GenAI-Enhanced Federated Multi-Agent DRL for Digital Twin-Assisted IoV Networks

Piyush Singh, *Graduate Student Member, IEEE*, Bishmita Hazarika, *Member, IEEE*, Keshav Singh, *Member, IEEE*, Wan-Jen Huang, *Member, IEEE*, and Trung Q. Duong, *Fellow, IEEE*

***Abstract*—Achieving real-time decision-making and efficient resource management in dynamic, large-scale Internet-of-Vehicles (IoV) networks is a significant challenge due to their inherent complexity and scale. To address this, we propose a digital twin (DT)-assisted IoV framework that integrates a novel semi-synchronous adaptive federated learning (AdFL) approach with multi-agent deep reinforcement learning, enhanced by generative artificial intelligence (GenAI) techniques, specifically conditional variational autoencoders (CVAE). The framework optimizes partial task offloading across distributed mobile edge computing (MEC) servers, ensuring scalable, efficient, and accurate decision-making in heterogeneous vehicular networks. By continuously mirroring the real-time states of vehicles and roadside units (RSUs), the DT framework enables precise resource allocation and adaptive task management. To tackle the complexities of dynamic environments, we design a global model with transformer layers embedded in the federated learning (FL) process, capturing long-range dependencies. A novel semi-synchronous aggregation mechanism is introduced to balance timely updates with model quality. The proposed adaptive federated multi-agent reinforcement learning (AF-MARL) algorithm facilitates decentralized, collaborative learning among vehicles and RSUs, optimizing overall cost, and energy efficiency, reducing delay, and improving task completion rates. Extensive simulations demonstrate the effectiveness of the proposed framework against other existing approaches, highlighting its potential to transform real-time decision-making in IoV networks.**

***Index Terms*—Internet-of-Vehicles (IoV), digital twin, federated learning (FL), multi-agent reinforcement learning (MARL), mobile edge computing.**

I. Introduction

THE rapid growth of Internet-of-Vehicles (IoV) networks, driven by the proliferation of autonomous vehicles and the increasing demand for intelligent transportation systems (ITS), has exposed significant challenges in traditional centralized computing models. Centralized cloud architectures often struggle with latency, scalability, and reliability, especially in the dynamic and data-intensive environments of IoV networks. These limitations underscore the need for decentralized edge intelligence, where decisions are made closer to the data source to enhance responsiveness and reduce latency. Mobile edge computing (MEC) has emerged as a promising solution by bringing computational resources closer to vehicles, and reducing dependence on distant cloud servers. However, MEC still faces limitations due to its reliance on centralized coordination, which can create bottlenecks and single points of failure. Recent studies have explored integrating MEC with IoV networks to improve real-time decision-making. For instance, [1] proposes a security solution for MEC in vehicular ad hoc networks, focusing on safeguarding privacy and computational overhead in large-scale deployments through a broadcast proxy re-encryption scheme and decentralized trust management. Similarly, [2] introduces a task offloading algorithm for dynamic vehicular networks, leveraging multivariate long short-term memory (LSTM) for workload prediction and distributed deep reinforcement learning (DRL) to optimize resource allocation in MEC environments. In another approach, [3] presents an intelligent task offloading strategy for IoV networks using reconfigurable intelligent surfaces (RISs) and MEC to enhance resource allocation and communication quality, demonstrating the versatility of MEC in different vehicular network contexts.

As vehicular networks evolve, the integration of digital twin (DT) technology has become crucial for real-time monitoring and predictive analytics. A DT is a virtual replica of a physical entity that continuously updates with real-time data, mirroring the state of vehicles and roadside units (RSUs) in IoV networks [4], [5]. This real-time synchronization significantly enhances decision-making at the network edge, where decentralized edge intelligence is essential to handle the high volume of data and ensure timely processing. Studies such as [6] and [7] have demonstrated the effectiveness of DTs in improving resource allocation in IoV networks by providing accurate, real-time information on vehicle states and network conditions. Additionally, intelligent task offloading frameworks leveraging DTs, as proposed in [8] and [9], highlight how DTs can optimize computational and communication resources by predicting vehicular behavior and network load. DTs also contribute to various other aspects of IoV networks, including traffic management [10], security and

The work of K. Singh and W. J. Huang was supported by the National Science and Technology Council of Taiwan, under Grants NSTC 112-2221-E-110-038-MY3, NSTC 113-2218-E-110-009, and NSTC 113-2221-E-110-059-MY3. The work of T. Q. Duong was supported by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109. (*Corresponding author: K. Singh.*)

P. Singh is with the Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan. (Email: piyush230497@gmail.com).

B. Hazarika is with the Faculty of Engineering and Applied Science, Memorial University, St. John's, Canada. (Email: bhazarika@mun.ca).

K. Singh, and W. J. Huang are with the Institute of Communications Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan. (Email: keshav.singh@mail.nsysu.edu.tw, wanjen.huang@g-mail.nsysu.edu.tw).

T. Q. Duong is with the Faculty of Engineering and Applied Science, Memorial University, 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. (Email: tduong@mun.ca).

This paper has been accepted in part at the IEEE Wireless Communications and Networking Conference (WCNC), 24–27 March 2025, Milan, Italy.

privacy [11], and navigation [12], underscoring their potential to transform IoV systems by enhancing operational efficiency and safety.

In response to the limitations of centralized architectures, decentralized machine learning (ML) approaches have gained traction in vehicular networks. Among these, federated learning (FL) has emerged as a promising paradigm that enables vehicles to collaboratively learn a shared model without exchanging raw data, thereby preserving privacy and reducing communication overhead [13]. Traditional FL methods, such as horizontal FL [14] and vertical FL [15], have been adapted for vehicular networks, with studies like [16] and [17] exploring their application in autonomous driving and other vehicular scenarios. Horizontal FL, where vehicles with similar data distributions participate in learning, has shown promise in scenarios requiring collaboration among homogeneous devices. On the other hand, vertical FL, which deals with different features distributed across vehicles, remains less common but presents unique opportunities for scenarios where different types of data are prevalent across the network [18]. However, these FL approaches often face challenges in dynamic vehicular environments, particularly concerning synchronization and model aggregation. Synchronous FL, which requires all participants to complete their updates before aggregation, can lead to significant delays, especially in networks with varying communication capabilities [19]. Asynchronous FL offers some relief but can result in stale updates and convergence issues in large-scale networks [20]. Although asynchronous approaches tailored for vehicular networks, such as those proposed in [21], [22], and [23], have demonstrated improved scalability, fully asynchronous learning remains unstable and unsuitable for realistic large-scale IoV frameworks due to issues like redundant model exchanges and increased operational costs. This has led to the exploration of semi-synchronous approaches, as seen in studies like [24] and [25], which balance timeliness and update quality by combining the benefits of synchronous and asynchronous FL. Table I shows the comparison between different FL approaches including the FL method we propose in this study in the subsequent sections.

Beyond traditional ML, the integration of DRL in a federated setting has shown promise for optimizing decision-making in complex vehicular networks [26], [27]. Multi-agent DRL (MADRL) allows vehicles to learn collaborative strategies for tasks such as resource allocation and task offloading [28]. When combined with FL, known as federated DRL, leverages the strengths of both paradigms—FL's privacy-preserving collaboration and DRL's adaptive decision-making. Studies like [29], [30], and [31] have explored federated DRL for task offloading in vehicular networks, demonstrating the potential for improved scalability and decision accuracy. However, in real-life IoV settings, the deployment of these approaches faces challenges in environments with heterogeneous agents and dynamic conditions, where the assumptions of homogeneity and stability do not hold. Moreover, the convergence of DRL in a federated setting can be slow, particularly in networks with high variability in agent behavior.

Despite the progress made with MEC, DT, FL and DRL, significant research gaps remain, particularly concerning their integration and application in large-scale, dynamic IoV networks. Traditional FL and DRL approaches often assume homogeneity and stability, which do not reflect the real-world conditions of IoV environments. The inherent variability in vehicle behavior, communication quality, and network topology can lead to inefficiencies and reduced accuracy in decision-making. Furthermore, while fully asynchronous FL methods offer scalability, they often result in redundant model exchanges and increased operational costs, making them unsuitable for large-scale deployments. Semi-synchronous approaches, although promising, require further exploration to optimize the trade-offs between update timeliness and model accuracy.

The introduction of generative artificial intelligence (GenAI) techniques presents an opportunity to address these challenges [32]. Generative models, such as conditional variational autoencoders (CVAE) [33], [34] and generative adversarial networks (GANs) [35], have shown promise in enhancing the adaptability and robustness of ML models in dynamic environments. In the context of IoV networks, these techniques can be leveraged to improve the accuracy and efficiency of FL and DRL frameworks by capturing the complexities of real-time vehicular data and enabling more responsive and adaptive decision-making processes. Although this is a relatively new area with limited research focused specifically on vehicular networks, studies such as [35]–[37] demonstrate the potential of GenAI-enhanced FL frameworks in various other wireless IoT network frameworks. For instance, these studies propose architectures that incorporate GenAI for distributed traffic flow prediction, tackle challenges related to non-IID data using GANs, and implement distributed model training that preserves privacy, reduces communication costs, and decreases training latency.

Thus, to address this gap, we propose a DT-assisted IoV framework that combines semi-synchronous FL with multi-agent DRL, enhanced by GenAI, particularly CVAE, for efficient partial task offloading. This framework is specifically designed to handle the complexities of large-scale, dynamic vehicular networks, overcoming the limitations of existing approaches and enhancing the scalability, efficiency, and accuracy of decision-making processes in IoV environments. To highlight our approach's distinct advantages and novelties Table II systematically compares the key features of our model against existing studies. The table outlines various critical aspects such as the type of FL, handling of heterogeneous data, incorporation of digital twins, use of genAI, support for real-time decision-making, resource optimization, and scalability in dynamic IoV networks. Our primary contributions are as follows:

- We develop a DT-assisted framework that enhances real-time decision-making and resource management in dynamic vehicular networks by continuously mirroring the real-time states of vehicles and RSUs, enabling accurate information flow and supporting optimal task offloading and adaptive resource allocation strategies.
- We introduce a GenAI-enhanced adaptive federated

TABLE I: Comparison of the Proposed Framework with Federated Learning Types

FL Type	Handles Non-IID Data	Adapts to Agent Variability	Supports Dynamic Environments	Key Characteristics
Proposed Framework	**Yes** (via CVAE + gradient norm evaluation)	**Yes** (semi-synchronous aggregation)	**Yes** (via real-time DT updates)	Combines Digital Twins, Generative AI, and Multi-Agent RL for IoV task offloading and resource management
Synchronous FL	**Partial** (requires IID or near-IID data for stability)	**Limited** (requires equal computational capabilities)	**No** (fixed aggregation intervals)	Aggregates updates only after all agents have completed training, leading to synchronization delays in dynamic IoV settings
Asynchronous FL	**Yes** (adapts to non-IID data)	**Yes** (agents can update independently)	**Moderate** (faster updates but less coordinated)	Allows updates at different times; suitable for heterogeneous agents but may face convergence issues in dynamic environments
Semi-Synchronous FL	**Yes** (balances IID and non-IID settings)	**Yes** (dynamic aggregation thresholds)	**Yes** (suitable for dynamic networks)	Combines benefits of synchronous and asynchronous FL, balancing timeliness and model quality; used in the proposed AdFL
Horizontal FL	**Partial** (requires IID data)	**Limited** (same feature space across agents)	**No** (limited to specific tasks)	Suitable for agents with similar feature distributions (e.g., IoV tasks requiring identical sensors across vehicles)
Vertical FL	**Yes** (handles heterogeneous features)	**Limited** (requires feature alignment)	**No** (applies to static datasets)	Designed for scenarios where different agents hold complementary features; not ideal for IoV networks
Personalized FL (pFL)	**Yes** (customized local models)	No (focuses on personalization, not collaboration)	**No** (static environment assumptions)	Trains separate models for each client to address non-IID data but lacks collaboration or dynamic adaptability
Clustered FL	**Yes** (groups agents with similar data)	**No** (fixed clusters)	**Low** (suffers from clustering overhead)	Groups agents based on data similarity; unsuitable for dynamic IoV scenarios due to clustering delays and inefficiency in highly mobile networks

TABLE II: Comparison of Proposed Framework with Related Works

Feature/Aspect	**Proposed**	**[4]**	**[5]**	**[14]**	**[15]**	**[16]**	**[17]**	**[19]**	**[20]**
FL Type	Semi-Sync	N/A	N/A	Horizontal	Async, Vertical	Horizontal	Horizontal	Sync	Async
Heterogeneous Data	✓	✓	✓	✓	✓	✓	✓	✓	✓
Digital Twin	✓	✓	✓	×	×	×	×	×	×
Generative AI	✓	×	×	×	×	×	×	×	×
Real-Time Decision	✓	✓	✓	×	×	✓	✓	✓	✓
Resource Optimization	✓	✓	✓	✓	✓	✓	✓	✓	✓
Adaptive Federated Learning	✓	×	×	✓	×	×	×	✓	×
Context-Aware Representations	✓	✓	✓	×	×	✓	×	✓	✓
Scalability in Dynamic IoV	✓	✓	✓	×	×	✓	×	✓	✓
Energy-Efficient	✓	✓	✓	✓	✓	✓	✓	✓	✓
Traffic Condition Adaptation	✓	✓	✓	×	×	✓	×	×	×

learning (AdFL) algorithm that leverages CVAE to generate context-aware latent representations, allowing the system to effectively capture the complexities of dynamic vehicular environments and improve the adaptability and scalability of the FL process.
- We design a global model with transformer layers to capture long-range dependencies in the dynamic IoV environment efficiently. We also introduce a novel semi-synchronous aggregation mechanism within the AdFL framework to balance timely updates with global model quality.
- We propose an adaptive federated multi-agent reinforcement learning (AF-MARL) algorithm that optimizes task offloading by enabling vehicles and RSUs to collaboratively learn optimal strategies in a distributed manner, thereby enhancing system performance in terms of energy efficiency, delay reduction, and task completion rates.
- Finally, we evaluate the framework's performance through extensive simulations, demonstrating the superiority of the proposed method for the IoV framework.

