenarios.

D. *Communication Cost*

Fig. 5 illustrates the communication cost over multiple rounds for different approaches: QV-FEDCOM (with Q-STP), QV-FEDCOM (without Q-STP), and QV-MetaFL (heuristic Q-STP). The QV-FEDCOM framework with Q-STP demonstrates the lowest and most stable communication cost across all training rounds. This outcome highlights the effectiveness of the RL-based dynamic mode switching mechanism in optimizing data transmission. By selecting operational modes intelligently, Q-STP minimizes unnecessary transmissions, ensuring efficient use of network resources. In contrast, QV-FEDCOM (without Q-STP) shows significantly higher and more fluctuating communication costs. Without the dynamic mode switching provided by Q-STP, this approach fails to adapt to varying network conditions, leading to inefficiencies and increased communication overhead. The fluctuation in costs underscores the importance of Q-STP for maintaining low and consistent communication expenses. QV-MetaFL, which uses a heuristic approach for Q-STP, performs better than QV-FEDCOM (without Q-STP) but still incurs higher costs than the RL-based Q-STP. The heuristic method, while providing some optimization, lacks the adaptability and precision of the RL-based dynamic mode switching, resulting in less efficient communication cost management.

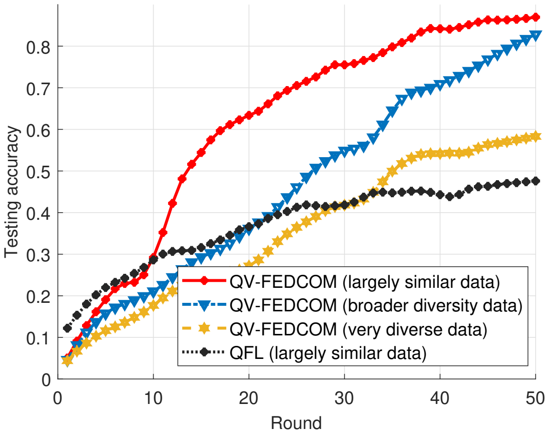

Fig. 6: Impact of different data heterogeneity levels on the accuracy.

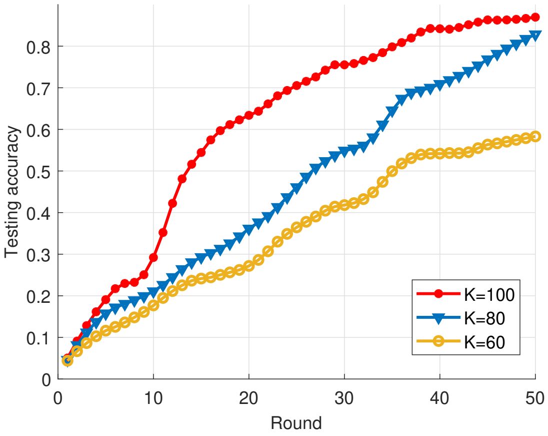

Fig. 7: Impact of number of vehicles on accuracy.

E. Data Heterogeneity

Fig. 6 illustrates the testing accuracy of the QV-FEDCOM framework across three distinct levels of data heterogeneity, compared to the basic QFL model with the least heterogeneous data ("largely similar" data). These levels represent varying degrees of diversity in the vehicles' contextual data, which are quantified using the normalized variance of the feature sets:

- **Largely Similar Data**: This dataset represents the least heterogeneous data, where vehicles have very similar contextual features. The normalized variance across all features is low (variance $\leq$ 0.1), indicating minimal diversity in vehicle speed, position, and other factors. In this case, the dataset is more homogeneous, leading to higher testing accuracy as the model can generalize more effectively to the vehicles' similar data points.
- **Broader Diversity Data**: This dataset presents moderate heterogeneity, with a higher degree of variation in contextual features among vehicles. The normalized variance is in the range of $0.1 <$ variance ≤ 0.3. This increased variation makes learning more complex, but the QV-FEDCOM framework still maintains relatively high accuracy due to the Q-VCG mechanism's ability to form effective vehicle groups.
- **Very Diverse Data**: This dataset represents the most diverse and heterogeneous data, where vehicle data is highly varied with a normalized variance greater than 0.3. The contextual features show significant variation, making the learning process more challenging. While the accuracy decreases due to this increased complexity, QV-FEDCOM still outperforms the baseline models, demonstrating its robustness even in highly dynamic environments.

As heterogeneity increases from "largely similar" to "very diverse" data, a decline in testing accuracy is observed for the QV-FEDCOM model. This trend reflects the increased complexity and challenges of learning from a more heterogeneous dataset. Despite this, QV-FEDCOM maintains relatively high accuracy across all heterogeneity levels, showcasing its robustness in handling data diversity.

The QV-FEDCOM with the least data heterogeneity ("largely similar") achieves the highest testing accuracy, followed by "broader diversity" and then "very diverse" data. Interestingly, there is a tradeoff between QV-FEDCOM with "very diverse" data and QFL with "largely similar" data, where QFL initially achieves high accuracy but fails to make further improvements, eventually being surpassed by QV-FEDCOM. It is to be noted that QV-FEDCOM even with the most complex data which is "very diverge" data outperforms the traditional QFL's accuracy with the simplest ("largely similar") data. This highlights the ability of QV-FEDCOM to manage diverse data contexts, even in highly dynamic vehicular networks.

F. Impact of the Number of Vehicles on Accuracy

Fig. 7 demonstrates the training accuracy of the QV-FEDCOM learning system with varying numbers of participating vehicles, denoted by K. The vehicles are engaged in a distributed learning process, and the graph shows that as K increases from 60 to 100, there is a corresponding improvement in accuracy. This pattern can be attributed to the greater variety and larger quantity of data provided by an expanded pool of vehicles. It implies that the model can generalize better when it has access to more varied and extensive data.

The enhanced performance with larger K values suggests that the QV-FEDCOM framework is particularly well-suited to environments with high vehicle density. In such settings, the framework can leverage extensive data to improve learning outcomes, indicating that it is optimized for large-scale deployments rather than smaller ones. This could be due to the framework's ability to handle increased complexity and variability in the data. This is more likely as the number of vehicles, along with the different kinds and amount of data they contribute, increases.

G. Impact of Vehicle Cluster Constraints on QV-FEDCOM

Fig. 8 presents the testing performance of the QV-FEDCOM framework, specifically analyzing the impact of

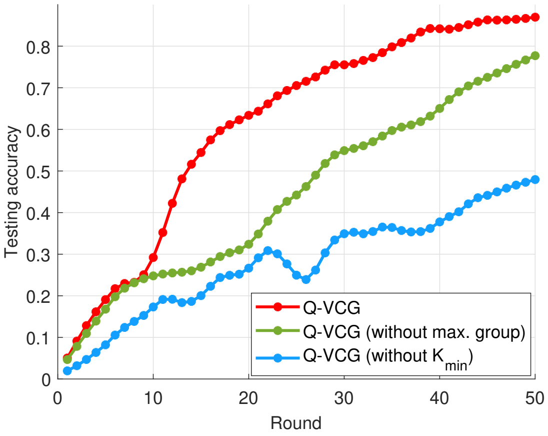

Fig. 8: Impact of Q-VCG's constraints on QV-FEDCOM.

group formation constraints within the Q-VCG component on the overall system. The graph compares three scenarios: (i) the standard Q-VCG with both the maximum number of groups, i.e., $\mathcal{G}$ and the minimum number of vehicles per group required to participate in the learning process, i.e., K_{min}, (ii) Q-VCG without limiting the maximum number of group formation, and (iii) Q-VCG without setting the K_min constraint. The results indicate that without the maximum group constraint, there is a propensity to form a larger number of small and potentially inefficient groups. This can lead to fragmented learning efforts where individual or very small groups of vehicles perform independent learning, which may not be effective due to the limited and possibly less diverse data. Such small group sizes do not capture enough context variability, leading to a learning process that may not generalize well, as evidenced by the lower accuracy in this scenario. On the other hand, imposing a K_{min} constraint ensures that only groups with a sufficient number of vehicles participate in the learning process. This minimizes resource wastage and avoids the participation of groups with negligible similarity context in the learning process. As shown in the graph, the version of Q-VCG that enforces both constraints achieves higher accuracy, demonstrating the importance of these constraints in maintaining meaningful group formations and ensuring effective FL across the network.