Structure of the paper: The flow of the paper is organized as follows: Section II describes the system model; Section III introduces the task offloading model; Section IV focuses on the problem formulation and solution approach; Section V

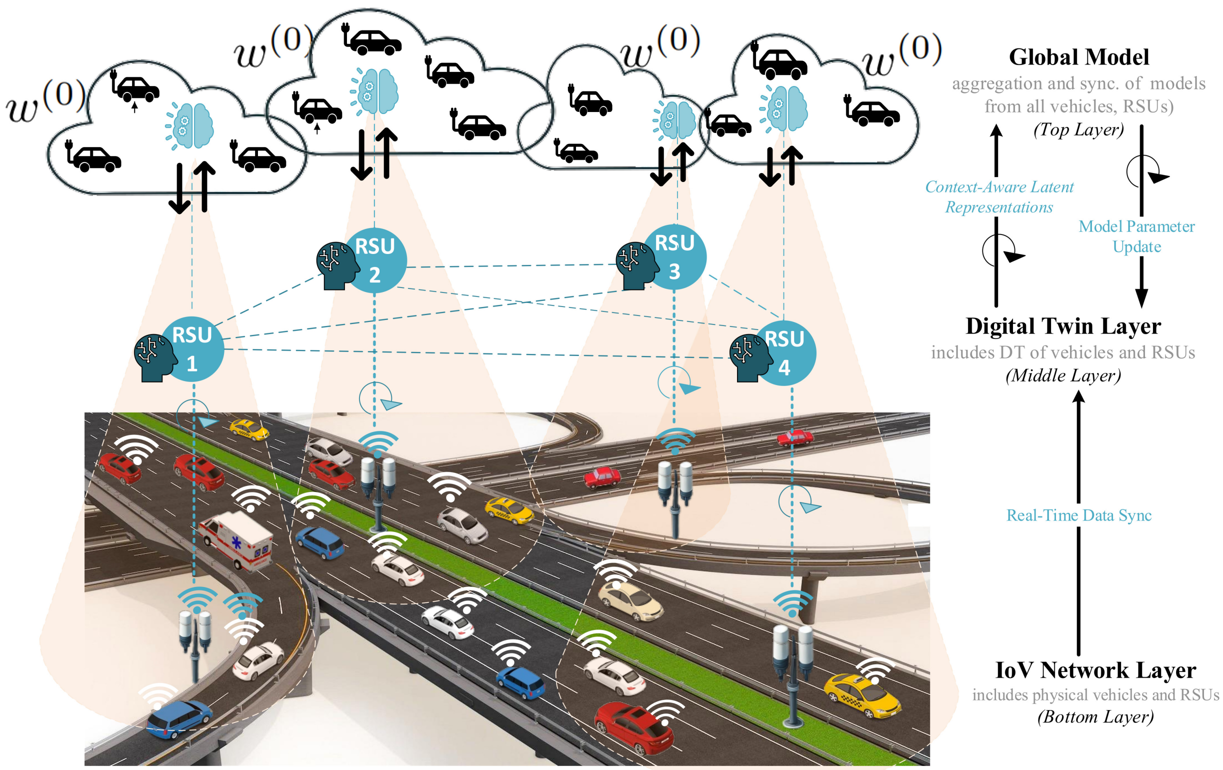


Fig. 1: Illustration of the considered dynamic DT-assisted IoV network with distributed edge intelligence.

explores AdFL for dynamic vehicular networks; Section VI explains the AF-MARL framework; Section VII provides the performance evaluation; Section VIII presents the result analysis. Finally, Section IX concludes the article.

II. System Model

We consider a dynamic urban IoV network designed to support real-time vehicular communication and computation, leveraging a distributed edge intelligence framework as shown in Fig. 1. This model ensures the system's adaptability to the highly dynamic and heterogeneous nature of vehicular networks, where low latency and energy efficiency are paramount. The urban area is modeled as a two-dimensional plane of size $X \times X$ square kilometers, representing a city with dense vehicular traffic and infrastructural support with MEC servers. The network comprises N vehicles $V = V_1, V_2, \ldots, V_N$, which are dynamically distributed throughout the area, and M RSUs, $R = R_1, R_2, \ldots, R_M$, each equipped with MEC capabilities. The RSUs are strategically positioned to ensure full coverage and connectivity, forming a distributed computing network that eliminates the need for centralized cloud services.

The movement of vehicles is modeled as a continuous-time stochastic process, reflecting realistic urban traffic dynamics. Each vehicle V_n, where $n \in 1, 2, \ldots, N$, is characterized by a position vector $\mathbf{p}_n(t)$ at time t, evolving based on its velocity vector $\mathbf{v}_n(t)$. To simulate the variability in traffic conditions, we model the velocity as:

$$\mathbf{v}_n(t) = \mathbf{v}_n^{\text{avg}}(t) + \mathbf{v}_n^{\text{rand}}(t)\,, \tag{1}$$

where $\mathbf{v}_n^{\text{avg}}(t)$ represents the average velocity based on current road conditions, and $\mathbf{v}_n^{\text{rand}}(t)$ is a stochastic component capturing random fluctuations due to factors such as traffic congestion or sudden stops. The position update for vehicle V_n is given by:

$$\mathbf{p}_n(t + \Delta t) = \mathbf{p}_n(t) + \mathbf{v}_n(t) \cdot \Delta t\,, \tag{2}$$

where Δt is a small time increment. This continuous change in position affects the connectivity between vehicles and RSUs, influencing task offloading decisions.

A. Distributed Edge Computing Architecture

The IoV network operates on a distributed edge computing architecture. Each RSU with MEC serves as a local computational hub, interfacing directly with vehicles within its communication range. The distributed nature of the edge computing framework ensures that computation is performed closer to the data source, minimizing latency. Each vehicle V_n is equipped with computational resources (e.g., central processing unit (CPU) cycles per second f_n), communication capabilities (e.g., data rate r_n), and energy capacity E_n. Similarly, each RSU R_m has its computational power f_m, communication capability r_m, and energy availability E_m.

The RSUs are connected in a mesh topology, enabling direct communication between neighboring RSUs. This topology enhances system robustness and reduces latency by allowing tasks to be rerouted through multiple paths if network congestion or failures occur. For instance, if an RSU experiences high traffic, the system can dynamically shift task offloading to neighboring RSUs, thereby balancing the load and maintaining low latency.

B. Digital Twin Framework

Digital Twins (DTs) play a central role in the proposed framework by providing real-time and historical data for vehicles and RSUs. These DTs act as virtual representations of physical IoV entities, enabling dynamic synchronization and predictive modeling. This section explains the construction, synchronization, and role of DTs in supporting the framework. To effectively manage the dynamic and heterogeneous nature of the IoV network, we implement a DT framework that enhances real-time decision-making and resource management. Each physical entity within the network, whether a vehicle or an RSU, is associated with a corresponding DT. The DT serves as a virtual replica of the physical entity, continuously mirroring its real-time state and operational parameters [4].

The DT of a vehicle V_n at time t, denoted as $\mathrm{DT}_n(t)$, encapsulates a comprehensive set of attributes that define the vehicle's current status. Specifically, this includes:

$$\mathrm{DT}_n(t) = \{f_n, r_n, E_n, \mathbf{v}_n(t), D_n, C_n, T_n\}\,, \tag{3}$$

where the DT of each vehicle is characterized by several key parameters, including the computational power f_n (measured in CPU cycles per second), communication capability r_n (in bits per second), energy capacity E_n (in joules), and the velocity vector $\mathbf{v}_n(t)$ at time t. Additionally, it includes the data size of the current task D_n (in bits), the required CPU cycles C_n for completing the task, and the maximum tolerable delay T_n (in seconds) for task processing.

Similarly, the DT of an RSU R_m, denoted as $\mathrm{DT}_m(t)$, maintains a real-time representation of the RSU's operational parameters:

$$\mathrm{DT}_m(t) = \{f_m, r_m, E_m\}\,. \tag{4}$$

The DTs continuously synchronize with their physical counterparts, ensuring that the digital representations accurately reflect the real-time state of the network. The synchronization process occurs at regular intervals, determined by the system's requirements for accuracy and responsiveness. While frequent synchronization ensures up-to-date information, it also introduces additional energy consumption and delay, denoted as E_n^{DT} and t_n^{DT}, respectively. These contributions must be accounted for in the overall energy and delay models, particularly in high-task offloading scenarios where the DT's decision-making process is critical.

In this study, the DTs aggregate real-time data and perform predictive analysis to optimize offloading decisions, resource allocation, and energy management strategies. Although the actual computation and task execution occur in the physical layer (i.e., within the vehicles and RSUs), the DTs provide the necessary intelligence to guide these processes. The DTs also facilitate the dynamic adaptation of the network to changing conditions, ensuring that the system remains efficient and responsive even under varying loads. The data generated by DTs serves as the foundation for creating context-aware latent representations using GenAI, as described in the subsequent sections.

III. Task Offloading Model

In this section, we present the task offloading model for vehicles in the IoV network. The model incorporates the dynamic interplay between local computing and edge offloading, enhanced by the DT framework [5].

A. Local Computing Model

When a vehicle V_n processes a task locally, the computation delay t_n^{loc} for a task requiring C_n CPU cycles is given by:

$$t_n^{\mathrm{loc}} = \frac{C_n}{f_n}\,, \tag{5}$$

where f_n is the computational power of the vehicle. The energy consumption for local task execution, E_n^{loc}, is modeled as:

$$E_n^{\mathrm{loc}} = \kappa_n \cdot C_n\,, \tag{6}$$

where κ_n represents the energy consumption coefficient, accounting for both dynamic and static power consumption [8].

B. Edge Computing Model

In the edge computing model, a vehicle V_n may offload a fraction ρ_n of its task to an RSU or another vehicle while processing the remaining portion $(1-\rho_n)$ locally. The total delay t_n^{off} includes the upload delay t_n^{up}, computation delay t_n^{comp}, download delay t_n^{down}, and the DT-induced delay t_n^{DT}. The corresponding energy consumption E_n^{off} accounts for transmission energy, computation energy, and the DT overhead.

1) Upload Delay and Energy: The upload delay t_n^{up} for transmitting data to the RSU or another vehicle is:

$$t_n^{\mathrm{up}} = \frac{\rho_n \cdot D_n}{r_n}\,, \tag{7}$$

where D_n is the data size of the task, and r_n is the data transmission rate:

$$r_n = W_n \log_2\left(1 + \frac{P_n d_n^{-\beta} h_n^2}{N_0 + I_n}\right)\,. \tag{8}$$

Here, W_n is the bandwidth of the wireless channel, P_n is the transmission power, d_n represents the distance to the RSU or another vehicle, β is the path loss exponent, h_n is the channel gain, N_0 is the noise power, and I_n represents interference.