H. Memory Usage

Table II lists the memory usage of Google Colab Pro in gigabytes (GB) for various setups of the federated learning framework. Configurations without Q-PCA generally consume more memory, while those with Q-PCA show reduced memory usage. The specific configurations compared are QV-FEDCOM (without and with Q-PCA), QV-MetaFL (without and with Q-PCA), and QFL.

IX. Complexity Analysis

To assess the computational efficiency of the QV-FEDCOM framework, we analyze the complexity of each component and their interactions as follows:

TABLE II: Memory Consumption for Different Configurations.

Configuration	**Memory Consumption (GB)**
QV-FEDCOM (without Q-PCA)	15.93
QV-FEDCOM (with Q-PCA)	13.04
QV-MetaFL (without Q-PCA)	14.64
QV-MetaFL (with Q-PCA)	12.32
QFL	13.67

1) Q-STP complexity: For the Q-STP, the complexity of mode decision and switching, along with counter updates for each vehicle in every round, is constant, being denoted by $\mathcal{O}(1)$. Across all vehicles and rounds, this complexity becomes $\mathcal{O}(K \cdot T)$, where K represents the number of vehicles and T denotes the total number of rounds.

2) Q-VCG complexity: The Q-VCG involves computing similarity scores between pairs of vehicles, which has a constant complexity of $\mathcal{O}(1)$. Considering all unique pairs, the complexity is $\mathcal{O}(V(V-1)/2)$, which simplifies to $\mathcal{O}(V^2)$. The group formation process per round is approximated by $\mathcal{O}(V \cdot \mathcal{G})$, where $\mathcal{G}$ is the number of groups and V denotes the number of vehicles for similarity score computation.

3) Q-PCA complexity: The complexity of covariance matrix construction in Q-PCA is $\mathcal{O}(K \cdot Q^2)$. Next, the complexity of eigen decomposition can be calculated as $\mathcal{O}(Q^3)$. Assuming that the eigen decomposition is done once, the total complexity of Q-PCA can be given as $\mathcal{O}(K \cdot Q^2 + Q^3)$.

4) FL Complexity: In the learning process, the parameter update complexity per vehicle per round is $\mathcal{O}(P)$, where P is the number of model parameters in each vehicle's local model. The complexity of aggregating per round (assuming linear complexity in the number of vehicles) is $\mathcal{O}(K \cdot P)$. Thus total for all rounds can be given as $\mathcal{O}(R \cdot K \cdot P)$.

5) Communication overhead: The complexity of communication cost per vehicle per synchronization is $\mathcal{O}(1)$. Hence the complexity for all vehicles and all synchronizations can be given as $\mathcal{O}(\frac{K \cdot T}{S})$.

Therefore, combining these complexities the total computational complexity of the QV-FEDCOM framework can be given by $\mathcal{O}(K \cdot T) + \mathcal{O}(V^2) + \mathcal{O}(V \cdot \mathcal{G}) + \mathcal{O}(K \cdot Q^2 + Q^3) + \mathcal{O}(R \cdot K \cdot P) + \mathcal{O}(\frac{K \cdot T}{S})$.

X. Conclusion

In this study, we introduced the QV-FEDCOM algorithm, representing an integration of quantum computing and FL tailored for the vehicular metaverse. The framework, strengthened by the Q-STP, Q-VCG, and Q-PCA mechanisms, innovates a decentralized, efficient QFL approach, streamlining model training, optimizing communication, managing vehicle operations, intelligently addressing data heterogeneity, and optimizing memory consumption. Our QTL function, specifically designed for trajectory prediction tasks, combines the Huber loss with an angular deviation penalty to robustly handle errors and penalize large deviations in the predicted trajectory angle. Through comprehensive simulations, the QV-FEDCOM framework demonstrated its superiority by outperforming its various adaptations and benchmark QFL approaches in terms of accuracy and efficiency. Although our experiments were constrained by the limitations of classical simulations, the results provide insightful benchmarks and underscore the framework's potential within the vehicular metaverse ecosystem.

References

[1] B. Hazarika, K. Singh, S. Biswas, and C.-P. Li, "DRL-based resource allocation for computation offloading in IoV networks," *IEEE Trans. Industr. Inform.*, vol. 18, no. 11, pp. 8027–8038, Apr. 2022.

[2] W. Wang, M. H. Fida, Z. Lian, Z. Yin, Q.-V. Pham, T. R. Gadekallu, K. Dev, and C. Su, "Secure-enhanced federated learning for ai-empowered electric vehicle energy prediction," *IEEE Consum. Electron. Mag.*, Sep. 2021.

[3] B. Hazarika and K. Singh, "AFL-DMAAC: Integrated resource management and cooperative caching for URLLC-IoV networks," *IEEE Trans. Intell. Veh.*, Aug. 2023.

[4] B. Hazarika, K. Singh, A. Paul, and T. Q. Duong, "Hybrid machine learning approach for resource allocation of digital twin in UAV-aided internet-of-vehicles networks," *IEEE Trans. Intell. Veh.*, Nov. 2023.

[5] A. Hammoud, H. Otrok, A. Mourad, and Z. Dziong, "On demand fog federations for horizontal federated learning in IoV," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 3, pp. 3062–3075, May. 2022.

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

[7] X. Kong, H. Gao, G. Shen, G. Duan, and S. K. Das, "FedVCP: A federated-learning-based cooperative positioning scheme for social internet of vehicles," *IEEE Trans. Comput. Soc.*, vol. 9, no. 1, pp. 197–206, Mar. 2021.

[8] P. Singh, B. Hazarika, K. Singh, C. Pan, W.-J. Huang, and C.-P. Li, "Drl-based federated learning for efficient vehicular caching management," *IEEE Internet Things J.*, pp. 1–1, 2024.

[9] P. Singh, B. Hazarika, K. Singh, W.-J. Huang, and C.-P. Li, "Augmented multi-agent drl for multi-incentive task prioritization in vehicular crowdsensing," *IEEE Internet Things J.*, pp. 1–1, 2024.

[10] B. Hazarika, P. Saikia, K. Singh, and C.-P. Li, "Enhancing vehicular networks with hierarchical o-ran slicing and federated drl," *IEEE Trans. Green Commun. Netw.*, vol. 8, no. 3, pp. 1099–1117, 2024.

[11] Y. Ren, R. Xie, F. R. Yu, T. Huang, and Y. Liu, "Quantum collective learning and many-to-many matching game in the metaverse for connected and autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 71, no. 11, pp. 12 128–12 139, Jul. 2022.

[12] M. Chehimi, S. Y.-C. Chen, W. Saad, D. Towsley, and M. Debbah, "Foundations of quantum federated learning over classical and quantum networks," *IEEE Netw.*, pp. 1–1, Oct. 2023.

[13] J. Feng and J. Zhao, "Resource allocation for augmented reality empowered vehicular edge metaverse," *IEEE Trans. Commun.*, Sep. 2023.

[14] K. Singh, B. Hazarika, C.-P. Li, K. F. Tsang, and S. Biswas, "Digital twin-assisted resource allocation in UAV-aided internet of vehicles networks," in *Proc. IEEE ICC*, May. 2023, pp. 409–414.

[15] B. Hazarika, K. Singh, C.-P. Li, A. Schmeink, and K. F. Tsang, "RADiT: Resource allocation in digital twin-driven UAV-aided internet of vehicle networks," *IEEE J. Sel. Areas Commun.*, Aug. 2023.

[16] Y. Cao, Y. Zhao, J. Li, R. Lin, J. Zhang, and J. Chen, "Hybrid trusted/untrusted relay-based quantum key distribution over optical backbone networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 9, pp. 2701–2718, Mar. 2021.