The corresponding energy consumption for data transmission is:

$$E_n^{\mathrm{tx}} = P_n \cdot t_n^{\mathrm{up}}\,. \tag{9}$$

2) Computation Delay and Energy: The computation delay, t_n^{comp}, depends on whether the task is offloaded to another vehicle or an RSU:

$$t_n^{\text{comp}} = \begin{cases} \frac{\rho_n C_n}{f'_n} & \text{if offloaded to another vehicle}\,, \\ \frac{\rho_n C_n}{f'_m} & \text{if offloaded to an RSU}\,, \end{cases} \tag{10}$$

where f'_n and f'_m are the computational powers of the receiving vehicle and RSU, respectively. The energy consumption for computation during offloading is:

$$E_n^{\text{comp}} = \rho_n \cdot \kappa'_m \cdot C_n + E_n^{\text{DT}}\,, \tag{11}$$

where κ'_m is the energy coefficient for the computational resource used, and E_n^{DT} is the additional energy consumed by the DT framework.

3) Download Delay and Energy: The download delay t_n^{down} for receiving the processed task results is:

$$t_n^{\text{down}} = \frac{\delta D_n}{r_n}\,. \tag{12}$$

where δ is the ratio of output to input data size, typically $\delta \ll 1$, making this delay often negligible. The energy consumed during the download is:

$$E_n^{\text{down}} = P_n \cdot t_n^{\text{down}}\,. \tag{13}$$

4) Total Offloading Delay and Energy: The total delay and energy consumption for offloading are given, respectively, as:

$$t_n^{\text{off}} = \max\left(t_n^{\text{up}} + t_n^{\text{comp}} + t_n^{\text{down}} + t_n^{\text{DT}}, t_n^{\text{loc}}\right)\,, \tag{14}$$

$$E_n^{\text{off}} = E_n^{\text{tx}} + E_n^{\text{comp}} + E_n^{\text{down}} + E_n^{\text{DT}}\,. \tag{15}$$

C. Overall Energy and Delay Considerations

The overall energy consumption E_n^{total} and delay t_n^{total} for a vehicle V_n are determined by comparing the local and offloading options:

$$E_n^{\text{total}} = \min\left(E_n^{\text{loc}}, E_n^{\text{off}}\right)\,, \tag{16}$$

$$t_n^{\text{total}} = \min\left(t_n^{\text{loc}}, t_n^{\text{off}}\right)\,. \tag{17}$$

D. Offloading Under Resource Constraints

In scenarios where a vehicle V_n lacks sufficient resources to process a task locally, the DT framework mandates task offloading. The DT continuously monitors the vehicle's computational power $f_n(t)$ and remaining energy $E_n(t)$ to determine the necessity for offloading:

$$f_n(t) < C_n \quad \text{or} \quad E_n(t) < E_n^{\text{loc}}(t)\,. \tag{18}$$

If offloading is required, the system recalculates the total energy consumption, E_n^{off}, and delay, t_n^{off}:

$$E_n^{\text{off}} = E_n^{\text{tx}} + E_m^{\text{comp}} + E_n^{\text{rx}}\,, \tag{19}$$

$$t_n^{\text{off}} = t_n^{\text{tx}} + t_m^{\text{comp}} + t_n^{\text{rx}}\,. \tag{20}$$

If the nearest RSU or vehicle cannot meet the delay constraints, the DT framework implements fallback mechanisms, such as task splitting or partial local processing, and may initiate an emergency broadcast to nearby nodes for immediate assistance.

When vehicle V_n decides to offload a task, it must consider the service cost $\mathcal{P}_n$ associated with the offloading. This cost is proportional to the fraction ρ_n of the task that is offloaded and is calculated as:

$$\mathcal{P}_n = \rho_n \cdot p_n\,, \tag{21}$$

where p_n is the price per unit of offloaded computation.

IV. Problem Formulation & Solution Approach

In this section, we formulate the optimization problem for intelligent task offloading in a distributed edge intelligence-enabled IoV network. The objective is to minimize the overall cost Ω, which balances delay, energy consumption, and service cost while ensuring timely task completion and efficient resource utilization. The total cost Ω integrates three key factors: offloading delay t_n^{off}, energy consumption E_n^{total}, and the offloading service cost $\mathcal{P}_n$. To enable a meaningful aggregation of these metrics into a single objective function, each factor is normalized by its respective maximum permissible value. The normalized offloading delay is given by $\frac{t_n^{\text{off}}}{T_n}$, where T_n is the maximum tolerable delay. Similarly, energy consumption is normalized as $\frac{E_n^{\text{total}}}{E_{\max}}$, where $E_{\max}$ is the maximum allowable energy consumption, and service cost is normalized as $\frac{\mathcal{P}_n}{P_{\max}}$, where $P_{\max}$ represents the maximum allowable service cost. The resulting objective function is expressed as:

$$\Omega = \sum_{n=1}^{N}\left(\frac{t_n^{\text{off}}}{T_n} + \frac{E_n^{\text{total}}}{E_{\max}} + \frac{\mathcal{P}_n}{P_{\max}}\right)\,. \tag{22}$$

Normalization ensures that each term contributes proportionally to the total cost, irrespective of its original scale or unit. This prevents any single metric, such as delay or energy consumption, from dominating the optimization and ensures a balanced trade-off among the factors.

The optimization problem is mathematically formulated as:

$$\begin{aligned} (\mathcal{P}1): \quad & \min_{\{\rho_n\}} \quad \Omega = \sum_{n=1}^{N}\left(\frac{t_n^{\text{off}}}{T_n} + \frac{E_n^{\text{total}}}{E_{\max}} + \frac{\mathcal{P}_n}{P_{\max}}\right) \\ \text{s.t.} \quad & (C.1) \quad t_n^{\text{off}} \le T_n \quad \forall n, \\ & (C.2) \quad \text{If } E_n(t) < E_n^{\text{loc}}, \text{ offload mandatory}, \forall n, \\ & (C.3) \quad 0 \le \rho_n \le 1, \quad \mathcal{P}_n = \rho_n \cdot p_n \quad \forall n, \\ & (C.4) \quad 0 \le f'_n, f'_m \le f_n, f_m \quad \forall n, \forall m, \\ & (C.5) \quad \sum_{i=1}^{N} \rho_i \cdot f'_n \le f_n \quad \forall n, \\ & (C.6) \quad \sum_{i=1}^{N} \rho_i \cdot f'_m \le f_m \quad \forall m, \\ & (C.7) \quad t_n^{\text{off}} + t_n^{\text{loc}} \le T_n \quad \forall n. \end{aligned} \tag{23}$$

In this formulation, $(C.1)$ ensures that the offloading delay t_n^{off} does not exceed the maximum tolerable delay T_n for each vehicle. Constraint $(C.2)$ governs energy consumption, mandating offloading if the energy required for local processing exceeds the available energy $E_n(t)$, with the total energy consumption E_n^{total} being the minimum of the local and offloading energy costs. Constraint $(C.3)$ enforces that the offloading fraction ρ_n is within the range [0, 1], with the offloading cost $\mathcal{P}_n$ directly proportional to ρ_n. Constraints $(C.4)$ through $(C.6)$ ensure that the computational resources allocated for offloading do not exceed the computational capacities of the receiving vehicle or RSU, and that the aggregate resources allocated by any vehicle V_n or RSU R_m to all offloaded tasks remain within their respective total capacities. Finally, $(C.7)$ ensures that the combined delay from offloading and local processing does not surpass T_n, thereby guaranteeing the timely completion of tasks.

Normalization also ensures fairness in balancing trade-offs between delay, energy consumption, and service cost. For example, if t_n^{off} approaches T_n, its contribution to the total cost increases, incentivizing strategies that reduce delay. Similarly, if energy consumption E_n^{total} or service costs $\mathcal{P}_n$ are close to their respective maximums, the optimization framework prioritizes minimizing these metrics to achieve a balanced solution. This design enables the framework to adapt to varying constraints and optimize task offloading under dynamic IoV network conditions.

A. Solution Approach

The problem formulated above is NP-hard due to its combinatorial nature and the complex interdependencies between task offloading decisions, energy consumption, and communication delays in a dynamic vehicular environment. Traditional optimization methods are insufficient for real-time, large-scale IoV networks.

To address the challenges of optimizing task offloading in dynamic vehicular networks, we develop an innovative federated deep reinforcement learning framework that integrates advanced technologies from GenAI and deep learning. This framework introduces a novel AdFL approach to create a decentralized and adaptive learning environment. Additionally, we employ a multi-agent DRL technique, trained within this adaptive federated learning framework, allowing agents to independently learn optimal task offloading strategies. The detailed implementation of these approaches is provided in the following sections.

V. Adaptive Federated Learning for Dynamic Vehicular Networks

In this section, we introduce the **Ad**aptive **F**ederated **L**earning (AdFL) algorithm, specifically designed for the dynamic and heterogeneous nature of IoV networks. The AdFL framework is responsible for training decentralized models across vehicles and RSUs using context-aware latent representations. AdFL addresses the challenges of non-IID data and dynamic network conditions by introducing client selection criteria, semi-synchronous aggregation, and gradient norm-based weighting. The need for low latency, energy efficiency, and decentralized decision-making in IoV networks necessitates a shift from traditional centralized approaches to a more distributed and adaptive framework. Thus, the algorithm leverages the capabilities of GenAI, specifically, CVAE, and transformer layers to enhance learning efficiency, preserve privacy, and adapt to the rapidly changing conditions of the IoV. The integration of CVAE allows for context-aware latent representations, while transformer layers excel in capturing sequential dependencies and contextual relationships within the data, making the AdFL algorithm highly responsive and scalable.

This approach is crucial as it addresses the unique challenges of IoV networks, where traditional federated learning methods fall short. By enabling real-time adaptation and decentralized learning, AdFL has the potential to revolutionize vehicular network training, leading to more efficient, reliable, and scalable systems. The steps in AdFL are given in detail below.

A. Client Selection and Initialization

*1) **Client Selection Criteria**:* In the AdFL framework, selecting vehicles (clients) for participation in the FL process is critical. The selection process is based on several factors that ensure the efficiency and effectiveness of the learning process:

- Computational Capability (f_n): Each vehicle V_n must possess a minimum computational power, f_{min}, to handle the local training tasks effectively which ensures that selected vehicles can contribute meaningful updates to the global model.
- Energy Levels (E_n): The energy level of the vehicle must exceed a predefined threshold, E_{min} which prevents vehicles from depleting their energy resources during the training process.
- Connectivity and Mobility (t_n^{stay}): Vehicles must maintain stable connectivity within the communication range of the RSU for a minimum duration, T_{min}

Mathematically, a vehicle V_n is eligible for participation if:

$$f_n \geq f_{\text{min}}, \quad E_n \geq E_{\text{min}}, \quad t_n^{\text{stay}} \geq T_{\text{min}}\,. \tag{24}$$

*2) **Global Model Initialization**:* In IoV networks, traditional FL models, often based on simple neural networks, struggle to capture the complex dependencies and dynamic nature of the environment, limiting their effectiveness in real-time decision-making. To overcome these limitations, we design a global model that incorporates transformer layers, capable of capturing long-range dependencies and contextual relationships in the data. The algorithm for the construction of the global model with transformer layers is given in **Algorithm 1**.

The global model, denoted by $w^{(0)}$, is initialized with parameters from a pre-trained IoV-relevant dataset. This approach provides a strong foundation, accelerating convergence during federated learning by leveraging transformers to effectively model the intricate dependencies inherent in IoV networks [38]. Transformers are integral to the global model, enabling it to capture long-range dependencies and contextual relationships between features such as vehicle speed, energy levels, and communication capabilities [39]. The key components of the transformer's architecture include the following components:

- **Input Embedding:** Each input feature vector $\mathbf{x}_n$ from the DT representations of vehicles and RSUs is transformed

Algorithm 1 Global Model with Transformer Layers

Input: Initial global model parameters $w^{(0)}$.
Embedding:
Transform input feature vector $\mathbf{x}_n$ into high-dimensional embedding using (25).
Positional Encoding:
Inject sequence information using (26).
Self-Attention Mechanism:
Compute attention weights and transform input features using (27).
Feedforward Neural Network:
Capture non-linear relationships using (28).
Layer Normalization and Residual Connection:
Normalize and add residuals as per (29).
Output Layer:
Generate predictions using (30).
Output: Updated global model $w^{(r)}$.

into a high-dimensional embedding. This embedding process is mathematically represented as:

$$\mathbf{e}_n = W_e \mathbf{x}_n + \mathbf{b}_e \,, \tag{25}$$

where W_e and $\mathbf{b}_e$ are the embedding matrix and bias term.

- **Positional Encoding:** Since transformers are inherently order-agnostic, we introduce positional encodings to inject sequence information into the model [40]:

$$\mathbf{p}_n(i) = \begin{cases} \sin\left(\frac{i}{10000^{2j/d}}\right), & \text{if } j \text{ is even}, \\ \cos\left(\frac{i}{10000^{2j/d}}\right), & \text{if } j \text{ is odd}, \end{cases} \tag{26}$$

where i is the position, j indexes the dimension, and d is the dimensionality of the model.