[17] W. J. Yun, J. P. Kim, S. Jung, J. Park, M. Bennis, and J. Kim, "Slimmable quantum federated learning," *arXiv preprint arXiv:2207.10221*, Jul. 2022.

[18] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf, "Training deep quantum neural networks," *Nat. Commun.*, vol. 11, no. 1, p. 808, Feb. 2020.

[19] R. Huang, X. Tan, and Q. Xu, "Quantum federated learning with decentralized data," *IEEE J. Sel. Top. Quantum Electron.*, vol. 28, pp. 1–10, Apr. 2022.

[20] D. Chandra, A. S. Cacciapuoti, M. Caleffi, and L. Hanzo, "Direct quantum communications in the presence of realistic noisy entanglement," *IEEE Trans. Commun.*, vol. 70, no. 1, pp. 469–484, 2021.

[21] Z. Sun, L. Song, Q. Huang, L. Yin, G. Long, J. Lu, and L. Hanzo, "Toward practical quantum secure direct communication: A quantum-memory-free protocol and code design," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5778–5792, 2020.

[22] J. Cui, Y. Xiong, S. X. Ng, and L. Hanzo, "Quantum approximate optimization algorithm based maximum likelihood detection," *IEEE Trans. Commun.*, vol. 70, no. 8, pp. 5386–5400, 2022.

[23] A. S. Cacciapuoti, M. Caleffi, R. Van Meter, and L. Hanzo, "When entanglement meets classical communications: Quantum teleportation for the quantum internet," *IEEE Trans. Commun.*, vol. 68, no. 6, pp. 3808–3833, 2020.

[24] M. Chehimi and W. Saad, "Physics-informed quantum communication networks: A vision toward the quantum internet," *IEEE Netw.*, vol. 36, no. 5, pp. 32–38, Nov. 2022.

[25] H. Zhao, "Non-IID quantum federated learning with one-shot communication complexity," *Quantum Mach.*, vol. 5, no. 1, p. 3, 2023.

[26] B. Narottama and S. Y. Shin, "Federated quantum neural network with quantum teleportation for resource optimization in future wireless communication," *IEEE Trans. Veh. Technol.*, May. 2023.

[27] Z. Qu, Y. Li, B. Liu, D. Gupta, and P. Tiwari, "DTQFL: A digital twin-assisted quantum federated learning algorithm for intelligent diagnosis in 5g mobile network," *IEEE J. Biomed. Health Inform.*, Aug. 2023.

[28] Y. Zhang, C. Zhang, C. Zhang, L. Fan, B. Zeng, and Q. Yang, "Federated learning with quantum secure aggregation," *arXiv preprint arXiv:2207.07444*, Jul. 2022.

[29] S. Zeng, Z. Li, H. Yu, Z. Zhang, L. Luo, B. Li, and D. Niyato, "HFedMS: Heterogeneous federated learning with memorable data semantics in industrial metaverse," *IEEE Trans. Cloud Comput.*, Mar. 2023.

[30] D. Gurung, S. R. Pokhrel, and G. Li, "Decentralized quantum federated learning for metaverse: Analysis, design and implementation," *arXiv preprint arXiv:2306.11297*, Jun. 2023.

[31] Y.-A. Chen, Q. Zhang, T.-Y. Chen *et al.*, "An integrated space-to-ground quantum communication network over 4,600 kilometres," *Nature*, vol. 589, no. 7841, pp. 214–219, Jan. 2021.

[32] I. Ahmad, R. M. Noor, I. Ahmedy, S. A. A. Shah, I. Yaqoob, E. Ahmed, and M. Imran, "Vanet–LTE based heterogeneous vehicular clustering for driving assistance and route planning applications," *Comput Netw*, vol. 145, pp. 128–140, 2018.

[33] T. Zhang, D.-g. Zhang, H.-r. Yan, J.-n. Qiu, and J.-x. Gao, "A new method of data missing estimation with fnn-based tensor heterogeneous ensemble learning for internet of vehicle," *Neurocomputing*, vol. 420, pp. 98–110, 2021.