- **Self-Attention Mechanism:** The core of the transformer is the self-attention mechanism, which allows the model to weigh the importance of different input features dynamically. The attention mechanism is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \,, \tag{27}$$

where $Q = W_q \mathbf{e}_n$, $K = W_k \mathbf{e}_n$, and $V = W_v \mathbf{e}_n$ are the query, key, and value matrices, respectively, and d_k is the dimensionality of the key vectors.

- **Feedforward Neural Network (FFN):** The output of the self-attention layer is passed through a position-wise fully connected feedforward network:

$$\text{FFN}(x) = \text{ReLU}(W_1 x + b_1) W_2 + b_2 \,, \tag{28}$$

where W_1 and W_2 are weight matrices, and b_1 and b_2 are bias terms. This allows the model to capture non-linear relationships within the data.

- **Layer Normalization and Residual Connection:** To stabilize and accelerate training, each sub-layer (self-attention and FFN) is followed by a layer normalization and a residual connection:

$$\mathbf{z}_n = \text{LayerNorm}(\mathbf{x}_n + \text{FFN}(\text{Attention}(Q, K, V))) \,, \tag{29}$$

where $\mathbf{z}_n$ represents the output of the transformer layer.

- **Output Layer:** The output layer generates predictions for the task at hand using a dense layer with linear activation. The output is computed as follows, where W_o represents the weight matrix and b_o represents the bias vector:

$$\mathbf{y}_n = W_o \mathbf{z}_n + b_o \,. \tag{30}$$

The global model update process employs a weighted aggregation strategy to account for heterogeneous data distributions across IoV nodes. In each training round, the global model is updated using local updates from participating clients as:

$$w^{(t+1)} = w^{(t)} + \frac{1}{\sum_{n \in \mathcal{S}t} \alpha_n} \sum n \in \mathcal{S}_t \alpha_n \Delta w_n^{(t)}, \tag{31}$$

where $\alpha_n = |D_n|$ represents the weight for client n based on its dataset size, and $\Delta w_n^{(t)}$ is the local model update. This ensures that clients with larger datasets contribute more substantially, addressing data heterogeneity effectively. Furthermore, semi-synchronous aggregation triggers, either time- or count-based, are used to balance timeliness and model quality, accommodating varying traffic and communication conditions.

The global model's parameters are updated through the FL process, with the initial parameters $w^{(0)}$ distributed to all participating vehicles. The objective function for the global model $J(w)$ during training is to minimize the aggregate loss across all participating vehicles:

$$J(w) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n(w) \,, \tag{32}$$

where $\mathcal{L}_n(w)$ is the local loss function for vehicle V_n. The global model update rule during federated learning is:

$$w^{(t+1)} = w^{(t)} - \eta \nabla J(w^{(t)}) \,, \tag{33}$$

where η is the learning rate.

In the proposed framework, local updates uploaded by vehicles are aggregated at the RSU using a weighted averaging strategy to ensure that clients with larger datasets have a proportionally greater influence on the global model. This is expressed as:

$$w^{(r+1)} = \frac{\sum_{n \in \mathcal{S}r} \alpha_n w_n^{(r)}}{\sum n \in \mathcal{S}_r \alpha_n}, , \tag{34}$$

where $w^{(r+1)}$ is the updated global model after round r, $\mathcal{S}_r$ is the set of participating clients, $w_n^{(r)}$ is the local model from client n, and $\alpha_n = |D_n|$ is the weight proportional to the size of client n's local dataset.

This aggregation mechanism ensures that updates from clients with more representative datasets contribute significantly to the global model, addressing data heterogeneity across IoV networks. Upon receiving the local updates, the RSU computes the weights α_n and performs the aggregation. The updated global model is then transmitted back to all participating clients, enabling the next round of training. This strategy enhances robustness and improves the alignment of the global model with the overall data distribution.

*3) **Digital Twin (DT) Initialization**:* Each vehicle V_n and RSU R_m is paired with a DT, which mirrors the real-time state of the physical entity. The DT framework enhances the AdFL process by providing accurate, up-to-date information on computational resources, energy levels, and mobility characteristics. At time t, the DTs of V_n and R_m, denoted as $\mathrm{DT}_n(t)$ and $\mathrm{DT}_m(t)$, encapsulate these properties:

$$\mathrm{DT}_n(t) = \{f_n, r_n, E_n, \mathbf{v}_n(t), D_n, C_n, T_n\}, \tag{35}$$

$$\mathrm{DT}_m(t) = \{f_m, r_m, E_m\}, \tag{36}$$

where f, r, and E denote computational power, communication capability, and energy levels, respectively.

The DTs dynamically synchronize their parameters with real-time data from the SUMO simulation at fixed intervals of **1 second**. Additionally, they aggregate historical metrics over a **10-second sliding window**:

$$\mathrm{Historical\_DT}_n(t) = \frac{1}{10} \sum_{k=t-10}^{t} \mathrm{DT}_n(k), \tag{37}$$

including metrics such as average delays, energy consumption trends, task success rates, and mobility patterns. These trends help capture temporal dynamics, which are crucial for efficient task offloading and learning.

The combined real-time and historical data provide a robust basis for generating context-aware training samples used in AdFL and AF-MARL. Each training sample combines real-time states (e.g., $\mathrm{DT}_n(t)$), decision variables (e.g., offloading fractions ρ_n), and observed outcomes (e.g., delay, energy consumption, reward). This ensures that models are trained on data reflecting both instantaneous and aggregated conditions, enabling effective decision-making in dynamic IoV environments.

B. CVAE in Adaptive Federated Learning

To handle the high dimensionality and non-linearity of IoV data, the GenAI component utilizes CVAE. These models transform raw DT data into compact, context-aware latent representations that capture complex dependencies, such as the interplay between energy consumption, mobility, and task-specific requirements.

The incorporation of CVAE into the AdFL process significantly enhances learning efficiency by generating context-aware latent representations that better capture the dynamic environment of IoV networks.

The encoder in the CVAE maps the input data $\mathbf{x}_n(t)$ and context $\mathbf{c}_n(t)$ to a latent representation $\mathbf{z}_n(t)$ [41], as:

$$q_\phi(\mathbf{z}_n(t) \mid \mathbf{x}_n(t), \mathbf{c}_n(t)) = \mathcal{N}(\mathbf{z}_n(t) \mid \mu_n(t), \sigma_n^2(t)), \tag{38}$$

where $\mu_n(t)$ and $\sigma_n^2(t)$ are the mean and variance of the latent distribution, respectively. Next, the decoder reconstructs the input data from the latent representation:

$$p_\theta(\mathbf{x}_n(t) \mid \mathbf{z}_n(t), \mathbf{c}_n(t)) = \mathcal{N}(\mathbf{x}_n(t) \mid \hat{\mu}_n(t), \hat{\sigma}_n^2(t)). \tag{39}$$

Algorithm 2 CVAE Process

1: **Input:** Training data $\mathbf{x}_n(t)$ and context $\mathbf{c}_n(t)$.
2: **Encoder:**
3: Compute latent representation $\mathbf{z}_n(t)$ using (38).
4: **Decoder:**
5: Reconstruct input data $\mathbf{x}_n(t)$ from $\mathbf{z}_n(t)$ using (39).
6: **Loss Function:**
7: Minimize the combined loss (40).
8: **Output:** Latent representation $\mathbf{z}_n(t)$.

The CVAE is trained by minimizing a loss function that combines the reconstruction loss and the Kullback-Leibler (KL) divergence:

$$\begin{aligned}\mathcal{L}_{\text{CVAE}} = \mathbb{E}_{q_\phi(\mathbf{z}_n(t)|\mathbf{x}_n(t),\mathbf{c}_n(t))} \left[\log p_\theta(\mathbf{x}_n(t) \mid \mathbf{z}_n(t), \mathbf{c}_n(t))\right] \\ - D_{\text{KL}}\left(q_\phi(\mathbf{z}_n(t) \mid \mathbf{x}_n(t), \mathbf{c}_n(t)) \parallel p(\mathbf{z}_n(t))\right).\end{aligned} \tag{40}$$

The latent representations generated by the CVAE are derived from $\mathbf{x}_n(t) = \mathrm{DT}_n(t)$ (real-time state) and $\mathbf{c}_n(t) = \mathrm{Historical\_DT}_n(t)$ (historical context). These features enable decentralized training of models tailored to the specific conditions of each vehicle and RSU. By capturing nonlinear relationships, such as the interplay between energy consumption and task deadlines, these latent representations enhance the contextual relevance of AdFL.

The CVAE process is detailed in **Algorithm 2**.

1) Role of Digital Twins in Training Data Preparation: The DT framework plays a pivotal role in enabling the training and operationalization of the AdFL and AF-MARL frameworks by serving as the primary source of real-time and historical data. Each vehicle V_n and RSU R_m is paired with a DT, which mirrors its state and aggregates key metrics as in Eqs. (35) and (36).

The DTs generate training samples by combining real-time states, decision variables, and observed outcomes. Real-time states are captured at specific time instances (e.g., $\mathrm{DT}_n(t)$ for vehicles and $\mathrm{DT}_m(t)$ for RSUs), decision variables include task offloading fractions ρ_n, and observed outcomes comprise delay, energy consumption, or reward values from previous actions. These training samples enable the AdFL and AF-MARL frameworks to address heterogeneity and dynamic conditions in IoV networks effectively.

The AdFL framework leverages the DT-generated data by combining real-time states and aggregated historical data to train decentralized models that reflect the unique operating conditions of each vehicle or RSU. This integration ensures that the AdFL process can adapt to varying environmental conditions. For the AF-MARL framework, the DT data defines the state space representations of agents, which are further processed through the CVAE module to produce context-aware latent representations. These representations encapsulate complex dependencies, such as the interplay between energy consumption, task deadlines, and mobility metrics, providing agents with a comprehensive view of their environment. This integration of DT data ensures efficient and robust task-offloading decisions, optimizing delay, energy consumption, and cost trade-offs.

C. Local Model Training

Once the vehicles are selected, and their DTs initialized, each vehicle V_n trains its local model using the context-aware latent representations generated by the CVAE. The local model updates are computed using stochastic gradient descent (SGD) on the local dataset D_n:

$$w_n^{(t)} = w^{(t-1)} - \eta \nabla \mathcal{L}_n(w^{(t-1)}; D_n)\,, \tag{41}$$

where η is the learning rate, and $\mathcal{L}_n(w; D_n)$ is the local loss function for vehicle V_n.

The local training process enables each vehicle to adapt the global model to its specific environment, capturing the unique characteristics and challenges of its operational context, such as variations in traffic density, signal strength, and energy consumption patterns.

D. Gradient Norm Assessment

To evaluate the contribution potential of each vehicle's local model to the global model, AdFL incorporates a gradient norm assessment across multiple layers. This approach provides a more comprehensive view of the model's training progress and ensures that only the most relevant updates are included in the global model aggregation. The gradient norm for a specific layer l of the local model is calculated as:

$$\sigma_{n,l}^{(t)} = \left| \frac{\partial \mathcal{L}_n}{\partial w_{l,n}^{(t)}} \right|_2\,, \tag{42}$$

where $w_{l,n}^{(t)}$ represents the weights of layer l at time t. The overall contribution potential $\sigma_n^{(t)}$ is then computed as a weighted sum of the gradient norms across the selected layers:

$$\sigma_n^{(t)} = \sum_{l \in \mathcal{L}} \omega_l \sigma_{n,l}^{(t)}\,, \tag{43}$$

where ω_l is the weight assigned to layer l, and $\mathcal{L}$ is the set of layers considered for evaluation.

E. Semi-Synchronous Model Aggregation Triggers

Given the dynamic nature of IoV networks, the aggregation of local model updates in AdFL is performed in a semi-synchronous and distributed manner. This approach allows the system to balance the trade-off between timely updates and the quality of the global model. Two primary mechanisms are employed for triggering the aggregation:

- **Time-Based Trigger:** During high traffic periods, the RSU delays aggregation for a fixed time interval τ_{agg}, ensuring regular global model updates even if all vehicles have not completed their local training.
- **Count-Based Trigger:** In lower traffic conditions, the RSU aggregates updates after receiving a sufficient number of updates N_{agg} from the participating vehicles. This mechanism ensures that the global model is updated only when enough new information is available, preserving computational resources.

The global model update following the aggregation is:

$$w^{(t+1)} = w^{(t)} + \frac{1}{|\mathcal{S}_t|} \sum_{n \in \mathcal{S}_t} \Delta w_n^{(t)}\,, \tag{44}$$

Algorithm 3 CVAE Enhanced Adaptive Federated Learning

Initialize global model $w^{(0)}$ and other parameters.
for $r \leftarrow 1$ to R_{max} **do**
    Client Selection:
    for each vehicle V_n **do**
        Calculate f_n, E_n, t_n^{stay}.
        if $f_n \geq f_{\text{min}} \wedge E_n \geq E_{\text{min}} \wedge t_n^{\text{stay}} \geq T_{\text{min}}$ **then**
            Select vehicle V_n for local training.
        end if
    end for
    if no vehicles are selected **then**
        Continue to the next round.
    end if
    Global Model Initialization:
    Construct the global model $w^{(r)}$ using **Algorithm 1**.
    Local Model Training:
    for each selected vehicle V_n **do**
        Initialize local model $w_n^{(r)}$.
        Generate context-aware latent representations using **Algorithm 2**.
        for each training epoch k **do**
            Select local dataset D_n.
            Update local model $w_n^{(r,k)}$ using SGD (41).
            Compute local loss $\mathcal{L}_n(w_n^{(r,k)}; D_n)$.
        end for
        Upload updated local model $w_n^{(r)}$ to the RSU.
    end for
    Gradient Norm Assessment:
    for each selected vehicle V_n **do**
        for each layer $l \in \mathcal{L}$ of local model **do**
            Calculate gradient norm $\sigma_{n,l}^{(t)}$ as per (42).
        end for
        Compute overall contribution potential $\sigma_n^{(t)}$ (43).
    end for
    Semi-Synchronous Aggregation:
    if Time-based aggregation (high traffic) **then**
        Wait for a fixed time interval τ_{agg}.
    else if Count-based aggregation (low traffic) **then**
        Wait until N_{agg} updates are received.
    end if
    Aggregate global model $w^{(r+1)}$ using (44).
    Adaptive Model Adjustment:
    Dynamically adjust σ_{min}, τ_{agg}, and N_{agg} as (45).
    Update CVAE latent space (38), (39), (40).
end for
Output: Final global model $w^{(R_{\text{max}})}$.

where $\mathcal{S}_t$ is the set of vehicles whose updates are aggregated at time t, and $\Delta w_n^{(t)}$ represents the update from vehicle V_n.

F. Adaptive Model Adjustment and Feedback

1) Adaptive Threshold Adjustment: The AdFL framework dynamically adjusts gradient norms and aggregation triggers based on real-time feedback, ensuring the learning process remains efficient and responsive to the evolving conditions of the vehicular network.

The adjustment of the minimum gradient norm threshold, σ_{min}, time-based trigger interval, τ_{agg}, and count-based trigger threshold, N_{agg}, is performed as follows:

$$\begin{aligned} \sigma_{\text{min}}^{(t+1)} &= \sigma_{\text{min}}^{(t)} + \Delta\sigma(t)\,, \\ \tau_{\text{agg}}^{(t+1)} &= \tau_{\text{agg}}^{(t)} + \Delta\tau(t)\,, \\ N_{\text{agg}}^{(t+1)} &= N_{\text{agg}}^{(t)} + \Delta N(t)\,. \end{aligned} \tag{45}$$

2) CVAE-Driven Adaptation: The CVAE continuously updates its latent space representations to adapt to changing contexts, ensuring that the federated learning process remains sensitive to variations in vehicular conditions, such as traffic load, energy availability, and communication quality. Integrating CVAE with the AdFL framework enhances the accuracy and contextual relevance of updates, thereby improving the overall effectiveness of the global model.

The models trained using AdFL are further utilized in the AF-MARL framework, which optimizes task offloading decisions, as described in the next section. The procedure of AdFL is detailed in **Algorithm 3**.

VI. Adaptive Federated Multi-Agent Reinforcement Learning

In this section, we introduce the **A**daptive **F**ederated **M**ulti-**A**gent **R**einforcement **L**earning (AF-MARL) algorithm, a cutting-edge approach specifically designed to optimize task offloading to minimize overall costs, including energy consumption and delay, while maximizing task completion rates in dynamic vehicular networks. Building on the outputs of AdFL, the AF-MARL framework focuses on optimizing task offloading strategies. This module combines reinforcement learning with federated learning principles to handle the decentralized nature of IoV networks while accounting for real-time and predictive metrics provided by DTs. AF-MARL is distinguished by its advanced features, specifically tailored to address the complexities of IoV networks. It employs a continuous action space, which allows for precise decision-making, and utilizes policy-based learning to dynamically map states to actions, enabling the system to adapt effectively to real-time changes in the network. The algorithm is built on an actor-critic architecture, where the actor-network selects actions and the critic network evaluates them, facilitating iterative refinement of decision-making.

A. Actor-Critic Architecture for AF-MARL

In the AF-MARL framework, each agent represents a vehicle or RSU within the IoV network, operating in a dynamic environment where states continuously evolve based on real-time conditions. These agents interact with their environment, making decisions on task offloading and resource allocation based on their observed states and predefined policies. The interactions among agents are modeled within a Markov game framework, ensuring that decisions are influenced by the states and actions of other agents in the network. The main properties of the environment are listed below:

1) Decentralized Agent Coordination: The distributed design of AF-MARL allows each agent, whether a vehicle or RSU, to function independently while coordinating through decentralized learning and decision-making. By relying on local observations and interactions, agents reduce communication overhead and enhance scalability and robustness [42], [43]. This approach enables the system to swiftly adapt to dynamic changes in network topology, such as vehicle mobility, RSU failures, or variations in communication quality, without relying on a centralized controller.

2) State and Action Spaces: Each agent observes its dynamically updated state $\mathbf{s}_n(t)$, represented as:

$$\mathbf{s}_n(t) = \{f_n(t), r_n(t), E_n(t), D_n, C_n, T_n, \text{AdFL context}\}\,. \tag{46}$$

The action space $\mathbf{a}_n(t)$ for each agent includes continuous decisions such as the task offloading fraction $\rho_n(t)$:

$$\mathbf{a}_n(t) = \{\rho_n(t)\}, \quad \rho_n(t) \in [0,1]\,. \tag{47}$$

Algorithm 4 Adaptive Federated Multi-Agent Reinforcement Learning

```
procedure AF-MARL(γ, τ, N_agg, τ_agg)
    Initialize actor network μ_θ and critic networks Q_θ1, Q_θ2.
    Initialize target networks μ_θ'(s), Q_θ'1(s,a), Q_θ'2(s,a) with θ' ← θ.
    for each episode e do
        Initialize state s_n(t) for each agent V_n
        for each timestep t do
            Action Selection:
            Select action a_n(t) = μ_θ(s_n(t)) for each agent.
            Execute a_n(t), observe reward r_n(t) and next state s_n(t+1).
            Store transition (s_n(t), a_n(t), r_n(t), s_n(t+1)) in replay buffer D.
            Critic Update:
            Sample random minibatch of transitions (s, a, r, s') from replay buffer D.
            Compute target value (49).
            Update critic networks by minimizing (50).
            Actor Update:
            Update actor network μ_θ(s) using (51).
            Target Networks Soft Update:
            Update target networks using (52).
            Aggregation:
            if Count-Based: N_agg updates received then
                Aggregate local models as (44).
            else if Time-Based: τ_agg elapsed then
                Aggregate local models as (44).
            end if
        end for
    end for
    Return global model w^(t+1)
end procedure
```

B. Integration of AdFL with Actor-Critic-Based AF-MARL

The AF-MARL algorithm employs a dual-critic architecture to stabilize learning. The actor network $\mu_\theta(s)$ selects actions, while two critic networks $Q_{\theta_1}(s,a)$ and $Q_{\theta_2}(s,a)$ evaluate the actions. The critics are trained to minimize the Bellman error:

$$\mathcal{L}_{\text{critic}_i}(\theta_i) = \mathbb{E}_{(s,a,r,s')}\left[\left(Q_{\theta_i}(s,a) - y\right)^2\right], \tag{48}$$

$$y = r + \gamma \min_{i=1,2} Q_{\theta_i'}(s', \mu_{\theta'}(s')), \tag{49}$$

where γ is the discount factor and θ_i' are the parameters of the target networks.

The two critics, Q_{θ_1} and Q_{θ_2}, are independent networks with no shared parameters or states. This design ensures that each critic provides an unbiased evaluation of the action-state pairs, enhancing the robustness of the training process. The synchronization between the critics occurs only through periodic updates to their respective target networks, $Q_{\theta_1'}$ and $Q_{\theta_2'}$, using a soft update mechanism:

$$\theta_i' \leftarrow \tau\theta_i + (1-\tau)\theta_i', \tag{50}$$

where $\tau \ll 1$ is the soft update rate. This approach ensures stable target values for the Bellman update while maintaining the independence of the critic networks. The independence of the critics reduces overestimation bias by leveraging the minimum of their evaluations in the target computation.

The policy gradient is computed as:

$$\nabla_\theta J(\mu_\theta) = \mathbb{E}_{s\sim\mathcal{D}}\left[\nabla_a Q_{\theta_1}(s,a)\big|_{a=\mu_\theta(s)} \nabla_\theta \mu_\theta(s)\right]. \tag{51}$$

Next, to maintain stability during training, target networks are used. The target networks for both the actor and the critics are updated using a soft update mechanism:

$$\theta_i' \leftarrow \tau\theta_i + (1-\tau)\theta_i', \tag{52}$$

where $\tau \ll 1$ is the soft update rate.

The reward function $R_n^{(t)}$ is designed to minimize the overall cost $\Omega_n(t)$, which includes the offloading delay $t_n^{\text{off}}(t)$, energy consumption $E_n^{\text{total}}(t)$, and the offloading service cost $\mathcal{P}_n(t)$:

$$R_n^{(t)} = \frac{1}{\Omega_n(t)} = \frac{1}{\left(\frac{t_n^{\text{off}}(t)}{T_n} + \frac{E_n^{\text{total}}(t)}{E_{\text{max}}} + \frac{\mathcal{P}_n(t)}{P_{\text{max}}}\right)}. \tag{53}$$

The gradients for the actor network are computed to maximize the expected cumulative reward:

$$\mathcal{L}_{\text{actor}}(\theta) = -\mathbb{E}_{s\sim\mathcal{D}}\left[Q_{\theta_1}(s, \mu_\theta(s))\right]. \tag{54}$$

The critic networks minimize the Bellman error through their respective loss functions, ensuring accurate value estimation for the chosen actions as:

$$\mathcal{L}_{\text{critic}}(\theta_i) = \mathbb{E}_{(s,a,r,s')}\left[\left(Q_{\theta_i}(s,a) - y\right)^2\right]. \tag{55}$$

In AF-MARL, local model updates are aggregated using a semi-synchronous approach, which ensures a balance between timely updates and communication efficiency. The aggregation process employs two distinct triggers: a **count-based trigger** that initiates aggregation once a predefined number of updates are received, and a **time-based trigger** that aggregates updates at fixed time intervals, as explained in Section V-E. After the reinforcement learning process, the global model is updated based on the aggregated local models as (44). The AF-MARL algorithm is detailed in **Algorithm 4**.

The effectiveness of the integrated framework, including DTs, GenAI, AdFL, and AF-MARL, is evaluated through extensive simulations, as detailed in the following sections.

VII. Performance Evaluation

In this section, we evaluate the performance of the proposed AdFL-based AF-MARL frameworks in dynamic vehicular networks. The evaluation is conducted through extensive simulations designed to reflect real-world conditions. We begin by describing the dataset preparation process, followed by the simulation setup used to assess the performance of our approach.

A. Simulation Setup

The simulations were conducted on a high-performance computing system configured with an NVIDIA T400 GPU 4GB, a 512GB SSD, and an Intel(R) Core(TM) i7-14700 CPU with 20 cores and a base speed of 2.10 GHz. Python 3.8 was used as the programming environment, with libraries such as NumPy, Matplotlib, and TensorFlow employed for implementing the AdFL and AF-MARL algorithms.

To simulate the dynamic environment of IoV networks, we used SUMO (Simulation of Urban MObility) integrated with Python via the Traffic Control Interface (TRACI) [24], [44]. The simulation modeled a 5 km $\times$ 5 km urban area, including realistic city layouts with intersections and traffic controls. Traffic scenarios included high density at 60 vehicles/km^2 for peak hours, low density at 20 vehicles/km^2 for off-peak times, and mixed conditions varying between 20 to 60 vehicles/km^2.