[34] Y. Li, Y. Zhou, A. Jolfaei, D. Yu, G. Xu, and X. Zheng, "Privacy-preserving federated learning framework based on chained secure multiparty computing," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6178–6186, 2021.

[35] A. Großwendt, H. Röglin, and M. Schmidt, "Analysis of ward's method," in *Proc. ACM-SIAM*. SIAM, 2019, pp. 2939–2957.

[36] Y. Xiong, S. X. Ng, and L. Hanzo, "Quantum error mitigation relying on permutation filtering," *IEEE Trans. Commun.*, vol. 70, no. 3, pp. 1927–1942, 2021.

[37] M. Greenacre, P. J. Groenen, T. Hastie, A. I. d'Enza, A. Markos, and E. Tuzhilina, "Principal component analysis," *Nat. Rev. Methods Primers*, vol. 2, no. 1, p. 100, Dec. 2022.

[38] Y. Wang and Y. Luo, "Resource-efficient quantum principal component analysis," *Quantum Sci. and Technol.*, vol. 9, no. 3, p. 035031, 2024.

[39] K. Sudharson and B. Alekhya, "A comparative analysis of quantum-based approaches for scalable and efficient data mining in cloud environments." *Quantum Inf. Comput.*, vol. 23, no. 9&10, pp. 783–813, 2023.

[40] E. Dri, A. Aita, T. Fioravanti, G. Franco, E. Giusto, G. Ranieri, D. Corbelletto, and B. Montrucchio, "Towards an end-to-end approach for quantum principal component analysis," in *Proc. IEEE QCE*, vol. 2, 2023, pp. 1–6.

[41] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nat. Phys.*, vol. 10, no. 9, pp. 631–633, Sep. 2014.

[42] C. He, J. Li, W. Liu, J. Peng, and Z. J. Wang, "A low-complexity quantum principal component analysis algorithm," *IEEE Trans. Quantum Eng.*, vol. 3, pp. 1–13, Jan. 2022.

[43] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, "General parameter-shift rules for quantum gradients," *Quantum*, vol. 6, p. 677, Mar. 2022.

[44] J. Xie, S. Liu, J. Chen, and J. Jia, "Huber loss based distributed robust learning algorithm for random vector functional-link network," *Artif. Intell. Rev.*, vol. 56, no. 8, pp. 8197–8218, Aug. 2023.

[45] Y. de Viragh, M. Bjelonic, C. D. Bellicoso, F. Jenelten, and M. Hutter, "Trajectory optimization for wheeled-legged quadrupedal robots using linearized zmp constraints," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1633–1640, 2019.

[46] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Proc. PMLR CoRL*, 2017, pp. 1–16.

[47] H. M. Fahmy, M. A. A. E. Ghany, and G. Baumann, "Vehicle risk assessment and control for lane-keeping and collision avoidance at low-speed and high-speed scenarios," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 4806–4818, 2018.

[48] N. Wang, G. Guo, B. Wang, and C. Wang, "Traffic clustering algorithm of urban data brain based on a hybrid-augmented architecture of quantum annealing and brain-inspired cognitive computing," *Tsinghua Sci Technol*, vol. 25, no. 6, pp. 813–825, 2020.

[49] C. Wang, T. Ji, and S. Wang, "Online taxi dispatching algorithm based on quantum annealing," in *Proc. Springer CAAI*, 2022, pp. 337–347.

[50] W. Pian, Y. Wu, X. Qu, J. Cai, and Z. Kou, "Spatial-temporal dynamic graph attention networks for ride-hailing demand prediction," *arXiv preprint arXiv:2006.05905*, 2020.

[51] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[52] J. R. Johansson, P. D. Nation, and F. Nori, "QuTiP: An open-source Python framework for the dynamics of open quantum systems," *Comput. Phys. Commun.*, vol. 183, no. 8, pp. 1760–1772, Aug. 2012.

[53] B. Hazarika, K. Singh, T. Q. Duong, and O. A. Dobre, "Quantum-driven context-aware federated learning in heterogeneous vehicular metaverse ecosystem," in *Proc. IEEE ICC 2024*, 2024, pp. 1533–1538.