For the simulation in this paper, the criteria for selecting vehicles (clients) in the FL process were determined based on computational capability, energy levels, and connectivity/mobility thresholds. A minimum computational capability of $f_{\text{min}} = 1.5$ GHz ensures that vehicles can complete local training tasks within a reasonable time frame (e.g., a task requiring 10^{11} FLOPs would take 67 seconds on a 1.5 GHz processor). An energy threshold of $E_{\text{min}} = 150$ Wh accounts for the energy consumed during training and communication, ensuring sufficient reserves for reliable participation without impacting primary vehicle operations. For connectivity, $T_{\text{min}} = 2$ seconds ensures adequate time to transmit model updates to RSUs (e.g., a 2 MB update at 10 Mbps takes 1.6 seconds, leaving a margin for synchronization). These thresholds, derived from empirical observations and simulation studies, ensure efficient and reliable client selection while addressing the dynamic nature of IoV environments.

To enhance realism, the DT framework was implemented to provide a virtual replica of each vehicle and RSU, capturing their real-time states and historical metrics. Each DT was structured as a vector of parameters, including computational power, communication capability, energy level, mobility characteristics (e.g., velocity and position), and task-specific details such as task size and deadlines.

TABLE III: SUMO Simulation Setup

Parameter	Value
Vehicle Types	Cars, Buses, Trucks
Max Speed	13 m/s
Acceleration	3 m/s^2
Deceleration	9 m/s^2
Lane Configuration	3 lanes per direction
Traffic Control	Traffic lights, Stop signs
Communication Range	Up to 1000 meters
Computation Power	$1.5 - 3.5$ GHz
Energy Capacity	$200 - 1500$ Wh
Stay Time	$2 - 10$ seconds
Task Size	$0.5 - 2.0$ MB
CPU Cycles	$0.5 - 1.5$ GHz

TABLE IV: Simulation Parameters

Param.	Value	Param.	Value
$X \times X$	5×5 km^2	N	200
M	20	f_n	$1.5 - 2.5$ GHz
f_m	$2.5 - 3.5$ GHz	r_n	$[20 - 50]$ Mbps
r_m	$[100 - 150]$ Mbps	E_n	$200 - 300$ Wh
E_m	$1000 - 1500$ Wh	$\mathbf{v}_n^{\text{avg}}$	$10 - 20$ m/s
$\mathbf{v}_n^{\text{rand}}$	$[-2, 2]$ m/s	Δt	1 s
T_n	$0.1 - 1.0$ s	C_n	$0.5 - 1.5$ GHz
D_n	$0.5 - 2.0$ MB	P_n	0.1 W
W_n	10 MHz	β	3
h_n	$10^{-7} - 10^{-8}$	N_0	-174 dBm/Hz
κ_n	10^{-6}	δ	0.1
ρ_n	$[0, 1]$	d_n	$[50 - 1000]$ m

The synchronization process was implemented using the TraCI API, which facilitated real-time data exchange between SUMO and the DT framework at a fixed interval of 1 second. Updates to DT attributes reflected data such as vehicle positions, speeds, communication ranges, and task-processing energy consumption. Additionally, the DTs aggregated historical data over a 10-second sliding window, which included task success rates, average delays, and energy consumption trends. This historical data was utilized for predictive modeling, including training the CVAE to generate latent representations capturing nonlinear relationships among vehicular parameters. The SUMO parameters are provided in **Table III**.

The detailed parameters and hyperparameters used in the simulation of this study are listed in **Table IV** and **Table V**, respectively.

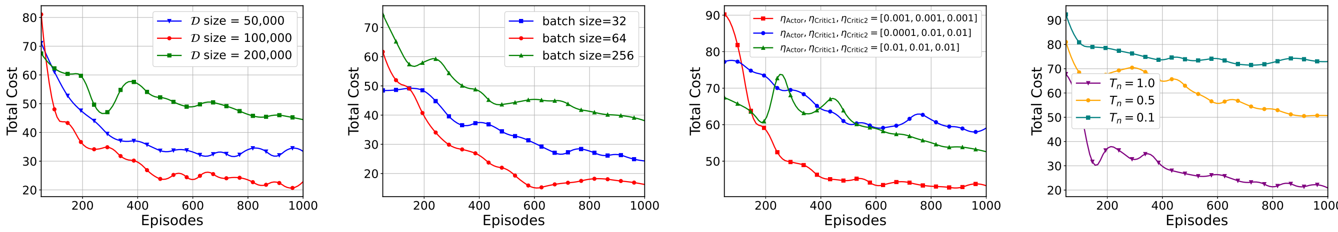
B. Deployment Feasibility in Real-World Environments

The integration of advanced components such as CVAE, transformer layers, and multi-agent DRL in the proposed framework necessitates careful consideration for real-world deployment. Below, we outline the strategies employed to address the associated challenges and ensure scalability and adaptability in diverse IoV network conditions:

- *Resource Management Across Nodes:* Computationally intensive tasks, such as CVAE-based encoding and global model training, are offloaded to MEC servers or RSUs, which have higher computational capacities. Vehicles execute lightweight decision-making models locally, reducing the burden on resource-constrained nodes and aligning with the hierarchical design of IoV networks.
- *Semi-Synchronous Aggregation:* To address synchronization delays caused by node heterogeneity, the AdFL framework employs a semi-synchronous aggregation mechanism. This allows faster nodes to proceed without waiting indefinitely for slower nodes, striking a balance between update quality and timeliness in dynamic vehicular environments.
- *Real-Time Decision-Making and Scalability:* Digital twins (DTs) maintain up-to-date representations of vehicles and edge servers, enabling efficient resource allocation and task offloading. The decentralized design of the multi-agent DRL framework supports independent agent operations, reducing reliance on centralized coordination and ensuring scalability in large-scale deployments.
- *Adaptation to Node Variability:* The task offloading process dynamically adjusts to account for differences in computational and communication capabilities. Nodes with lower resources prioritize offloading to RSUs or nearby vehicles with higher capacities. Sliding-window aggregation of historical data in DTs further enhances adaptability by capturing temporal patterns.
- *Future Validation Efforts:* Future work involves validating the framework through hardware-in-the-loop (HiL) simulations and pilot deployments in controlled vehicular testbeds. These efforts aim to address challenges such as communication unreliability and hardware

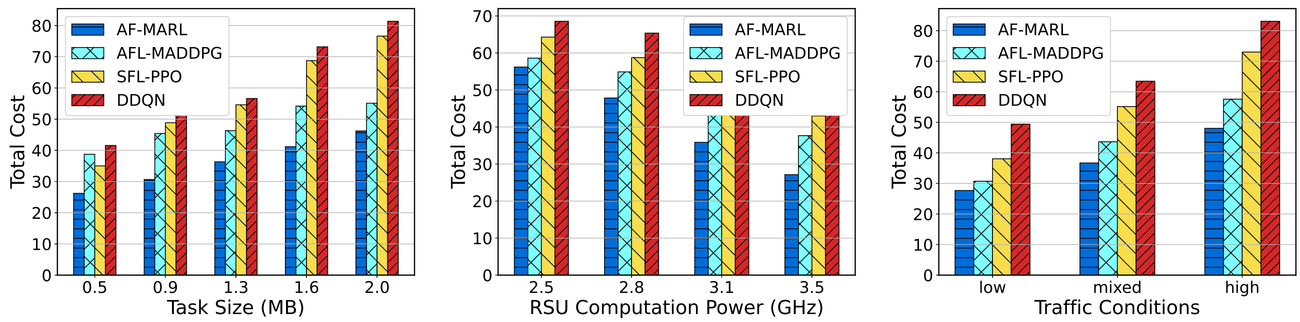
TABLE V: Hyperparameters for AdFL, CVAE, and AF-MARL

Param.	Value	Param.	Value
AdFL			
f_{min}	1.5 GHz	E_{min}	150 Wh
T_{min}	2 s	τ_{agg}	5 s
N_{agg}	10	η_{AdFL}	0.01
Batch Size (AdFL)	64	Optimizer	Adam
Activation (Transformer)	ReLU	η_{Global}	0.01
CVAE			
Latent Dim.	64	η_{CVAE}	0.001
Batch Size (CVAE)	32	KL Coefficient	0.1
Encoder Layers	3	Decoder Layers	3
Activation (CVAE)	ReLU	Output Activation	Sigmoid
Regularization	L2	Dropout Rate	0.2
AF-MARL			
γ	0.99	τ	0.005
η_{Critic1}	0.001	η_{Critic2}	0.001
η_{Actor}	0.001	Buffer Size	$100,000$
Batch Size (AF-MARL)	64	Noise Std. Dev.	0.2
Noise Decay	0.99	Discount Factor	0.99
Soft Update Rate	0.005	Reward Scaling	1.0
Critic Layers	2	Actor Layers	2
Activation (Critic)	ReLU	Activation (Actor)	Tanh
Optimizer (AF-MARL)	Adam	Exploration Rate	0.15



(a) Cost for different $\mathcal{D}$ sizes. (b) Cost for different DRL batch sizes. (c) Impact of different LR on cost. (d) Impact of T_n on cost.

Fig. 2: Total cost analysis for the parameters $\mathcal{D}$, batch size, LR, T_n.



(a) Cost with different task-sizes. (b) Cost with different RSU computation power. (c) Cost for different traffic conditions.

Fig. 3: Total cost analysis for different task sizes, RSU computation capacity, and traffic conditions.

failures, guiding further improvements for large-scale IoV applications.

This multi-faceted approach ensures that the proposed framework is resource-efficient, scalable, and adaptable, making it viable for deployment in real-world IoV environments.

C. Benchmark Schemes

We compare the proposed framework against the following methods:

1) **AFL-MADDPG**: Asynchronous federated learning (AFL) proposed in [23], with multi-agent deep deterministic policy gradient (MADDPG).
2) **SFL-PPO**: Synchronous federated learning (SFL), where all vehicles synchronize updates at each iteration, utilizing proximal policy optimization (PPO) for decision-making.
3) **DDQN**: Double Deep Q-Network (DDQN), a non-distributed DRL approach that learns task offloading strategies without the benefits of federated learning or multi-agent coordination.
4) **non-DRL**: This approach offloads tasks to the closest RSU with the lowest current load, optimizing for immediate resource availability but without considering long-term impacts.

VIII. Result Analysis

In this section, we present the simulation results and analysis of the proposed digital twin-assisted IoV framework.

A. Total cost analysis

1) Cost Analysis Across Different Parameter Setting: Fig. 2(a) shows that a replay buffer size of $100,000$ yields the lowest total cost across episodes in the AF-MARL framework, balancing recent experience exploitation with data diversity. The $50,000$ buffer, while effective, leads to slightly higher costs due to limited data variety, whereas the $200,000$ buffer causes slower convergence and fluctuations, likely from overfitting to outdated experiences. In Fig. 2(b), the batch size of 64 achieves the lowest total cost by effectively balancing learning efficiency and stability. The smaller batch size of 32 leads to slower convergence and higher costs due to insufficient data variability, while the larger size of 256 stabilizes learning but includes too much older data, causing slower initial learning and higher overall costs. Fig. 2(c) shows that the learning rate (LR) combination $[\eta_{\text{Actor}}, \eta_{\text{Critic1}}, \eta_{\text{Critic2}}] = [0.001, 0.001, 0.001]$ achieves the best performance, with rapid cost reduction and stable convergence. Moderate learning rates enable effective policy learning and robust updates, while higher rates $[0.01, 0.01, 0.01]$ cause instability due to aggressive updates, and lower rates $[0.0001, 0.01, 0.01]$ slow convergence, limiting adaptability. Fig. 2(d) examines the impact of different task deadlines T_n on the total cost. With a uniform deadline across tasks, the results show that a relaxed deadline ($T_n = 1.0$ s) allows for optimal resource allocation, minimizing total costs. A tighter deadline ($T_n = 0.1$ s) restricts scheduling flexibility, leading to higher costs, while the intermediate $T_n = 0.5$ s offers a balanced performance. In subsequent experiments, task deadlines are dynamically varied between 0.1 and 1.0 seconds to better reflect realistic vehicular network conditions.

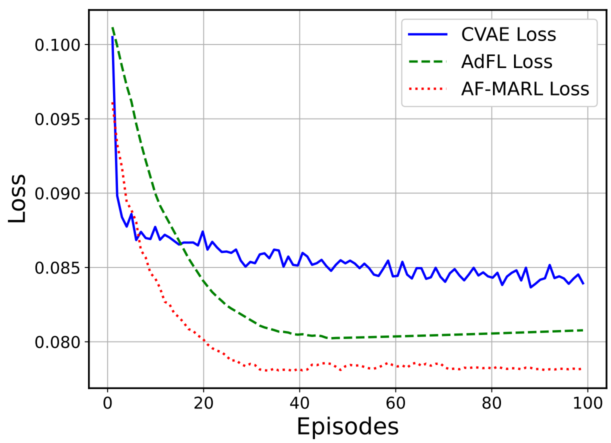
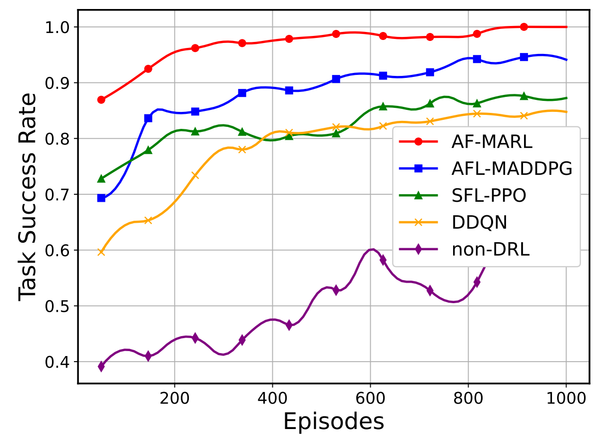
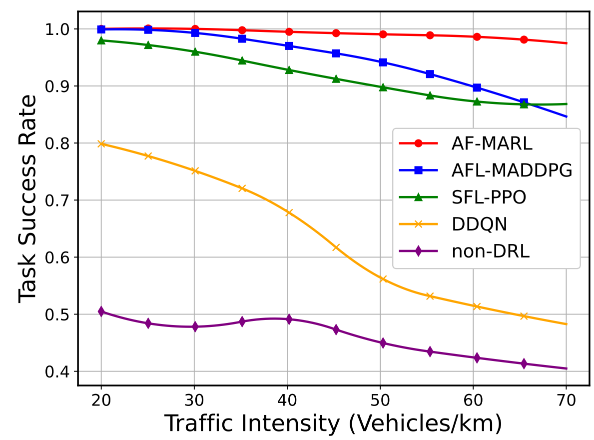


Fig. 4: Loss vs. Episodes



(a) Ratio of successful offloaded tasks vs. Episodes.



(b) Ratio of successful offloaded tasks for different traffic intensity.

Fig. 5: Ratio of successful offloaded tasks.

Thus, based on these findings, we select a replay buffer size of $100,000$, batch size of 64, and learning rate combination of $[0.001, 0.001, 0.001]$ for optimal performance in our framework.

2) Cost Analysis Across Varying Conditions: In Fig. 3(a), total costs increase with task size across all methods. AF-MARL consistently achieves the lowest costs due to its AdFL mechanism combined with multi-agent DRL, which effectively manages the increased computational demand. AFL-MADDPG and SFL-PPO incur higher costs because their static, less context-aware strategies struggle with larger tasks. DDQN performs the worst, particularly as task size grows, due to its inefficiency in handling larger data loads. Fig. 3(b) demonstrates that increasing the RSU computation power from 2.5 GHz to 3.5 GHz results in a reduction of total costs across all methods. The AF-MARL framework exhibits the most significant cost reduction, leveraging its real-time, context-aware decision-making capabilities facilitated by the DT framework. AFL-MADDPG and SFL-PPO benefit less due to their inflexible offloading strategies, while DDQN shows minimal improvement, unable to fully leverage the increased computational capacity. In Fig. 3(c), total costs rise with increasing traffic density. AF-MARL adapts well to varying traffic conditions, leveraging its semi-synchronous FL and multi-agent DRL framework to adjust dynamically to real-time traffic, minimizing costs even under high-density scenarios. AFL-MADDPG and SFL-PPO struggle more with mixed or high-traffic conditions due to their less responsive learning frameworks, while DDQN's performance deteriorates significantly under heavy traffic, leading to inefficient task allocation and higher costs.

It is important to note that fixed conditions considered in Fig. 3 are specific to this particular assessment, while our framework generally considers heterogeneous conditions as outlined in Table IV.

B. Loss

In Fig. 4, the loss trends for CVAE, AdFL, and AF-MARL are compared across episodes. The CVAE loss starts the highest and gradually decreases, reflecting the process of learning effective latent representations for the task. However, its stabilization at a higher loss level compared to AdFL and AF-MARL indicates that, while effective, CVAE alone does not fully optimize the overall learning objective. The AdFL loss begins at a lower level and decreases more rapidly than the CVAE loss, stabilizing around episode 50. This suggests that the AdFL process efficiently utilizes the context-aware representations generated by the CVAE to fine-tune the model, leading to improved performance. The AF-MARL loss starts at the lowest level and decreases the fastest, stabilizing much earlier than both the CVAE and AdFL losses. This rapid decline and lower stabilization point demonstrate the strength of the integrated multi-agent DRL approach within the AF-MARL framework. By effectively combining the adaptive federated learning and context-aware latent representations, AF-MARL achieves superior optimization, minimizing loss more effectively and ensuring robust performance in dynamic vehicular environments. This emphasizes the efficiency and effectiveness of the proposed framework, with all components working together to achieve optimal system performance.

C. Task Success Rate

In Fig. 5(a), the task success rate for various algorithms is shown, reflecting the percentage of offloaded tasks completed over time. AF-MARL consistently achieves the highest task success rate, approaching 100% as the episodes progress. This is due to its advanced features such as the AdFL mechanism, multi-agent DRL, CVAE, and transformer layers, which together enable effective context-aware decision-making and adaptability in dynamic environments. AFL-MADDPG also performs well but shows limitations in adaptability because vehicles train independently and asynchronously, which can lead to suboptimal task offloading decisions under varying conditions. SFL-PPO performs moderately, benefiting from policy optimization but lacking the context-aware capabilities of AF-MARL. DDQN and non-DRL approaches exhibit the lowest success rates, particularly as task complexity increases, due to their lack of collaborative and adaptive mechanisms.

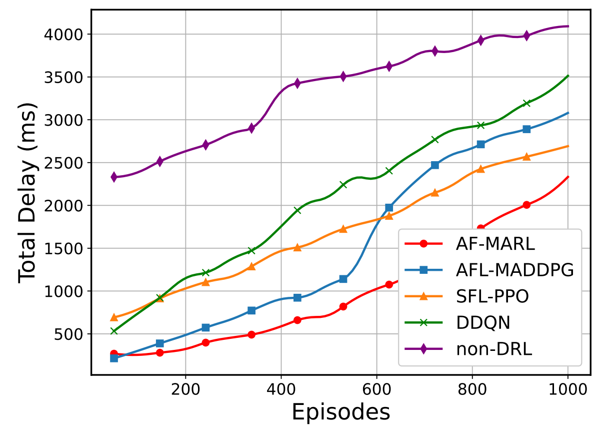
Fig. 5(b) examines the task success rate under varying traffic intensity. AF-MARL maintains the highest success rate even as traffic intensity increases, demonstrating its ability to adapt through real-time context awareness and efficient resource management. AFL-MADDPG, while still effective, shows a slight decline at higher traffic densities due to the lack of synchronized updates across vehicles. SFL-PPO's success rate drops more noticeably with increased traffic, as its less adaptive resource allocation struggles under heavier loads. DDQN and non-DRL methods experience significant declines in success rate under higher traffic, as they are less equipped to manage the increased complexity and demand.

D. Total Energy Consumption & Delay

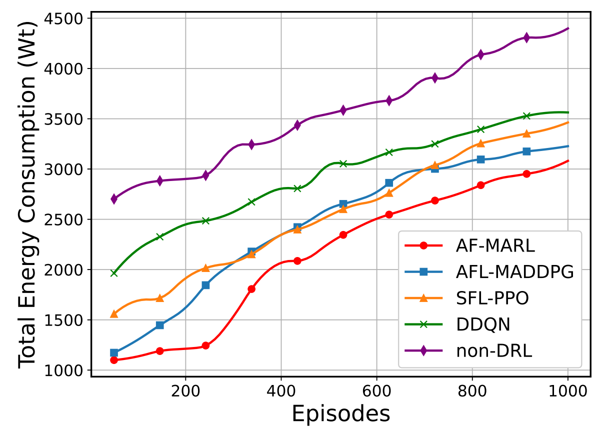
In Fig. 6(a), the total delay across various algorithms is presented, showcasing the efficiency of each method in minimizing task processing times. AF-MARL consistently achieves the lowest delay, leveraging the integration of CVAE and transformer layers for rapid, context-aware decision-making that optimizes task offloading and processing. AFL-MADDPG, while effective, incurs slightly higher delays due to its asynchronous nature, which, while reducing synchronization time, can lead to less coordinated decision-making across vehicles.

In Fig. 6(b), we observe the total energy consumption across the different methods over training episodes. AF-MARL again demonstrates the lowest energy consumption, benefiting from its ability to dynamically adjust to network conditions and distribute tasks efficiently. AFL-MADDPG and SFL-PPO show a trade-off in energy consumption. SFL-PPO, with its synchronous updates, ensures consistent global model updates, but the required coordination introduces additional energy overhead. AFL-MADDPG, on the other hand, reduces this overhead through asynchronous updates, though at the cost of potentially less optimal task decisions in real-time due to the lack of simultaneous updates. Both DDQN and non-DRL methods exhibit significantly higher delays and energy consumption. DDQN's lack of collaborative learning leads to bottlenecks and inefficiencies, while the non-DRL approach, relying on static rules without real-time optimization, is slow to adapt to network changes, resulting in the highest delays and energy use.

Overall, Fig. 6 highlights the superiority of our proposed AF-MARL framework in effectively minimizing both delay and energy consumption, demonstrating its ability to balance dynamic network demands better than the other methods.



(a) Total delay vs. Episodes.



(b) Total energy consumption vs. Episodes.

Fig. 6: Total energy consumption & overall delay.

IX. Conclusion and Future Work

In this paper, we proposed a novel DT-assisted IoV framework designed to address the challenges of real-time decision-making and efficient resource management in large-scale, dynamic vehicular networks. By integrating AdFL and AF-MARL, enhanced with CVAE and transformer layers, the framework effectively optimizes partial task offloading and resource allocation across distributed MEC servers. The extensive simulation results demonstrated that the proposed framework consistently outperforms existing methods across various metrics, including total cost, task success rate, energy consumption, and delay. Specifically, AF-MARL exhibited superior adaptability and efficiency, achieving the lowest costs and highest task success rates, even under varying task sizes, RSU computational capacities, and traffic densities. The results also revealed important trade-offs, particularly between AFL-MADDPG and SFL-PPO, highlighting the balance between synchronous and asynchronous learning approaches in managing energy consumption and delay. The proposed framework's ability to dynamically adjust to real-time conditions, utilize context-aware decision-making, and efficiently distribute computational tasks underscores its potential to significantly enhance IoV networks' performance and scalability.

This framework can be extended to real-world scenarios through pilot deployments in controlled vehicular testbeds or HiL simulations, combining real hardware components with virtual testing environments. These efforts would evaluate its adaptability to real-time challenges such as fluctuating network quality, dynamic traffic patterns, and hardware limitations. Additionally, partial integration with real-world IoV datasets could refine the framework, providing critical insights for large-scale deployments while addressing practical issues like communication reliability and scalability. Such extensions would bridge the gap between simulation-based validation and real-world applicability.

References

[1] X. Han, D. Tian, J. Zhou, X. Duan, Z. Sheng, and V. C. M. Leung, "Privacy-preserving proxy re-encryption with decentralized trust management for MEC-empowered VANETs," *IEEE Trans. Intell. Veh.*, vol. 8, no. 8, pp. 4105–4119, 2023.

[2] E. Karimi, Y. Chen, and B. Akbari, "Intelligent and decentralized resource allocation in vehicular edge computing networks," *IEEE Internet Things Mag.*, vol. 6, no. 4, pp. 112–117, 2023.

[3] B. Hazarika, K. Singh, S. Biswas, S. Mumtaz, and C.-P. Li, "Multi-agent DRL-based task offloading in multiple RIS-aided IoV networks," *IEEE Trans. Veh. Technol.*, vol. 73, no. 1, pp. 1175–1190, 2024.

[4] L. Zhao, Z. Zhao, E. Zhang, A. Hawbani, A. Y. Al-Dubai, Z. Tan, and A. Hussain, "A digital twin-assisted intelligent partial offloading approach for vehicular edge computing," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 11, pp. 3386–3400, 2023.

[5] B. Cao, Z. Li, X. Liu, Z. Lv, and H. He, "Mobility-aware multiobjective task offloading for vehicular edge computing in digital twin environment," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3046–3055, 2023.

[6] N. Cheng, X. Wang, Z. Li, Z. Yin, T. Luan, and X. S. Shen, "Toward enhanced reinforcement learning-based resource management via digital twin: Opportunities, applications, and challenges," *IEEE Netw.*, pp. 1–1, 2024.

[7] J. Zheng, T. H. Luan, Y. Zhang, R. Li, Y. Hui, L. Gao, and M. Dong, "Data synchronization in vehicular digital twin network: A game theoretic approach," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 11, pp. 7635–7647, 2023.

[8] 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.*, vol. 41, no. 11, pp. 3369–3385, 2023.

[9] B. Li, W. Xie, Y. Ye, L. Liu, and Z. Fei, "Flexedge: Digital twin-enabled task offloading for UAV-aided vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 72, no. 8, pp. 11 086–11 091, 2023.

[10] L. Nie, X. Wang, Q. Zhao, Z. Shang, L. Feng, and G. Li, "Digital twin for transportation big data: A reinforcement learning-based network traffic prediction approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 1, pp. 896–906, 2024.

[11] X. Luo, J. Wen, J. Kang, J. Nie, Z. Xiong, Y. Zhang, Z. Yang, and S. Xie, "Privacy attacks and defenses for digital twin migrations in vehicular metaverses," *IEEE Netw.*, vol. 37, no. 6, pp. 82–91, 2023.

[12] K. Wang, Z. Li, K. Nonomura, T. Yu, K. Sakaguchi, O. Hashash, and W. Saad, "Smart mobility digital twin based automated vehicle navigation system: A proof of concept," *IEEE Trans. Intell. Veh.*, vol. 9, no. 3, pp. 4348–4361, 2024.

[13] B. Hazarika, K. Singh, O. A. Dobre, C.-P. Li, and T. Q. Duong, "Quantum-enhanced federated learning for metaverse-empowered vehicular networks," *IEEE Trans. Commun.*, pp. 1–1, 2024.

[14] X. Zhang, A. Mavromatis, A. Vafeas, R. Nejabati, and D. Simeonidou, "Federated feature selection for horizontal federated learning in IoT networks," *IEEE Internet Things J.*, vol. 10, no. 11, pp. 10 095–10 112, 2023.

[15] B. Gu, A. Xu, Z. Huo, C. Deng, and H. Huang, "Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6103–6115, 2022.

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

[17] P. Zhang, N. Chen, S. Li, K.-K. R. Choo, C. Jiang, and S. Wu, "Multi-domain virtual network embedding algorithm based on horizontal federated learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 3363–3375, 2023.

[18] Y. Liu, Y. Kang, T. Zou, Y. Pu, Y. He, X. Ye, Y. Ouyang, Y.-Q. Zhang, and Q. Yang, "Vertical federated learning: Concepts, advances, and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 3615–3634, 2024.

[19] X. Zhang, Z. Chang, T. Hu, W. Chen, X. Zhang, and G. Min, "Vehicle selection and resource allocation for federated learning-assisted vehicular network," *IEEE Trans. Mob. Comput.*, vol. 23, no. 5, pp. 3817–3829, 2024.

[20] A. S. Kumar, L. Zhao, and X. Fernando, "Asynchronous federated based vehicular edge computation offloading," *IEEE Trans. Veh. Technol.*, pp. 1–11, 2024.

[21] 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.

[22] B. Hazarika and K. Singh, "AFL-DMAAC: Integrated resource management and cooperative caching for URLLC-IoV networks," *IEEE Trans. Intell. Veh.*, vol. 9, no. 6, pp. 5101–5117, 2024.

[23] 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.*, vol. 9, no. 1, pp. 2923–2939, 2024.

[24] Y. Liu, L. Huo, J. Wu, and A. K. Bashir, "Swarm learning-based dynamic optimal management for traffic congestion in 6G-driven intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 7831–7846, 2023.

[25] C. You, D. Feng, K. Guo, H. H. Yang, C. Feng, and T. Q. S. Quek, "Semi-synchronous personalized federated learning over mobile edge networks," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 4, pp. 2262–2277, 2023.

[26] 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.

[27] 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, pp. 1533–1538.

[28] 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.

[29] Y. Hou, Z. Wei, R. Zhang, X. Cheng, and L. Yang, "Hierarchical task offloading for vehicular fog computing based on multi-agent deep reinforcement learning," *IEEE Trans. Wirel. Commun.*, vol. 23, no. 4, pp. 3074–3085, 2024.

[30] N. Li, X. Song, K. Li, R. Jiang, and J. Li, "Multiagent federated deep-reinforcement-learning-enabled resource allocation for an air–ground-integrated internet of vehicles network," *IEEE Internet Comput.*, vol. 27, no. 5, pp. 15–23, 2023.

[31] M. K. Hasan, N. Jahan, M. Z. A. Nazri, S. Islam, M. Attique Khan, A. I. Alzahrani, N. Alalwan, and Y. Nam, "Federated learning for computational offloading and resource management of vehicular edge computing in 6G-V2X network," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3827–3847, 2024.

[32] R. Zhang, K. Xiong, H. Du, D. Niyato, J. Kang, X. Shen, and H. V. Poor, "Generative AI-enabled vehicular networks: Fundamentals, framework, and case study," *IEEE Netw.*, vol. 38, no. 4, pp. 259–267, 2024.

[33] S. Barmada, P. D. Barba, N. Fontana, M. E. Mognaschi, and M. Tucci, "Electromagnetic field reconstruction and source identification using conditional variational autoencoder and CNN," *IEEE J. Multiscale Multiphysics Comput. Tech.*, vol. 8, pp. 322–331, 2023.

[34] Z. Zheng, L. Yang, and Z. Zhang, "Conditional variational autoencoder informed probabilistic wind power curve modeling," *IEEE Trans. Sustain. Energy*, vol. 14, no. 4, pp. 2445–2460, 2023.

[35] J. Zhang, L. Zhao, K. Yu, G. Min, A. Y. Al-Dubai, and A. Y. Zomaya, "A novel federated learning scheme for generative adversarial networks," *IEEE Trans. Mob. Comput.*, vol. 23, no. 5, pp. 3633–3649, 2024.

[36] Y. Ye, Z. Zhao, L. Liu, J. Feng, J. Du, and Q. Pei, "Federated generative artificial intelligence empowered traffic flow prediction under vehicular

computing power networks," *IEEE Internet Things Mag.*, vol. 7, no. 3, pp. 56–61, 2024.

[37] X. Huang, P. Li, H. Du, J. Kang, D. Niyato, D. I. Kim, and Y. Wu, "Federated learning-empowered AI-generated content in wireless networks," *IEEE Netw.*, pp. 1–1, 2024.

[38] H. Li, Z. Cai, J. Wang, J. Tang, W. Ding, C.-T. Lin, and Y. Shi, "FedTP: Federated learning by transformer personalization," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2023.

[39] F. Barbato, G. Rizzoli, and P. Zanuttigh, "Depthformer: Multimodal positional encodings and cross-input attention for transformer-based segmentation networks," in *Proc. IEEE ICASSP*, 2023, pp. 1–5.

[40] Q. Zhang, M. Ge, H. Zhu, E. Ambikairajah, Q. Song, Z. Ni, and H. Li, "An empirical study on the impact of positional encoding in transformer-based monaural speech enhancement," in *Proc. IEEE ICASSP*, 2024, pp. 1001–1005.

[41] W. Sun, W. Xiong, H. Chen, R. Chiplunkar, and B. Huang, "A novel CVAE-based sequential monte carlo framework for dynamic soft sensor applications," *IEEE Trans. Ind. Inform.*, vol. 20, no. 3, pp. 3789–3800, 2024.

[42] B. Hazarika, K. Singh, S. K. Singh, C. Pan, and T. Q. Duong, "Asynchronous federated learning-based resource management in URLLC-IoV networks," in *Proc. IEEE GLOBECOM*, 2023, pp. 6874–6879.

[43] B. Hazarika, K. Singh, C.-P. Li, and S. Biswas, "Multi-agent DRL-based computation offloading in multiple ris-aided IoV networks," in *Proc. IEEE MILCOM*, 2022, pp. 1–6.

[44] Y. Liu, L. Huo, X. Zhang, and J. Wu, "A multi-objective resource pre-allocation scheme using SDN for intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 1, pp. 571–586, 2024.

Piyush Singh (**Graduate Student Member, IEEE**) received the B.Tech. degree in Information Technology from Dr. A.P.J. Abdul Kalam Technical University, India, in 2020. He is currently pursuing Ph.D. in the Department of Electrical Engineering from the National Sun Yat-sen University (NSYSU), Kaohsiung, Taiwan. His current research interests include machine learning, artificial intelligence, edge computing, wireless networks, and 5G/6G communication.

Bishmita Hazarika (**Member, IEEE**) received the B.Tech degree in Information Technology from the Central Institute of Technology (CIT), Kokrajhar, India, in 2018, the M.Tech. degree in Computer Science and Engineering from the Indian Institute of Information Technology, Guwahati (IIIT-G), India, in 2020, and Ph.D. degree in Communication Engineering from the National Sun Yat-sen University (NSYSU), Kaohsiung, Taiwan, in 2024. She is currently a post-doc research fellow at Memorial University, Canada. Her current research interests include heterogeneous vehicular networks, edge computing, digital twins, machine learning, artificial intelligence, and quantum machine learning.

Keshav Singh (Member, IEEE) received the Ph.D. degree in Communication Engineering from National Central University, Taiwan, in 2015. He is currently with the Institute of Communications Engineering, National Sun Yat-sen University (NSYSU), Taiwan, as an Associate Professor. He is also an Adjunct Professor at the Memorial University, Canada. Prior to this, he held the position of Research Associate from 2016 to 2019 at the Institute of Digital Communications, University of Edinburgh, U.K. From 2019 to 2020, he was associated with the University College Dublin, Ireland as a Research Fellow. He leads research in the areas of green communications, resource allocation, transceiver design for full-duplex radio, ultra-reliable low-latency communication, non-orthogonal multiple access, machine learning for wireless communications, integrated sensing and communications, non-terrestrial networks, and large intelligent surface-assisted communications.

Dr. Singh chaired workshops on conferences like IEEE GLOBECOM 2023 and IEEE WCNC 2024. He also serves as leading guest editor for IEEE Transactions on Green Communications and Networking Special Issue on Design of Green Near-Field Wireless Communication Networks and IEEE Internet of Things Journal Special Issue on Positioning and Sensing for Near-Filed (NF)-driven Internet-of-Everything.

Wan-Jen Huang (**Member, IEEE**) received her B.S. degree in the Department of Physics from National Taiwan University, Taiwan, in 2001; M.S. degree in Institute of Communication Engineering from National Taiwan University, Taiwan, in 2003; and Ph.D. degree in the Department of Electrical Engineering from the University of Southern California, CA, in 2008. She is currently working as an associate professor with the Institute of Communications Engineering at the National Sun Yat-Sen University, Taiwan. Her research interests include signal estimation and detection and baseband signal processing of wireless communication systems such as massive MIMO, RIS, cooperative communications, etc.

Trung Q. Duong (Fellow, IEEE) is a Canada Excellence Research Chair (CERC) and a Full Professor at Memorial University, Canada. He is also the Chair Professor in Telecommunications at Queen's University Belfast, UK and a Research Chair of Royal Academy of Engineering, UK. He was a Distinguished Advisory Professor at Inje University, South Korea (2017-2019), an Adjunct Professor and the Director of Institute for AI and Big Data at Duy Tan University, Vietnam (2012-present), and a Visiting Professor (under Eminent Scholar program) at Kyung Hee University, South Korea (2023-2025). His current research interests include wireless communications, quantum machine learning, and quantum optimisation.

Dr. Duong has served as an Editor/Guest Editor for the IEEE Transactions on Wireless Communications, IEEE Transactions on Communications, IEEE Transactions on Vehicular Technology, IEEE Communications Letters, IEEE Wireless Communications Letters, IEEE Wireless Communications, IEEE Communications Magazines, and IEEE Journal on Selected Areas in Communications. He is the Editor-in-Chief of IEEE Communications Surveys & Tutorials. He received the Best Paper Award at the IEEE VTC-Spring 2013, IEEE ICC 2014, IEEE GLOBECOM 2016, 2019, 2022, IEEE DSP 2017, IWCMC 2019, 2023, 2024 and IEEE CAMAD 2023, 2024. He has received the two prestigious awards from the Royal Academy of Engineering (RAEng): RAEng Research Chair (2021-2025) and the RAEng Research Fellow (2015-2020). He is the recipient of the prestigious Newton Prize 2017. He is a fellow of the Engineering Institute of Canada (EIC).